Версия

# Оглавление

Table of contents

# TaskManager

**TaskManager** — это приложение для управления задачами.

## Функционал

Приложение предоставляет следующий функционал:

* **Управление задачами** : хранение данных (приоритет, сложность, статус, прогресс, статус, сроки, время, реальные показатели, метаданные), создание, редактирование, удаление, управление иерархией, планирование в календаре.
* **Статистика** : анализ задач по категориям, тегам, сложности, приоритету, времени и срокам.
* **Анализ** : ранжирование задач, выявление отстающих.
* **Прогнозирование** : оценка времени, реального показателя, срока, вероятности выполнения в срок.
* **Уведомления** : напоминания о запланированном времени выполнения.

## Набор технологий

### Основной набор технологий

Основным набором технологий для проекта является:

* Платформа разработки .NET и язык программирования C#
* Фреймворк для построения реактивных пользовательских интерфейсов ReactiveUI
* Фреймфорк Avalonia для создания кроссплатформенного пользовательского интерфейса
* База данных SQLite для локального хранения данных
* ORM-фреймворк Entity Framework Core для работы с базами данных
* Платформа машинного обучения Accord.NET
* Библиотека для модульного тестирования NUnit

### Дополнительный набор технологий

Дополнительно были выбраны следующие технологии:

* Дополнение Autofac для библиотеки Splat, предоставляющее интеграцию с IoC-контейнером
* Библиотека для работы с JSON Newtonsoft.Json
* Библиотека для построения интерактивных графиков и диаграмм LiveCharts
* Пользовательская реализация элемента управления Avalonia для работы с календарём CalendarControl.Avalonia

## Контакты

* Репозиторий
* GitHub

# Алфавитный указатель пространств имен

## Список пакетов

Полный список документированных пакетов.

**MachineLearning**  24

**MachineLearning.Converters**  25

**MachineLearning.DataProcessors**  26

**MachineLearning.DistanceMetrics**  27

**MachineLearning.Interfaces**  28

**MachineLearning.Interfaces.Generals**  29

**MachineLearning.LearningEvaluators**  30

**MachineLearning.LearningModels**  31

**MachineLearning.Scalers**  32

**MachineLearning.ScoreMetrics**  33

**Model**  34

**Model.Interfaces**  35

**Model.Tasks**  36

**Model.Times**  37

**TrackableFeatures**  38

**View**  39

**View.Android**  40

**View.Controls**  41

**View.Converters**  42

**View.DataTemplates**  43

**View.Desktop**  44

**View.Desktop.Linux**  45

**View.Desktop.macOS**  46

**View.Desktop.Windows**  47

**View.Extensions**  48

**View.Implementations**  49

**View.iOS**  50

**View.Technilcals**  51

**View.Views**  52

**View.Views.Modals**  53

**View.Views.Pages**  54

**ViewModel**  55

**ViewModel.Implementations**  56

**ViewModel.Implementations.AppStates**  57

**ViewModel.Implementations.AppStates.Sessions**  58

**ViewModel.Implementations.AppStates.Sessions.Database**  59

**ViewModel.Implementations.AppStates.Sessions.Database.DbContexts**  60

**ViewModel.Implementations.AppStates.Sessions.Database.Domains**  61

**ViewModel.Implementations.AppStates.Sessions.Database.Entities**  62

**ViewModel.Implementations.AppStates.Sessions.Database.Mappers**  63

**ViewModel.Implementations.AppStates.Settings**  64

**ViewModel.Implementations.DataManagers**  65

**ViewModel.Implementations.DataManagers.Editors**  66

**ViewModel.Implementations.DataManagers.Factories**  67

**ViewModel.Implementations.Mocks**  68

**ViewModel.Implementations.ModelLearning**  69

**ViewModel.Implementations.ModelLearning.Converters**  70

**ViewModel.Implementations.Sessions**  71

**ViewModel.Implementations.Sessions.Database**  72

**ViewModel.Implementations.Sessions.Database.Entities**  73

**ViewModel.Interfaces**  74

**ViewModel.Interfaces.AppStates**  75

**ViewModel.Interfaces.AppStates.Sessions**  76

**ViewModel.Interfaces.AppStates.Settings**  77

**ViewModel.Interfaces.DataManagers**  78

**ViewModel.Interfaces.DataManagers.Generals**  79

**ViewModel.Interfaces.ModelLearning**  80

**ViewModel.Technicals**  81

**ViewModel.ViewModels**  82

**ViewModel.ViewModels.Modals**  83

**ViewModel.ViewModels.Pages**  84

# Иерархический список классов

## Иерархия классов

Иерархия классов.

ViewModel.Implementations.AppStates.Settings.AppConfigurationFormat 102

Application

View.App 98

View.iOS.Application 105

AvaloniaAppDelegate

View.iOS.AppDelegate 103

AvaloniaMainActivity

View.Android.MainActivity 348

ViewModel.Technicals.CalendarInterval 161

ViewModel.ViewModels.Modals.CopyTasksViewModelResult 175

MachineLearning.DataProcessorResult< T > 178

DbContext

ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext 119

ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.SqliteDbContext 417

EventArgs

ViewModel.Implementations.Mocks.NotificationEventArgs 380

IActivatableViewModel

ViewModel.ViewModels.BaseViewModel 158

ViewModel.ViewModels.BaseDialogViewModel< A, R > 122

ViewModel.ViewModels.Modals.AddTaskViewModel 88

ViewModel.ViewModels.Modals.EditTaskViewModel 205

ViewModel.ViewModels.Modals.EditTimeIntervalViewModel 208

ViewModel.ViewModels.Modals.RemoveTasksViewModel 405

ViewModel.ViewModels.Modals.TasksViewModel< A, R > 486

ViewModel.ViewModels.Modals.AddTimeIntervalViewModel 91

ViewModel.ViewModels.Modals.CopyTasksViewModel 172

ViewModel.ViewModels.Modals.MoveTasksViewModel 375

ViewModel.ViewModels.BasePageViewModel 128

ViewModel.ViewModels.Pages.EditorViewModel 202

ViewModel.ViewModels.Pages.SettingsViewModel 411

ViewModel.ViewModels.Pages.StatisticViewModel 424

ViewModel.ViewModels.Pages.TimeViewModel 515

ViewModel.ViewModels.Pages.ToDoListViewModel 520

ViewModel.ViewModels.MainViewModel 353

ViewModel.Interfaces.AppStates.IAppLifeState 228

View.Android.MainActivity 348

View.Desktop.Linux.DesktopAppLifeState 197

View.Desktop.Windows.DesktopAppLifeState 199

View.Desktop.macOS.DesktopAppLifeState 198

View.iOS.AppDelegate 103

ViewModel.Implementations.Mocks.MockAppLifeState 367

ICloneable

Model.Tasks.TaskComposite 428

ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain 434

Model.Tasks.TaskElement 446

ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskElementDomain 459

ViewModel.Technicals.TaskMetadata 478

MachineLearning.Interfaces.Generals.IDataProcessor< T, R > 235

MachineLearning.Interfaces.IPointDataProcessor 265

MachineLearning.DataProcessors.CorrelationColumnProcessor 177

MachineLearning.DataProcessors.DuplicatesRowProcessor 200

MachineLearning.DataProcessors.LowVariationColumnProcessor 346

MachineLearning.DataProcessors.OutlierRowProcessor 382

MachineLearning.Interfaces.IPrimaryPointDataProcessor 267

MachineLearning.DataProcessors.InvalidValuesColumnProcessor 262

MachineLearning.Interfaces.IPrimaryPointDataProcessor 267

ViewModel.Interfaces.ModelLearning.IDataTransformer< T, R > 236

ViewModel.Implementations.ModelLearning.Converters.MetadataCategoriesTransformer 356

ViewModel.Implementations.ModelLearning.Converters.MetadataTagsTransformer 362

MachineLearning.Interfaces.Generals.IDistanceMetric< T > 240

MachineLearning.Interfaces.IPointDistanceMetric 266

MachineLearning.DistanceMetrics.EuclideanDistanceMetric 212

ViewModel.Implementations.AppStates.Sessions.Database.Domains.IDomain 241

ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain 434

ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskElementDomain 459

ViewModel.Implementations.AppStates.Sessions.Database.Domains.TimeIntervalElementDomain 494

ViewModel.Interfaces.DataManagers.Generals.IEditorService 244

ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T > 242

ViewModel.Interfaces.DataManagers.ITaskElementsEditorProxy 309

ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy 469

ViewModel.Interfaces.DataManagers.ITasksEditorProxy 311

ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy 482

ViewModel.Interfaces.DataManagers.ITimeIntervalElementsEditorProxy 321

ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy 498

ViewModel.Interfaces.DataManagers.Generals.IFactory< out T > 245

ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.IDbContextFactory< T > 238

ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.DbContextFactory 182

ViewModel.Implementations.DataManagers.Factories.TaskCompositeFactory 440

ViewModel.Implementations.DataManagers.Factories.TaskElementCreatorProxyFactory 457

ViewModel.Implementations.DataManagers.Factories.TaskElementFactory 465

ViewModel.Implementations.DataManagers.Factories.TaskMetadataFactory 481

ViewModel.Implementations.DataManagers.Factories.TimeIntervalElementFactory 497

ViewModel.Implementations.ModelLearning.Converters.ScalerFactory 409

ViewModel.Interfaces.AppStates.IFileService 247

ViewModel.Implementations.AppStates.FileService 223

MachineLearning.Interfaces.Generals.ILearningConverter< T, R, DT, DR > 252

MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR > 125

MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR > 138

ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR > 144

ViewModel.Implementations.ModelLearning.Converters.DeadlineTaskElementLearningConverter 192

ViewModel.Implementations.ModelLearning.Converters.ExecutionChanceTaskElementLearningConverter 215

ViewModel.Implementations.ModelLearning.Converters.PlannedRealTaskElementLearningConverter 387

ViewModel.Implementations.ModelLearning.Converters.PlannedTimeTaskElementLearningConverter 393

MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR > 153

MachineLearning.Interfaces.Generals.ISupervisedLearningConverter< T, R, D, DT, DR > 287

MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR > 138

MachineLearning.Interfaces.Generals.IUnsupervisedLearningConverter< T, R, D, DT, DR > 327

MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR > 153

MachineLearning.Interfaces.Generals.ILearningEvaluator 254

MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R > 289

MachineLearning.Interfaces.IClassificationEvaluator 229

MachineLearning.LearningEvaluators.ClassificationCrossValidationEvaluator 163

MachineLearning.Interfaces.IRegressionEvaluator 269

MachineLearning.LearningEvaluators.RegressionCrossValidationEvaluator 400

MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R > 130

MachineLearning.LearningEvaluators.ClassificationCrossValidationEvaluator 163

MachineLearning.LearningEvaluators.RegressionCrossValidationEvaluator 400

MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R > 329

MachineLearning.Interfaces.IClusteringEvaluator 232

MachineLearning.LearningEvaluators.ClusteringCrossValidationEvaluator 167

MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R > 149

MachineLearning.LearningEvaluators.ClusteringCrossValidationEvaluator 167

MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator 116

MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R > 130

MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R > 149

MachineLearning.Interfaces.Generals.ILearningModel< T, R > 255

MachineLearning.Interfaces.Generals.ISupervisedLearningModel< T, R > 291

MachineLearning.Interfaces.IClassificationModel 230

MachineLearning.LearningModels.KNearestNeighborsModel 340

MachineLearning.Interfaces.IRegressionModel 270

MachineLearning.LearningModels.MultipleLinearRegressionModel 378

MachineLearning.Interfaces.Generals.IUnsupervisedLearningModel< T, R > 331

MachineLearning.Interfaces.IClusteringModel 233

MachineLearning.LearningModels.KMeanLearningModel 337

IList

Model.Interfaces.ITaskComposite 299

Model.Tasks.TaskComposite 428

Model.Interfaces.ITimeIntervalList 323

Model.Times.TimeIntervalList 503

TrackableFeatures.TrackableCollection< T > 522

Model.Tasks.TaskComposite 428

Model.Times.TimeIntervalList 503

IList

Model.Interfaces.ITaskComposite 299

Model.Interfaces.ITimeIntervalList 323

TrackableFeatures.TrackableCollection< T > 522

ViewModel.Interfaces.AppStates.Settings.ILocalizationManager 256

View.Implementations.AvaloniaLocalizationManager 110

ViewModel.Implementations.Mocks.MockLocalizationManager 368

ViewModel.Interfaces.AppStates.ILogger 257

ViewModel.Implementations.AppStates.FileLogger 221

ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 > 258

ViewModel.Implementations.AppStates.Sessions.Database.Mappers.MetadataMapper 360

ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskCompositeMapper 442

ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskElementMapper 467

ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskMapper 476

ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TimeIntervalMapper 507

IMetric

MachineLearning.LearningModels.MetricAdapter 364

ViewModel.Interfaces.ModelLearning.IModelTeacher< D > 260

ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR > 250

ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR > 135

ViewModel.Implementations.ModelLearning.DeadlineTaskElementEvaluatorLearningController 190

ViewModel.Implementations.ModelLearning.ExecutionChanceTaskElementEvaluatorLearningController 213

ViewModel.Implementations.ModelLearning.PlannedRealTaskElementEvaluatorLearningController 385

ViewModel.Implementations.ModelLearning.PlannedTimeTaskElementEvaluatorLearningController 391

ViewModel.Interfaces.ModelLearning.ISupervisedLearningController< T, R, D, DT, DR > 285

ViewModel.Interfaces.ModelLearning.IUnsupervisedLearningController< T, R, D, DT, DR > 325

ViewModel.Interfaces.AppStates.INotificationManager 261

View.Android.AndroidNotificationManager 96

View.Desktop.Linux.LinuxNotificationManager 345

View.Desktop.Windows.WindowsNotificationManager 536

View.Desktop.macOS.MacOsNotificationManager 347

View.iOS.IOsNotificationManager 264

ViewModel.Implementations.Mocks.MockNotificationManager 369

INotifyCollectionChanged

TrackableFeatures.TrackableCollection< T > 522

INotifyDataErrorInfo

TrackableFeatures.TrackableObject 529

MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator 116

MachineLearning.LearningModels.KMeanLearningModel 337

MachineLearning.LearningModels.KNearestNeighborsModel 340

Model.Tasks.TaskElement 446

Model.Times.TimeIntervalElement 491

ViewModel.Implementations.AppStates.Sessions.Database.Domains.TimeIntervalElementDomain 494

TrackableFeatures.TrackableCollection< T > 522

View.Implementations.AvaloniaLocalizationManager 110

View.Implementations.AvaloniaThemeManager 114

ViewModel.Implementations.AppStates.Sessions.DbSession 184

ViewModel.Implementations.AppStates.Settings.AppSettings 106

ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy 498

ViewModel.Implementations.DataManagers.TaskElementCreatorProxy 450

ViewModel.Technicals.AppConfiguration 99

ViewModel.Technicals.TaskMetadata 478

INotifyPropertyChanged

TrackableFeatures.TrackableObject 529

ViewModel.Interfaces.DataManagers.Generals.IProxy< out T > 268

ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T > 242

ViewModel.Interfaces.DataManagers.ITaskElementProxy 305

ViewModel.Implementations.DataManagers.TaskElementCreatorProxy 450

IRecyclingDataTemplate

View.DataTemplates.PageDataTemplate 383

ViewModel.Interfaces.AppStates.IResourceService 272

View.Implementations.AvaloniaResourceService 112

ViewModel.Implementations.Mocks.MockResourceService 371

MachineLearning.Interfaces.IScaler 273

MachineLearning.Scalers.MinMaxScaler 365

MachineLearning.Interfaces.Generals.IScoreMetric 275

MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T > 293

MachineLearning.Interfaces.IClassificationScoreMetric 231

MachineLearning.ScoreMetrics.AccuracyScoreMetric 85

MachineLearning.ScoreMetrics.AdjustedRandIndexScoreMetric 94

MachineLearning.ScoreMetrics.F1ScoreMetric 219

MachineLearning.Interfaces.IRegressionScoreMetric 271

MachineLearning.ScoreMetrics.RSquaredScoreMetric 407

MachineLearning.ScoreMetrics.SmapeScoreMetric 415

MachineLearning.Interfaces.Generals.IUnsupervisedScoreMetric< T, D > 333

MachineLearning.Interfaces.IClusteringScoreMetric 234

MachineLearning.ScoreMetrics.SilhouetteScoreMetric 413

ViewModel.Interfaces.AppStates.ISerializer 276

ViewModel.Implementations.AppStates.JsonSerializer 335

ViewModel.Interfaces.AppStates.IStorageService 283

ViewModel.Interfaces.AppStates.Sessions.ISession 278

ViewModel.Implementations.AppStates.Sessions.DbSession 184

ViewModel.Interfaces.AppStates.Settings.ISettings 282

ViewModel.Implementations.AppStates.Settings.AppSettings 106

Model.Interfaces.ITask 295

Model.Interfaces.ITaskComposite 299

Model.Interfaces.ITaskElement 301

Model.Tasks.TaskElement 446

ViewModel.Interfaces.DataManagers.ITaskElementProxy 305

ViewModel.Interfaces.DataManagers.ITaskElementsEditorProxy 309

ViewModel.Interfaces.DataManagers.ITasksEditorProxy 311

ViewModel.Interfaces.AppStates.Sessions.ItemsUpdatedEventArgs 315

ViewModel.Interfaces.AppStates.Settings.IThemeManager 317

View.Implementations.AvaloniaThemeManager 114

ViewModel.Implementations.Mocks.MockThemeManager 373

Model.Interfaces.ITimeInterval 318

Model.Interfaces.ITimeIntervalElement 319

Model.Times.TimeIntervalElement 491

ViewModel.Interfaces.DataManagers.ITimeIntervalElementsEditorProxy 321

Model.Interfaces.ITimeIntervalList 323

ViewModel.Interfaces.ITimeScheduler 324

ViewModel.Implementations.AppStates.TimeScheduler 512

ITreeDataTemplate

View.DataTemplates.TaskDataTemplate 444

IValueConverter

View.Converters.FormattableToStringConverter 227

View.Converters.StatisticElementsToColumnSeriesConverter 421

View.Converters.StatisticElementsToPieSeriesConverter 422

View.Converters.TagsToStringConverter 427

MachineLearning.LearningModelData< T, R > 343

MarkupExtension

View.Extensions.DesignExtension 196

View.Extensions.EnumsExtension 211

ViewModel.Implementations.AppStates.Sessions.Database.Entities.MetadataEntity 358

View.Technilcals.PageMetadata 384

View.Desktop.Linux.Program 397

View.Desktop.macOS.Program 398

View.Desktop.Windows.Program 399

ReactiveObject

ViewModel.ViewModels.BaseViewModel 158

ReactiveUserControl

View.Views.MainView 350

View.Views.Modals.AddTaskView 87

View.Views.Modals.AddTimeIntervalView 90

View.Views.Modals.CopyTasksView 171

View.Views.Modals.EditTaskView 204

View.Views.Modals.EditTimeIntervalView 207

View.Views.Modals.MoveTasksView 374

View.Views.Modals.RemoveTasksView 404

View.Views.Pages.EditorView 201

View.Views.Pages.SettingsView 410

View.Views.Pages.StatisticView 423

View.Views.Pages.TimeView 514

View.Views.Pages.ToDoListView 519

ViewModel.Technicals.StatisticElement 419

ViewModel.Implementations.Sessions.Database.Entities.TagEntity 426

ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskCompositeEntity 439

ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity 462

ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskEntity 474

ViewModel.ViewModels.Modals.TasksViewModelArgs 489

ViewModel.ViewModels.Modals.ItemsTasksViewModelArgs 313

ViewModel.ViewModels.Modals.TimeIntervalViewModelArgs 508

ViewModel.Implementations.AppStates.Sessions.Database.Entities.TimeIntervalEntity 501

ViewModel.ViewModels.Modals.TimeIntervalViewModelResult 510

ViewModel.Technicals.ToDoListElement 517

UserControl

View.Controls.DateTimePicker 180

MachineLearning.LearningEvaluators.ValidationFold 534

Window

View.Views.MainWindow 355

# Алфавитный указатель классов

## Классы

Классы с их кратким описанием.

**MachineLearning.ScoreMetrics.AccuracyScoreMetric (Класс метрики оценки точности для модели обучения классификации )** 85

**View.Views.Modals.AddTaskView (Класс пользовательского элемента диалога добавления задачи )** 87

**ViewModel.ViewModels.Modals.AddTaskViewModel (Класс диалога добавления задачи )** 88

**View.Views.Modals.AddTimeIntervalView (Класс пользовательского элемента диалога добавления временного интервала )** 90

**ViewModel.ViewModels.Modals.AddTimeIntervalViewModel (Класс диалога добавления временного интервала )** 91

**MachineLearning.ScoreMetrics.AdjustedRandIndexScoreMetric (Класс метрики оценки скорректированного индекса Рэнда для модели обучения классификации )** 94

**View.Android.AndroidNotificationManager (Класс менеджера уведомлений Android )** 96

**View.App (Класс приложения )** 98

**ViewModel.Technicals.AppConfiguration (Класс конфигурации приложения )** 99

**ViewModel.Implementations.AppStates.Settings.AppConfigurationFormat (Класс формата конфигурации приложении )** 102

**View.iOS.AppDelegate (Класс делегата приложения )** 103

**View.iOS.Application (Класс приложения )** 105

**ViewModel.Implementations.AppStates.Settings.AppSettings (Класс настроек приложения )** 106

**View.Implementations.AvaloniaLocalizationManager (Класс менеджера локализации Avalonia )** 110

**View.Implementations.AvaloniaResourceService (Класс сервис ресурсов Avalonia )** 112

**View.Implementations.AvaloniaThemeManager (Класс менеджер тем Avalonia )** 114

**MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator (Абстрактный класс базовой оценки модели обучения методом кросс-валидации )** 116

**ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext (Базовый класс контекста базы данных )** 119

**ViewModel.ViewModels.BaseDialogViewModel< A, R > (Базовый абстрактный класс контроллера диалога )** 122

**MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR > (Абстрактный класс базового конвертора данных в данные для предсказания и наоборот )** 125

**ViewModel.ViewModels.BasePageViewModel (Базовый класс контроллера страницы )** 128

**MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R > (Абстрактный класс базовой оценки модели обучения с учителем методом кросс-валидации )** 130

**ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR > (Абстрактный класс базового контроллера обучения модели обучения с учителем )** 135

**MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR > (Абстрактный класс базового конвертора данных в данные для предсказания с учителем и наоборот )** 138

**ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR > (Абстрактный класс базового конвертора элементраных задач в данные для предсказания с учителем и наоборот )** 144

**MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R > (Абстрактный класс базовой оценки модели обучения без учителя методом кросс-валидации )** 149

**MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR > (Абстрактный класс базового конвертора данных в данные для предсказания без учителя и наоборот )** 153

**ViewModel.ViewModels.BaseViewModel (Базовый класс для контроллеров )** 158

**ViewModel.Technicals.CalendarInterval (Класс интервала календаря )** 161

**MachineLearning.LearningEvaluators.ClassificationCrossValidationEvaluator (Класс оценки модели обучения классификации методом кросс-валидации )** 163

**MachineLearning.LearningEvaluators.ClusteringCrossValidationEvaluator (Класс оценки модели обучения кластеризации методом кросс-валидации )** 167

**View.Views.Modals.CopyTasksView (Класс пользовательского элемента диалога копирования задач )** 171

**ViewModel.ViewModels.Modals.CopyTasksViewModel (Класс диалога копирования задач )** 172

**ViewModel.ViewModels.Modals.CopyTasksViewModelResult (Класс результата диалога CopyTasksViewModel )** 175

**MachineLearning.DataProcessors.CorrelationColumnProcessor (Класс обработчика столбцов для устранения корреляции )** 177

**MachineLearning.DataProcessorResult< T > (Структура хранения данных результата преобразования )** 178

**View.Controls.DateTimePicker (Класс пользовательского элемента для выбора даты и времени )** 180

**ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.DbContextFactory (Класс фабрики, создающая контексты базы данных )** 182

**ViewModel.Implementations.AppStates.Sessions.DbSession (Класс сессии базы данных )** 184

**ViewModel.Implementations.ModelLearning.DeadlineTaskElementEvaluatorLearningController (Класс контроллера обучения модели обучения с учителем для предсказания срока элементарной задачи )** 190

**ViewModel.Implementations.ModelLearning.Converters.DeadlineTaskElementLearningConverter (Класс конвертора элементраных задач в данные для предсказания срока с учителем и наоборот )** 192

**View.Extensions.DesignExtension (Класс расширений синтаксиса разметки для дизайнера, который использует контейнер зависимости )** 196

**View.Desktop.Linux.DesktopAppLifeState (Класс для управления состоянием жизненного цикла desktop-приложения )** 197

**View.Desktop.macOS.DesktopAppLifeState (Класс для управления состоянием жизненного цикла desktop-приложения )** 198

**View.Desktop.Windows.DesktopAppLifeState (Класс для управления состоянием жизненного цикла desktop-приложения )** 199

**MachineLearning.DataProcessors.DuplicatesRowProcessor (Класс обработчика строк для устранения дубликатов )** 200

**View.Views.Pages.EditorView (Класс пользовательского элемента страницы изменения задач )** 201

**ViewModel.ViewModels.Pages.EditorViewModel (Класс контроллера страницы изменения задач )** 202

**View.Views.Modals.EditTaskView (Класс пользовательского элемента диалога изменения задачи )** 204

**ViewModel.ViewModels.Modals.EditTaskViewModel (Класс диалога изменения задачи )** 205

**View.Views.Modals.EditTimeIntervalView (Класс пользовательского элемента диалога изменения временного интервала )** 207

**ViewModel.ViewModels.Modals.EditTimeIntervalViewModel (Класс диалога изменения временного интервала )** 208

**View.Extensions.EnumsExtension (Класс расширений синтаксиса разметки для перечислений )** 211

**MachineLearning.DistanceMetrics.EuclideanDistanceMetric (Класс метрики Евклидового расстояния )** 212

**ViewModel.Implementations.ModelLearning.ExecutionChanceTaskElementEvaluatorLearningController (Класс контроллера обучения модели обучения с учителем для предсказания шанса выполнения элементарной задачи )** 213

**ViewModel.Implementations.ModelLearning.Converters.ExecutionChanceTaskElementLearningConverter (Класс конвертора элементраных задач в данные для предсказания шанс выполнения с учителем и наоборот )** 215

**MachineLearning.ScoreMetrics.F1ScoreMetric (Класс метрики F1 оценки для модели обучения классификации )** 219

**ViewModel.Implementations.AppStates.FileLogger (Класс сервиса логгирования сообщений в файл )** 221

**ViewModel.Implementations.AppStates.FileService (Класс файлового сервиса )** 223

**View.Converters.FormattableToStringConverter (Класс конвертора форматируемого объекта в строку )** 227

**ViewModel.Interfaces.AppStates.IAppLifeState (Интерфейс для управления состоянием жизненного цикла приложения )** 228

**MachineLearning.Interfaces.IClassificationEvaluator (Интерфейс оценки модели обучения классификации )** 229

**MachineLearning.Interfaces.IClassificationModel (Интерфейс модели обучения классификации )** 230

**MachineLearning.Interfaces.IClassificationScoreMetric (Интерфейс метрики оценки для модели обучения классификации )** 231

**MachineLearning.Interfaces.IClusteringEvaluator (Интерфейс оценки модели обучения кластеризации )** 232

**MachineLearning.Interfaces.IClusteringModel (Интерфейс модели обучения кластеризации )** 233

**MachineLearning.Interfaces.IClusteringScoreMetric (Интерфейс метрики оценки для модели обучения кластеризации )** 234

**MachineLearning.Interfaces.Generals.IDataProcessor< T, R > (Интерфейс обработчика данных )** 235

**ViewModel.Interfaces.ModelLearning.IDataTransformer< T, R > (Интерфейс для преобразования данных из одного типа в другой )** 236

**ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.IDbContextFactory< T > (Интерфейс фабрики, создающая контексты базы данных )** 238

**MachineLearning.Interfaces.Generals.IDistanceMetric< T > (Интерфейс метрики расстояния )** 240

**ViewModel.Implementations.AppStates.Sessions.Database.Domains.IDomain (Индерфейс доменной модели )** 241

**ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T > (Интерфейс заменителя для изменения данных )** 242

**ViewModel.Interfaces.DataManagers.Generals.IEditorService (Интерфейс сервиса изменения )** 244

**ViewModel.Interfaces.DataManagers.Generals.IFactory< out T > (Интерфейс фабрики, создающая объекты )** 245

**ViewModel.Interfaces.AppStates.IFileService (Интерфейс сервиса файлов )** 247

**ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR > (Интерфейс контроллера обучения модели обучения )** 250

**MachineLearning.Interfaces.Generals.ILearningConverter< T, R, DT, DR > (Интерфейс конвертора данных в данные для предсказания и наоборот )** 252

**MachineLearning.Interfaces.Generals.ILearningEvaluator (Интерфейс оценки модели обучения )** 254

**MachineLearning.Interfaces.Generals.ILearningModel< T, R > (Интерфейс модели обучения )** 255

**ViewModel.Interfaces.AppStates.Settings.ILocalizationManager (Интерфейс менеджера локализаций )** 256

**ViewModel.Interfaces.AppStates.ILogger (Интерфейс сервиса логирования сообщений )** 257

**ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 > (Интерфейс для преобразования значений между двумя предметными областями )** 258

**ViewModel.Interfaces.ModelLearning.IModelTeacher< D > (Интерфейс для обучения модели на основе данных )** 260

**ViewModel.Interfaces.AppStates.INotificationManager (Интерфейс менеджера уведомлений )** 261

**MachineLearning.DataProcessors.InvalidValuesColumnProcessor (Класс обработчика столбцов для устранения некорректных значений )** 262

**View.iOS.IOsNotificationManager (Класс менеджера уведомлений iOS )** 264

**MachineLearning.Interfaces.IPointDataProcessor (Интерфейс обработчика точечных данных )** 265

**MachineLearning.Interfaces.IPointDistanceMetric (Интерфейс метрики расстояния для точечных данных )** 266

**MachineLearning.Interfaces.IPrimaryPointDataProcessor (Интерфейс первичного обработчика точечных данных )** 267

**ViewModel.Interfaces.DataManagers.Generals.IProxy< out T > (Интерфейс заместителя )** 268

**MachineLearning.Interfaces.IRegressionEvaluator (Интерфейс оценки модели обучения классификации )** 269

**MachineLearning.Interfaces.IRegressionModel (Интерфейс модели обучения регресии )** 270

**MachineLearning.Interfaces.IRegressionScoreMetric (Интерфейс метрики оценки для модели обучения регрессии )** 271

**ViewModel.Interfaces.AppStates.IResourceService (Интерфейс сервиса ресурсов )** 272

**MachineLearning.Interfaces.IScaler (Интерфейс масштабирования данных )** 273

**MachineLearning.Interfaces.Generals.IScoreMetric (Интерфейс метрики оценки )** 275

**ViewModel.Interfaces.AppStates.ISerializer (Интерфейс сериализатора )** 276

**ViewModel.Interfaces.AppStates.Sessions.ISession (Интерфейс сессии для хранения и изменения данных )** 278

**ViewModel.Interfaces.AppStates.Settings.ISettings (Интерфейс настроек )** 282

**ViewModel.Interfaces.AppStates.IStorageService (Интерфейс для сервиса хранилища )** 283

**ViewModel.Interfaces.ModelLearning.ISupervisedLearningController< T, R, D, DT, DR > (Интерфейс контроллера обучения модели обучения с учителем )** 285

**MachineLearning.Interfaces.Generals.ISupervisedLearningConverter< T, R, D, DT, DR > (Интерфейс конвертора данных в данные для предсказания с учителем и наоборот )** 287

**MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R > (Интерфейс оценки модели обучения с учителем )** 289

**MachineLearning.Interfaces.Generals.ISupervisedLearningModel< T, R > (Интерфейс модели обучения c учителем )** 291

**MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T > (Интерфейс метрики оценки для модели обучения с учителем )** 293

**Model.Interfaces.ITask (Интерфейс задачи )** 295

**Model.Interfaces.ITaskComposite (Интерфейс составной задачи )** 299

**Model.Interfaces.ITaskElement (Интерфейс элементарной задачи )** 301

**ViewModel.Interfaces.DataManagers.ITaskElementProxy (Интерфейс заместителя элементарной задачи )** 305

**ViewModel.Interfaces.DataManagers.ITaskElementsEditorProxy (Интерфейс заместителя элементарной задачи для редактирования )** 309

**ViewModel.Interfaces.DataManagers.ITasksEditorProxy (Интерфейс заместителя задачи для редактирования )** 311

**ViewModel.ViewModels.Modals.ItemsTasksViewModelArgs (Класс аргументов диалога элементов списка )** 313

**ViewModel.Interfaces.AppStates.Sessions.ItemsUpdatedEventArgs (Класс аргумента события при обновлении данных )** 315

**ViewModel.Interfaces.AppStates.Settings.IThemeManager (Интерфейс менеджера тем )** 317

**Model.Interfaces.ITimeInterval (Интерфейс временного интервала )** 318

**Model.Interfaces.ITimeIntervalElement (Интерфейс элементарного временного интервала )** 319

**ViewModel.Interfaces.DataManagers.ITimeIntervalElementsEditorProxy (Интерфейс заместителя элементарного временного интервала для редактирования )** 321

**Model.Interfaces.ITimeIntervalList (Интерфейс списка временных интервалов )** 323

**ViewModel.Interfaces.ITimeScheduler (Интерфейс планировщика времени, который отслеживает достижение временных меток )** 324

**ViewModel.Interfaces.ModelLearning.IUnsupervisedLearningController< T, R, D, DT, DR > (Интерфейс контроллера обучения модели обучения без учителя )** 325

**MachineLearning.Interfaces.Generals.IUnsupervisedLearningConverter< T, R, D, DT, DR > (Интерфейс конвертора данных в данные для предсказания без учителя и наоборот )** 327

**MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R > (Интерфейс оценки модели обучения без учителя )** 329

**MachineLearning.Interfaces.Generals.IUnsupervisedLearningModel< T, R > (Интерфейс модели обучение без учителя )** 331

**MachineLearning.Interfaces.Generals.IUnsupervisedScoreMetric< T, D > (Интерфейс метрики оценки для модели обучения без учителя )** 333

**ViewModel.Implementations.AppStates.JsonSerializer (Класс Json-сериализатора )** 335

**MachineLearning.LearningModels.KMeanLearningModel (Класс модель обучения кластеризации с алгоритмом K-средних )** 337

**MachineLearning.LearningModels.KNearestNeighborsModel (Класс модель обучения классификации с алгоритмом K-ближащих соседей )** 340

**MachineLearning.LearningModelData< T, R > (Структура хранения данных для модели обучения )** 343

**View.Desktop.Linux.LinuxNotificationManager (Класс менеджера уведомлений Linux )** 345

**MachineLearning.DataProcessors.LowVariationColumnProcessor (Класс обработчика столбцов для устранения низкой вариативности )** 346

**View.Desktop.macOS.MacOsNotificationManager (Класс менеджера уведомлений macOS )** 347

**View.Android.MainActivity (Класс основной активности )** 348

**View.Views.MainView (Класс основного пользовательского элемента приложения )** 350

**ViewModel.ViewModels.MainViewModel (Класс главного контроллера )** 353

**View.Views.MainWindow (Класс основного окна приложения )** 355

**ViewModel.Implementations.ModelLearning.Converters.MetadataCategoriesTransformer (Класс преобразования категории метаданных в данные для предсказания )** 356

**ViewModel.Implementations.AppStates.Sessions.Database.Entities.MetadataEntity (Класс сущности метаданных )** 358

**ViewModel.Implementations.AppStates.Sessions.Database.Mappers.MetadataMapper (Класс перобразования значений метаданных между двумя предметными областями )** 360

**ViewModel.Implementations.ModelLearning.Converters.MetadataTagsTransformer (Класс преобразования тегов метаданных в данные для предсказания )** 362

**MachineLearning.LearningModels.MetricAdapter (Адаптер для использования метрики расстояния IPointDistanceMetric с интерфейсом IMetric<double[]> )** 364

**MachineLearning.Scalers.MinMaxScaler (Класс масштабрирования данных с помощью минимума и максимума )** 365

**ViewModel.Implementations.Mocks.MockAppLifeState (Класс-заглушка управления состоянием жизненного цикла приложения )** 367

**ViewModel.Implementations.Mocks.MockLocalizationManager (Класс-заглушка менеджера локализаций )** 368

**ViewModel.Implementations.Mocks.MockNotificationManager (Класс-заглушка менеджера уведомлений )** 369

**ViewModel.Implementations.Mocks.MockResourceService (Класс-заглушка сервиса ресурсов )** 371

**ViewModel.Implementations.Mocks.MockThemeManager (Класс-заглушка менеджера тем )** 373

**View.Views.Modals.MoveTasksView (Класс пользовательского элемента диалога перемещения задач )** 374

**ViewModel.ViewModels.Modals.MoveTasksViewModel (Класс диалога перемещения задач )** 375

**MachineLearning.LearningModels.MultipleLinearRegressionModel (Класс модель обучения регрессии с алгоритмом мультилинейной регрессии )** 378

**ViewModel.Implementations.Mocks.NotificationEventArgs (Класс аргументов события уведомления )** 380

**MachineLearning.DataProcessors.OutlierRowProcessor (Класс обработчика строк для устранения выбросов )** 382

**View.DataTemplates.PageDataTemplate (Класс шаблона данных для страниц )** 383

**View.Technilcals.PageMetadata (Класс метаданных страницы )** 384

**ViewModel.Implementations.ModelLearning.PlannedRealTaskElementEvaluatorLearningController (Класс контроллера обучения модели обучения с учителем для предсказания запланированого реального показателя элементарной задачи )** 385

**ViewModel.Implementations.ModelLearning.Converters.PlannedRealTaskElementLearningConverter (Класс конвертора элементраных задач в данные для предсказания запланированного реального показателя с учителем и наоборот )** 387

**ViewModel.Implementations.ModelLearning.PlannedTimeTaskElementEvaluatorLearningController (Класс контроллера обучения модели обучения с учителем для предсказания запланированого времени элементарной задачи )** 391

**ViewModel.Implementations.ModelLearning.Converters.PlannedTimeTaskElementLearningConverter (Класс конвертора элементраных задач в данные для предсказания запланированного времени с учителем и наоборот )** 393

**View.Desktop.Linux.Program (Класс программы )** 397

**View.Desktop.macOS.Program (Класс программы )** 398

**View.Desktop.Windows.Program (Класс программы )** 399

**MachineLearning.LearningEvaluators.RegressionCrossValidationEvaluator (Класс оценки модели обучения регрессии методом кросс-валидации )** 400

**View.Views.Modals.RemoveTasksView (Класс пользовательского элемента диалога удаления задач )** 404

**ViewModel.ViewModels.Modals.RemoveTasksViewModel (Класс диалога удаления задач )** 405

**MachineLearning.ScoreMetrics.RSquaredScoreMetric (РљР»Р°СЃСЃ РјРµС‚СЂРёРєРё RВІ РѕС†РµРЅРєРё РґР»СЏ РјРѕРґРµР»Рё РѕР±СѓС‡РµРЅРёСЏ СЂРµРіСЂРµСЃСЃРёРё )** 407

**ViewModel.Implementations.ModelLearning.Converters.ScalerFactory (Класс фабрики, создающая масштабирования данных )** 409

**View.Views.Pages.SettingsView (Класс пользовательского элемента страницы настроек )** 410

**ViewModel.ViewModels.Pages.SettingsViewModel (Класс контроллера страницы настроек )** 411

**MachineLearning.ScoreMetrics.SilhouetteScoreMetric (Класс метрики оценки Силуэта для модели обучения кластеризации )** 413

**MachineLearning.ScoreMetrics.SmapeScoreMetric (Класс метрики оценки SMAPE для модели обучения регрессии )** 415

**ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.SqliteDbContext (Класс контекста базы данных SQLite )** 417

**ViewModel.Technicals.StatisticElement (Класс элемента статистики )** 419

**View.Converters.StatisticElementsToColumnSeriesConverter (Класс конвертора элементов статистики в столбцы столбчатой диаграммы )** 421

**View.Converters.StatisticElementsToPieSeriesConverter (Класс конвертора элементов статистики в сектора круговой диаграммы )** 422

**View.Views.Pages.StatisticView (Класс пользовательского элемента страницы статистики )** 423

**ViewModel.ViewModels.Pages.StatisticViewModel (Класс контроллера страницы статистики )** 424

**ViewModel.Implementations.Sessions.Database.Entities.TagEntity (Класс сущности тега )** 426

**View.Converters.TagsToStringConverter (Класс конвертора тегов в строку и наоборот )** 427

**Model.Tasks.TaskComposite (Класс составной задачи )** 428

**ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain (Класс домменной модели составной задачи )** 434

**ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskCompositeEntity (Класс сущности составной задачи )** 439

**ViewModel.Implementations.DataManagers.Factories.TaskCompositeFactory (Класс фабрики, создающая составные задачи )** 440

**ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskCompositeMapper (Класс перобразования значений составных задач между двумя предметными областями )** 442

**View.DataTemplates.TaskDataTemplate (Класс шаблона данных для задач )** 444

**Model.Tasks.TaskElement (Класс элементарной задачи )** 446

**ViewModel.Implementations.DataManagers.TaskElementCreatorProxy**  450

**ViewModel.Implementations.DataManagers.Factories.TaskElementCreatorProxyFactory (Класс фабрики, создающая заместителей элементарной задачи )** 457

**ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskElementDomain (Класс домменной модели элементарной задачи )** 459

**ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity (Класс сущности элементарной задачи )** 462

**ViewModel.Implementations.DataManagers.Factories.TaskElementFactory (Класс фабрики, создающая элементарные задачи )** 465

**ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskElementMapper (Класс перобразования значений элементарных задач между двумя предметными областями )** 467

**ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy (Класс заместитель элементарых задач для редактирования )** 469

**ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskEntity (Класс сущности задачи )** 474

**ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskMapper (Класс перобразования значений задач между двумя предметными областями )** 476

**ViewModel.Technicals.TaskMetadata (Класс метаданных задачи )** 478

**ViewModel.Implementations.DataManagers.Factories.TaskMetadataFactory (Класс фабрики, создающая метаданные задачи )** 481

**ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy (Класс заместитель задач для редактирования )** 482

**ViewModel.ViewModels.Modals.TasksViewModel< A, R > (Класс диалога задач )** 486

**ViewModel.ViewModels.Modals.TasksViewModelArgs (Класс аргументов диалога списка задач )** 489

**Model.Times.TimeIntervalElement (Класс элементарного временного интервала )** 491

**ViewModel.Implementations.AppStates.Sessions.Database.Domains.TimeIntervalElementDomain (Класс домменной модели элементарного временного интервала )** 494

**ViewModel.Implementations.DataManagers.Factories.TimeIntervalElementFactory (Класс фабрики, создающая элементарные временные интервалы )** 497

**ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy (Класс заместитель элементарного временного интервала для редактирования )** 498

**ViewModel.Implementations.AppStates.Sessions.Database.Entities.TimeIntervalEntity (Класс сущности временных интервалов )** 501

**Model.Times.TimeIntervalList (Класс списка временных интервалов )** 503

**ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TimeIntervalMapper (Класс перобразования значений временных интервалов между двумя предметными областями )** 507

**ViewModel.ViewModels.Modals.TimeIntervalViewModelArgs (Класс аргументов диалога AddTimeIntervalViewModel )** 508

**ViewModel.ViewModels.Modals.TimeIntervalViewModelResult (Класс результата диалога AddTimeIntervalViewModel )** 510

**ViewModel.Implementations.AppStates.TimeScheduler (Класс планировщика времени, который отслеживает достижение временных меток )** 512

**View.Views.Pages.TimeView (Класс пользовательского элемента страницы календаря )** 514

**ViewModel.ViewModels.Pages.TimeViewModel (Класс контроллера страницы календаря )** 515

**ViewModel.Technicals.ToDoListElement (Класс элемента списка задач для выполнения )** 517

**View.Views.Pages.ToDoListView (Класс пользовательского элемента страницы списка задач для выполнения )** 519

**ViewModel.ViewModels.Pages.ToDoListViewModel (Класс контроллера страницы списка задач для выполнения )** 520

**TrackableFeatures.TrackableCollection< T > (Базовый класс коллекции, предоставляющий поддержку отслеживания изменений в коллекции )** 522

**TrackableFeatures.TrackableObject (Базовый класс объекта, предоставляющий поддержку отслеживания изменений свойств и ошибок )** 529

**MachineLearning.LearningEvaluators.ValidationFold (Структура хранения данных сегмента валидации )** 534

**View.Desktop.Windows.WindowsNotificationManager (Класс менеджера уведомлений Windows )** 536

# Пространства имен

## Пространство имен MachineLearning

### Классы

* class **ArrayHelper**

*Вспомогательный статичный класс для преобразования в массивы и матрицы.*

struct **DataProcessorResult< T >***Структура хранения данных результата преобразования.*

* class **LearningHelper**

*Вспомогательный статичный класс для обучения.*

struct **LearningModelData< T, R >***Структура хранения данных для модели обучения.*

### Перечисления

* enum **ScoreMetricCategory** { **Excellent**, **Good**, **Satisfactory**, **Bad**, **Horrible** }

*Перечисление категорий метрики оценки.*

### Перечисления

#### enum MachineLearning.ScoreMetricCategory

Перечисление категорий метрики оценки.

##### Элементы перечислений:

|  |  |
| --- | --- |
| Excellent | Первосходно. |
| Good | Хорошо. |
| Satisfactory | Удовлетворительно. |
| Bad | Плохо. |
| Horrible | Непригодно. |

См. определение в файле **ScoreMetricCategory.cs** строка **6**

## Пространство имен MachineLearning.Converters

### Классы

class **BaseLearningConverter< R, D, DT, DR >***Абстрактный класс базового конвертора данных в данные для предсказания и наоборот.*

class **BaseSupervisedLearningConverter< R, D, DT, DR >***Абстрактный класс базового конвертора данных в данные для предсказания с учителем и наоборот.*

class **BaseUnsupervisedLearningConverter< R, D, DT, DR >***Абстрактный класс базового конвертора данных в данные для предсказания без учителя и наоборот.*

## Пространство имен MachineLearning.DataProcessors

### Классы

class **CorrelationColumnProcessor***Класс обработчика столбцов для устранения корреляции.*

class **DuplicatesRowProcessor***Класс обработчика строк для устранения дубликатов.*

class **InvalidValuesColumnProcessor***Класс обработчика столбцов для устранения некорректных значений.*

class **LowVariationColumnProcessor***Класс обработчика столбцов для устранения низкой вариативности.*

class **OutlierRowProcessor***Класс обработчика строк для устранения выбросов.*

## Пространство имен MachineLearning.DistanceMetrics

### Классы

class **EuclideanDistanceMetric***Класс метрики Евклидового расстояния.*

## Пространство имен MachineLearning.Interfaces

### Классы

interface **IClassificationEvaluator***Интерфейс оценки модели обучения классификации.*

interface **IClassificationModel***Интерфейс модели обучения классификации.*

interface **IClassificationScoreMetric***Интерфейс метрики оценки для модели обучения классификации.*

interface **IClusteringEvaluator***Интерфейс оценки модели обучения кластеризации.*

interface **IClusteringModel***Интерфейс модели обучения кластеризации.*

interface **IClusteringScoreMetric***Интерфейс метрики оценки для модели обучения кластеризации.*

interface **IPointDataProcessor***Интерфейс обработчика точечных данных.*

interface **IPointDistanceMetric***Интерфейс метрики расстояния для точечных данных.*

interface **IPrimaryPointDataProcessor***Интерфейс первичного обработчика точечных данных.*

interface **IRegressionEvaluator***Интерфейс оценки модели обучения классификации.*

interface **IRegressionModel***Интерфейс модели обучения регресии.*

interface **IRegressionScoreMetric***Интерфейс метрики оценки для модели обучения регрессии.*

interface **IScaler***Интерфейс масштабирования данных.*

## Пространство имен MachineLearning.Interfaces.Generals

### Классы

interface **IDataProcessor< T, R >***Интерфейс обработчика данных.*

interface **IDistanceMetric< T >***Интерфейс метрики расстояния.*

interface **ILearningConverter< T, R, DT, DR >***Интерфейс конвертора данных в данные для предсказания и наоборот.*

interface **ILearningEvaluator***Интерфейс оценки модели обучения.*

interface **ILearningModel< T, R >***Интерфейс модели обучения.*

interface **IScoreMetric***Интерфейс метрики оценки.*

interface **ISupervisedLearningConverter< T, R, D, DT, DR >***Интерфейс конвертора данных в данные для предсказания с учителем и наоборот.*

interface **ISupervisedLearningEvaluator< T, R >***Интерфейс оценки модели обучения с учителем.*

interface **ISupervisedLearningModel< T, R >***Интерфейс модели обучения c учителем.*

interface **ISupervisedScoreMetric< T >***Интерфейс метрики оценки для модели обучения с учителем.*

interface **IUnsupervisedLearningConverter< T, R, D, DT, DR >***Интерфейс конвертора данных в данные для предсказания без учителя и наоборот.*

interface **IUnsupervisedLearningEvaluator< T, R >***Интерфейс оценки модели обучения без учителя.*

interface **IUnsupervisedLearningModel< T, R >***Интерфейс модели обучение без учителя.*

interface **IUnsupervisedScoreMetric< T, D >***Интерфейс метрики оценки для модели обучения без учителя.*

## Пространство имен MachineLearning.LearningEvaluators

### Классы

class **BaseCrossValidationLearningEvaluator***Абстрактный класс базовой оценки модели обучения методом кросс-валидации.*

class **BaseSupervisedCrossValidationEvaluator< T, R >***Абстрактный класс базовой оценки модели обучения с учителем методом кросс-валидации.*

class **BaseUnsupervisedCrossValidationEvaluator< T, R >***Абстрактный класс базовой оценки модели обучения без учителя методом кросс-валидации.*

class **ClassificationCrossValidationEvaluator***Класс оценки модели обучения классификации методом кросс-валидации.*

class **ClusteringCrossValidationEvaluator***Класс оценки модели обучения кластеризации методом кросс-валидации.*

class **RegressionCrossValidationEvaluator***Класс оценки модели обучения регрессии методом кросс-валидации.*

struct **ValidationFold***Структура хранения данных сегмента валидации.*

## Пространство имен MachineLearning.LearningModels

### Классы

class **KMeanLearningModel***Класс модель обучения кластеризации с алгоритмом K-средних.*

class **KNearestNeighborsModel***Класс модель обучения классификации с алгоритмом K-ближащих соседей.*

class **MetricAdapter***Адаптер для использования метрики расстояния* ***IPointDistanceMetric*** *с интерфейсом IMetric<double[]>.*

class **MultipleLinearRegressionModel***Класс модель обучения регрессии с алгоритмом мультилинейной регрессии.*

## Пространство имен MachineLearning.Scalers

### Классы

class **MinMaxScaler***Класс масштабрирования данных с помощью минимума и максимума.*

## Пространство имен MachineLearning.ScoreMetrics

### Классы

class **AccuracyScoreMetric***Класс метрики оценки точности для модели обучения классификации.*

class **AdjustedRandIndexScoreMetric***Класс метрики оценки скорректированного индекса Рэнда для модели обучения классификации.*

class **F1ScoreMetric***Класс метрики F1 оценки для модели обучения классификации.*

class **RSquaredScoreMetric***РљР»Р°СЃСЃ РјРµС‚СЂРёРєРё RВІ РѕС†РµРЅРєРё РґР»СЏ РјРѕРґРµР»Рё РѕР±СѓС‡РµРЅРёСЏ СЂРµРіСЂРµСЃСЃРёРё.*

class **SilhouetteScoreMetric***Класс метрики оценки Силуэта для модели обучения кластеризации.*

class **SmapeScoreMetric***Класс метрики оценки SMAPE для модели обучения регрессии.*

## Пространство имен Model

### Классы

* class **TaskHelper**

*Вспомогательный статичный класс для работы с задачами.*

### Перечисления

* enum **TaskStatus** { **Cancelled**, **Blocked**, **Deferred**, **OnHold**, **InProgress**, **Planned**, **Closed** }

*Перечисление статусов задачи.*

### Перечисления

#### enum Model.TaskStatus

Перечисление статусов задачи.

##### Элементы перечислений:

|  |  |
| --- | --- |
| Cancelled | Отменена. |
| Blocked | Заблокирована. |
| Deferred | Отложена. |
| OnHold | На удержании. |
| InProgress | В процессе. |
| Planned | Запланирована. |
| Closed | Закрыта. |

См. определение в файле **TaskStatus.cs** строка **6**

## Пространство имен Model.Interfaces

### Классы

interface **ITask***Интерфейс задачи.*

interface **ITaskComposite***Интерфейс составной задачи.*

interface **ITaskElement***Интерфейс элементарной задачи.*

interface **ITimeInterval***Интерфейс временного интервала.*

interface **ITimeIntervalElement***Интерфейс элементарного временного интервала.*

interface **ITimeIntervalList***Интерфейс списка временных интервалов.*

## Пространство имен Model.Tasks

### Классы

class **TaskComposite***Класс составной задачи.*

class **TaskElement***Класс элементарной задачи.*

## Пространство имен Model.Times

### Классы

class **TimeIntervalElement***Класс элементарного временного интервала.*

class **TimeIntervalList***Класс списка временных интервалов.*

## Пространство имен TrackableFeatures

### Классы

class **TrackableCollection< T >***Базовый класс коллекции, предоставляющий поддержку отслеживания изменений в коллекции.*

class **TrackableObject***Базовый класс объекта, предоставляющий поддержку отслеживания изменений свойств и ошибок.*

## Пространство имен View

### Классы

class **App***Класс приложения.*

## Пространство имен View.Android

### Классы

class **AndroidNotificationManager***Класс менеджера уведомлений* ***Android****.*

class **MainActivity***Класс основной активности.*

* class **Resource**

## Пространство имен View.Controls

### Классы

class **DateTimePicker***Класс пользовательского элемента для выбора даты и времени.*

## Пространство имен View.Converters

### Классы

class **FormattableToStringConverter***Класс конвертора форматируемого объекта в строку.*

class **StatisticElementsToColumnSeriesConverter***Класс конвертора элементов статистики в столбцы столбчатой диаграммы.*

class **StatisticElementsToPieSeriesConverter***Класс конвертора элементов статистики в сектора круговой диаграммы.*

class **TagsToStringConverter***Класс конвертора тегов в строку и наоборот.*

## Пространство имен View.DataTemplates

### Классы

class **PageDataTemplate***Класс шаблона данных для страниц.*

class **TaskDataTemplate***Класс шаблона данных для задач.*

## Пространство имен View.Desktop

## Пространство имен View.Desktop.Linux

### Классы

class **DesktopAppLifeState***Класс для управления состоянием жизненного цикла desktop-приложения.*

class **LinuxNotificationManager***Класс менеджера уведомлений* ***Linux****.*

class **Program***Класс программы.*

## Пространство имен View.Desktop.macOS

### Классы

class **DesktopAppLifeState***Класс для управления состоянием жизненного цикла desktop-приложения.*

class **MacOsNotificationManager***Класс менеджера уведомлений* ***macOS****.*

class **Program***Класс программы.*

## Пространство имен View.Desktop.Windows

### Классы

class **DesktopAppLifeState***Класс для управления состоянием жизненного цикла desktop-приложения.*

class **Program***Класс программы.*

class **WindowsNotificationManager***Класс менеджера уведомлений* ***Windows****.*

## Пространство имен View.Extensions

### Классы

class **DesignExtension***Класс расширений синтаксиса разметки для дизайнера, который использует контейнер зависимости.*

class **EnumsExtension***Класс расширений синтаксиса разметки для перечислений.*

## Пространство имен View.Implementations

### Классы

class **AvaloniaLocalizationManager***Класс менеджера локализации Avalonia.*

class **AvaloniaResourceService***Класс сервис ресурсов Avalonia.*

class **AvaloniaThemeManager***Класс менеджер тем Avalonia.*

## Пространство имен View.iOS

### Классы

class **AppDelegate***Класс делегата приложения.*

class **Application***Класс приложения.*

class **IOsNotificationManager***Класс менеджера уведомлений* ***iOS****.*

## Пространство имен View.Technilcals

### Классы

class **PageMetadata***Класс метаданных страницы.*

* class **ViewContainerHelper**

*Вспомогательный статичный класс для работы с контейнером зависимостей в контексте* ***View****.*

## Пространство имен View.Views

### Классы

class **MainView***Класс основного пользовательского элемента приложения.*

class **MainWindow***Класс основного окна приложения.*

## Пространство имен View.Views.Modals

### Классы

class **AddTaskView***Класс пользовательского элемента диалога добавления задачи.*

class **AddTimeIntervalView***Класс пользовательского элемента диалога добавления временного интервала.*

class **CopyTasksView***Класс пользовательского элемента диалога копирования задач.*

class **EditTaskView***Класс пользовательского элемента диалога изменения задачи.*

class **EditTimeIntervalView***Класс пользовательского элемента диалога изменения временного интервала.*

class **MoveTasksView***Класс пользовательского элемента диалога перемещения задач.*

class **RemoveTasksView***Класс пользовательского элемента диалога удаления задач.*

## Пространство имен View.Views.Pages

### Классы

class **EditorView***Класс пользовательского элемента страницы изменения задач.*

class **SettingsView***Класс пользовательского элемента страницы настроек.*

class **StatisticView***Класс пользовательского элемента страницы статистики.*

class **TimeView***Класс пользовательского элемента страницы календаря.*

class **ToDoListView***Класс пользовательского элемента страницы списка задач для выполнения.*

## Пространство имен ViewModel

## Пространство имен ViewModel.Implementations

## Пространство имен ViewModel.Implementations.AppStates

### Классы

class **FileLogger***Класс сервиса логгирования сообщений в файл.*

class **FileService***Класс файлового сервиса.*

class **JsonSerializer***Класс Json-сериализатора.*

class **TimeScheduler***Класс планировщика времени, который отслеживает достижение временных меток.*

## Пространство имен ViewModel.Implementations.AppStates.Sessions

### Классы

class **DbSession***Класс сессии базы данных.*

## Пространство имен ViewModel.Implementations.AppStates.Sessions.Database

## Пространство имен ViewModel.Implementations.AppStates.Sessions.Database.DbContexts

### Классы

class **BaseDbContext***Базовый класс контекста базы данных.*

class **DbContextFactory***Класс фабрики, создающая контексты базы данных.*

interface **IDbContextFactory< T >***Интерфейс фабрики, создающая контексты базы данных.*

class **SqliteDbContext***Класс контекста базы данных SQLite.*

## Пространство имен ViewModel.Implementations.AppStates.Sessions.Database.Domains

### Классы

interface **IDomain***Индерфейс доменной модели.*

class **TaskCompositeDomain***Класс домменной модели составной задачи.*

class **TaskElementDomain***Класс домменной модели элементарной задачи.*

class **TimeIntervalElementDomain***Класс домменной модели элементарного временного интервала.*

## Пространство имен ViewModel.Implementations.AppStates.Sessions.Database.Entities

### Классы

class **MetadataEntity***Класс сущности метаданных.*

class **TaskCompositeEntity***Класс сущности составной задачи.*

class **TaskElementEntity***Класс сущности элементарной задачи.*

class **TaskEntity***Класс сущности задачи.*

class **TimeIntervalEntity***Класс сущности временных интервалов.*

## Пространство имен ViewModel.Implementations.AppStates.Sessions.Database.Mappers

### Классы

interface **IMapper< T1, T2 >***Интерфейс для преобразования значений между двумя предметными областями.*

class **MetadataMapper***Класс перобразования значений метаданных между двумя предметными областями.*

class **TaskCompositeMapper***Класс перобразования значений составных задач между двумя предметными областями.*

class **TaskElementMapper***Класс перобразования значений элементарных задач между двумя предметными областями.*

class **TaskMapper***Класс перобразования значений задач между двумя предметными областями.*

class **TimeIntervalMapper***Класс перобразования значений временных интервалов между двумя предметными областями.*

## Пространство имен ViewModel.Implementations.AppStates.Settings

### Классы

class **AppConfigurationFormat***Класс формата конфигурации приложении.*

class **AppSettings***Класс настроек приложения.*

## Пространство имен ViewModel.Implementations.DataManagers

### Классы

class **TaskElementCreatorProxy**

## Пространство имен ViewModel.Implementations.DataManagers.Editors

### Классы

class **TaskElementsEditorProxy***Класс заместитель элементарых задач для редактирования.*

class **TasksEditorProxy***Класс заместитель задач для редактирования.*

class **TimeIntervalElementsEditorProxy***Класс заместитель элементарного временного интервала для редактирования.*

## Пространство имен ViewModel.Implementations.DataManagers.Factories

### Классы

class **TaskCompositeFactory***Класс фабрики, создающая составные задачи.*

class **TaskElementCreatorProxyFactory***Класс фабрики, создающая заместителей элементарной задачи.*

class **TaskElementFactory***Класс фабрики, создающая элементарные задачи.*

class **TaskMetadataFactory***Класс фабрики, создающая метаданные задачи.*

class **TimeIntervalElementFactory***Класс фабрики, создающая элементарные временные интервалы.*

## Пространство имен ViewModel.Implementations.Mocks

### Классы

class **MockAppLifeState***Класс-заглушка управления состоянием жизненного цикла приложения.*

class **MockLocalizationManager***Класс-заглушка менеджера локализаций.*

class **MockNotificationManager***Класс-заглушка менеджера уведомлений.*

class **MockResourceService***Класс-заглушка сервиса ресурсов.*

class **MockThemeManager***Класс-заглушка менеджера тем.*

class **NotificationEventArgs***Класс аргументов события уведомления.*

## Пространство имен ViewModel.Implementations.ModelLearning

### Классы

class **BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >***Абстрактный класс базового контроллера обучения модели обучения с учителем.*

class **DeadlineTaskElementEvaluatorLearningController***Класс контроллера обучения модели обучения с учителем для предсказания срока элементарной задачи.*

class **ExecutionChanceTaskElementEvaluatorLearningController***Класс контроллера обучения модели обучения с учителем для предсказания шанса выполнения элементарной задачи.*

class **PlannedRealTaskElementEvaluatorLearningController***Класс контроллера обучения модели обучения с учителем для предсказания запланированого реального показателя элементарной задачи.*

class **PlannedTimeTaskElementEvaluatorLearningController***Класс контроллера обучения модели обучения с учителем для предсказания запланированого времени элементарной задачи.*

## Пространство имен ViewModel.Implementations.ModelLearning.Converters

### Классы

class **BaseTaskElementSupervisedLearningConverter< R, DR >***Абстрактный класс базового конвертора элементраных задач в данные для предсказания с учителем и наоборот.*

class **DeadlineTaskElementLearningConverter***Класс конвертора элементраных задач в данные для предсказания срока с учителем и наоборот.*

class **ExecutionChanceTaskElementLearningConverter***Класс конвертора элементраных задач в данные для предсказания шанс выполнения с учителем и наоборот.*

class **MetadataCategoriesTransformer***Класс преобразования категории метаданных в данные для предсказания.*

class **MetadataTagsTransformer***Класс преобразования тегов метаданных в данные для предсказания.*

class **PlannedRealTaskElementLearningConverter***Класс конвертора элементраных задач в данные для предсказания запланированного реального показателя с учителем и наоборот.*

class **PlannedTimeTaskElementLearningConverter***Класс конвертора элементраных задач в данные для предсказания запланированного времени с учителем и наоборот.*

class **ScalerFactory***Класс фабрики, создающая масштабирования данных.*

## Пространство имен ViewModel.Implementations.Sessions

## Пространство имен ViewModel.Implementations.Sessions.Database

## Пространство имен ViewModel.Implementations.Sessions.Database.Entities

### Классы

class **TagEntity***Класс сущности тега.*

## Пространство имен ViewModel.Interfaces

### Классы

interface **ITimeScheduler***Интерфейс планировщика времени, который отслеживает достижение временных меток.*

## Пространство имен ViewModel.Interfaces.AppStates

### Классы

interface **IAppLifeState***Интерфейс для управления состоянием жизненного цикла приложения.*

interface **IFileService***Интерфейс сервиса файлов.*

interface **ILogger***Интерфейс сервиса логирования сообщений.*

interface **INotificationManager***Интерфейс менеджера уведомлений.*

interface **IResourceService***Интерфейс сервиса ресурсов.*

interface **ISerializer***Интерфейс сериализатора.*

interface **IStorageService***Интерфейс для сервиса хранилища.*

## Пространство имен ViewModel.Interfaces.AppStates.Sessions

### Классы

interface **ISession***Интерфейс сессии для хранения и изменения данных.*

class **ItemsUpdatedEventArgs***Класс аргумента события при обновлении данных.*

### Перечисления

* enum **UpdateItemsState** { **Reset**, **Add**, **Edit**, **Remove**, **Move** }

*Перечисление состояний обновлений объектов.*

### Перечисления

#### enum ViewModel.Interfaces.AppStates.Sessions.UpdateItemsState

Перечисление состояний обновлений объектов.

##### Элементы перечислений:

|  |  |
| --- | --- |
| Reset | Перезагружено. |
| Add | Добавлено. |
| Edit | Изменено. |
| Remove | Удалено. |
| Move | Перемещено. |

См. определение в файле **UpdateItemsState.cs** строка **6**

## Пространство имен ViewModel.Interfaces.AppStates.Settings

### Классы

interface **ILocalizationManager***Интерфейс менеджера локализаций.*

interface **ISettings***Интерфейс настроек.*

interface **IThemeManager***Интерфейс менеджера тем.*

## Пространство имен ViewModel.Interfaces.DataManagers

### Классы

interface **ITaskElementProxy***Интерфейс заместителя элементарной задачи.*

interface **ITaskElementsEditorProxy***Интерфейс заместителя элементарной задачи для редактирования.*

interface **ITasksEditorProxy***Интерфейс заместителя задачи для редактирования.*

interface **ITimeIntervalElementsEditorProxy***Интерфейс заместителя элементарного временного интервала для редактирования.*

## Пространство имен ViewModel.Interfaces.DataManagers.Generals

### Классы

interface **IEditorProxy< T >***Интерфейс заменителя для изменения данных.*

interface **IEditorService***Интерфейс сервиса изменения.*

interface **IFactory< out T >***Интерфейс фабрики, создающая объекты.*

interface **IProxy< out T >***Интерфейс заместителя.*

## Пространство имен ViewModel.Interfaces.ModelLearning

### Классы

interface **IDataTransformer< T, R >***Интерфейс для преобразования данных из одного типа в другой.*

interface **ILearningController< D, DT, DR >***Интерфейс контроллера обучения модели обучения.*

interface **IModelTeacher< D >***Интерфейс для обучения модели на основе данных.*

interface **ISupervisedLearningController< T, R, D, DT, DR >***Интерфейс контроллера обучения модели обучения с учителем.*

interface **IUnsupervisedLearningController< T, R, D, DT, DR >***Интерфейс контроллера обучения модели обучения без учителя.*

## Пространство имен ViewModel.Technicals

### Классы

class **AppConfiguration***Класс конфигурации приложения.*

class **CalendarInterval***Класс интервала календаря.*

class **StatisticElement***Класс элемента статистики.*

class **TaskMetadata***Класс метаданных задачи.*

class **ToDoListElement***Класс элемента списка задач для выполнения.*

* class **ViewModelContainerHelper**

*Вспомогательный статичный класс для работы с контейнером зависимостей в контексте* ***ViewModel****.*

## Пространство имен ViewModel.ViewModels

### Классы

class **BaseDialogViewModel< A, R >***Базовый абстрактный класс контроллера диалога.*

class **BasePageViewModel***Базовый класс контроллера страницы.*

class **BaseViewModel***Базовый класс для контроллеров.*

class **MainViewModel***Класс главного контроллера.*

## Пространство имен ViewModel.ViewModels.Modals

### Классы

class **AddTaskViewModel***Класс диалога добавления задачи.*

class **AddTimeIntervalViewModel***Класс диалога добавления временного интервала.*

class **CopyTasksViewModel***Класс диалога копирования задач.*

class **CopyTasksViewModelResult***Класс результата диалога* ***CopyTasksViewModel****.*

class **EditTaskViewModel***Класс диалога изменения задачи.*

class **EditTimeIntervalViewModel***Класс диалога изменения временного интервала.*

class **ItemsTasksViewModelArgs***Класс аргументов диалога элементов списка.*

class **MoveTasksViewModel***Класс диалога перемещения задач.*

class **RemoveTasksViewModel***Класс диалога удаления задач.*

class **TasksViewModel< A, R >***Класс диалога задач.*

class **TasksViewModelArgs***Класс аргументов диалога списка задач.*

class **TimeIntervalViewModelArgs***Класс аргументов диалога* ***AddTimeIntervalViewModel****.*

class **TimeIntervalViewModelResult***Класс результата диалога* ***AddTimeIntervalViewModel****.*

## Пространство имен ViewModel.ViewModels.Pages

### Классы

class **EditorViewModel***Класс контроллера страницы изменения задач.*

class **SettingsViewModel***Класс контроллера страницы настроек.*

class **StatisticViewModel***Класс контроллера страницы статистики.*

class **TimeViewModel***Класс контроллера страницы календаря.*

class **ToDoListViewModel***Класс контроллера страницы списка задач для выполнения.*

# Классы

## Класс MachineLearning.ScoreMetrics.AccuracyScoreMetric

Класс метрики оценки точности для модели обучения классификации.

Граф наследования:MachineLearning.ScoreMetrics.AccuracyScoreMetric:

### Открытые члены

* double **CalculateScore** (IEnumerable< int > actual, IEnumerable< int > predicted)
* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

##### *Аргументы*

|  |  |
| --- | --- |
| *score* | Оценка. |

##### *Возвращает*

Возвращает категорию оценки.

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T >

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< T > predicted)

*Вычисляет оценку на основе фактических и предсказанных данных.*

### Подробное описание

Класс метрики оценки точности для модели обучения классификации.

Реализует **IClassificationScoreMetric**.

См. определение в файле **AccuracityScoreMetric.cs** строка **11**

### Методы

#### double MachineLearning.ScoreMetrics.AccuracyScoreMetric.CalculateScore (IEnumerable< int > actual, IEnumerable< int > predicted)

См. определение в файле **AccuracityScoreMetric.cs** строка **14**

#### ScoreMetricCategory MachineLearning.ScoreMetrics.AccuracyScoreMetric.GetScoreCategory (double score)

Определяет категорию оценки.

##### Аргументы

|  |  |
| --- | --- |
| *score* | Оценка. |

##### Возвращает

Возвращает категорию оценки.

Замещает **MachineLearning.Interfaces.Generals.IScoreMetric** (*стр.275*).

## Класс View.Views.Modals.AddTaskView

Класс пользовательского элемента диалога добавления задачи.

Граф наследования:View.Views.Modals.AddTaskView:

### Открытые члены

* **AddTaskView** ()

*Создаёт экземпляр класса* ***AddTaskView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента диалога добавления задачи.

Наследует ReactiveUserControl<AddTaskViewModel>.

См. определение в файле **AddTaskView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Modals.AddTaskView.AddTaskView ()

Создаёт экземпляр класса **AddTaskView** по умолчанию.

См. определение в файле **AddTaskView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Modals.AddTaskViewModel

Класс диалога добавления задачи.

Граф наследования:ViewModel.ViewModels.Modals.AddTaskViewModel:

### Открытые члены

* **AddTaskViewModel** ()

*Создаёт экземпляр класса* ***AddTaskViewModel*** *по умолчанию.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* async Task< R > **Invoke** (**BaseViewModel** parent, A args)

*Вызывает диалог.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Защищенные члены

* override void **GetArgs** (**ITask** args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* void **GetArgs** (A args)

*Получает аргументы, переданные в диалог.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* TaskCompletionSource< R >? **\_taskSource**

*Источник для завершения задачи, который используется для возврата результата диалога.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* **BaseViewModel**? **Parent** [get]

*Возвращает родителя.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс диалога добавления задачи.

Наследует **BaseDialogViewModel<ITask, bool>**.

См. определение в файле **AddTaskViewModel.cs** строка **16**

### Конструктор(ы)

#### ViewModel.ViewModels.Modals.AddTaskViewModel.AddTaskViewModel ()

Создаёт экземпляр класса **AddTaskViewModel** по умолчанию.

См. определение в файле **AddTaskViewModel.cs** строка **32**

## Класс View.Views.Modals.AddTimeIntervalView

Класс пользовательского элемента диалога добавления временного интервала.

Граф наследования:View.Views.Modals.AddTimeIntervalView:

### Открытые члены

* **AddTimeIntervalView** ()

*Создаёт экземпляр класса* ***AddTimeIntervalView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента диалога добавления временного интервала.

Наследует ReactiveUserControl<AddTimeIntervalViewModel>.

См. определение в файле **AddTimeIntervalView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Modals.AddTimeIntervalView.AddTimeIntervalView ()

Создаёт экземпляр класса **AddTimeIntervalView** по умолчанию.

См. определение в файле **AddTimeIntervalView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Modals.AddTimeIntervalViewModel

Класс диалога добавления временного интервала.

Граф наследования:ViewModel.ViewModels.Modals.AddTimeIntervalViewModel:

### Открытые члены

* **AddTimeIntervalViewModel** ()

*Создаёт экземпляр класса* ***AddTimeIntervalViewModel*** *по умолчанию.*

#### Открытые члены унаследованные от ViewModel.ViewModels.Modals.TasksViewModel< A, R >

* **TasksViewModel** ()

*Создаёт экземпляр класса* ***TasksViewModel<A, R>*** *по умолчанию.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* async Task< R > **Invoke** (**BaseViewModel** parent, A args)

*Вызывает диалог.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Защищенные члены

* override void **GetArgs** (**TimeIntervalViewModelArgs** args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.Modals.TasksViewModel< A, R >

* override void **GetArgs** (A args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* void **GetArgs** (A args)

*Получает аргументы, переданные в диалог.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.Modals.TasksViewModel< A, R >

* IObservable< bool > **\_canExecuteGoToPrevious**

*Наблюдатель, который отслеживает возможность выполнения GoToPrevious.*

* IObservable< bool > **\_canExecuteGo**

*Наблюдатель, который отслеживает возможность выполнения Go.*

* IEnumerable< **ITask** > **\_mainList**

*Основной список.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* TaskCompletionSource< R >? **\_taskSource**

*Источник для завершения задачи, который используется для возврата результата диалога.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* **BaseViewModel**? **Parent** [get]

*Возвращает родителя.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс диалога добавления временного интервала.

Наследует **BaseDialogViewModel<TimeIntervalViewModelArgs,TimeIntervalViewModelResult>**.

См. определение в файле **AddTimeIntervalModel.cs** строка **17**

### Конструктор(ы)

#### ViewModel.ViewModels.Modals.AddTimeIntervalViewModel.AddTimeIntervalViewModel ()

Создаёт экземпляр класса **AddTimeIntervalViewModel** по умолчанию.

См. определение в файле **AddTimeIntervalModel.cs** строка **40**

### Методы

#### override void ViewModel.ViewModels.Modals.AddTimeIntervalViewModel.GetArgs (TimeIntervalViewModelArgs args)[protected]

См. определение в файле **AddTimeIntervalModel.cs** строка **57**

## Класс MachineLearning.ScoreMetrics.AdjustedRandIndexScoreMetric

Класс метрики оценки скорректированного индекса Рэнда для модели обучения классификации.

Граф наследования:MachineLearning.ScoreMetrics.AdjustedRandIndexScoreMetric:

### Открытые члены

* double **CalculateScore** (IEnumerable< int > actual, IEnumerable< int > predicted)
* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

##### *Аргументы*

|  |  |
| --- | --- |
| *score* | Оценка. |

##### *Возвращает*

Возвращает категорию оценки.

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T >

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< T > predicted)

*Вычисляет оценку на основе фактических и предсказанных данных.*

### Подробное описание

Класс метрики оценки скорректированного индекса Рэнда для модели обучения классификации.

Реализует **IClassificationScoreMetric**.

См. определение в файле **AdjustedRandIndexScoreMetric.cs** строка **13**

### Методы

#### double MachineLearning.ScoreMetrics.AdjustedRandIndexScoreMetric.CalculateScore (IEnumerable< int > actual, IEnumerable< int > predicted)

См. определение в файле **AdjustedRandIndexScoreMetric.cs** строка **16**

#### ScoreMetricCategory MachineLearning.ScoreMetrics.AdjustedRandIndexScoreMetric.GetScoreCategory (double score)

Определяет категорию оценки.

##### Аргументы

|  |  |
| --- | --- |
| *score* | Оценка. |

##### Возвращает

Возвращает категорию оценки.

Замещает **MachineLearning.Interfaces.Generals.IScoreMetric** (*стр.275*).

## Класс View.Android.AndroidNotificationManager

Класс менеджера уведомлений **Android**.

Граф наследования:View.Android.AndroidNotificationManager:

### Открытые члены

* **AndroidNotificationManager** ()

*Создаёт экземпляр класса* ***AndroidNotificationManager*** *по умолчанию.*

* void **SendNotification** (string description, string title)

*Отправляет уведомление.*

##### *Аргументы*

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

### Подробное описание

Класс менеджера уведомлений **Android**.

Реализует **INotificationManager**.

См. определение в файле **AndroidNotificationManager.cs** строка **15**

### Конструктор(ы)

#### View.Android.AndroidNotificationManager.AndroidNotificationManager ()

Создаёт экземпляр класса **AndroidNotificationManager** по умолчанию.

См. определение в файле **AndroidNotificationManager.cs** строка **30**

### Методы

#### void View.Android.AndroidNotificationManager.SendNotification (string description, string title)

Отправляет уведомление.

##### Аргументы

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

Замещает **ViewModel.Interfaces.AppStates.INotificationManager** (*стр.261*).

См. определение в файле **AndroidNotificationManager.cs** строка **36**

## Класс View.App

Класс приложения.

Граф наследования:View.App:

### Открытые члены

* override void **Initialize** ()
* override void **OnFrameworkInitializationCompleted** ()

### События

* EventHandler< ContainerBuilder >? **ContainerBuilderCreated**

*Событие, возникающее, когда конфигуратор контейнера зависимостей создан.*

### Подробное описание

Класс приложения.

Наследует Application.

См. определение в файле **App.axaml.cs** строка **18**

### Методы

#### override void View.App.Initialize ()

См. определение в файле **App.axaml.cs** строка **26**

#### override void View.App.OnFrameworkInitializationCompleted ()

См. определение в файле **App.axaml.cs** строка **32**

### Cобытия

#### EventHandler<ContainerBuilder>? View.App.ContainerBuilderCreated

Событие, возникающее, когда конфигуратор контейнера зависимостей создан.

См. определение в файле **App.axaml.cs** строка **23**

## Класс ViewModel.Technicals.AppConfiguration

Класс конфигурации приложения.

Граф наследования:ViewModel.Technicals.AppConfiguration:

### Свойства

* CultureInfo **ActualLocalization** [get, set]

*Возвращает и задаёт актуальную локализацию.*

* IEnumerable< CultureInfo > **Localizations** [get, set]

*Возвращает и задаёт локализации.*

* object **ActualTheme** [get, set]

*Возвращает и задаёт актуальную тему.*

* IEnumerable< object > **Themes** = Enumerable.Empty<object>() [get, set]

*Возвращает и задаёт темы.*

* string **SavePath** [get, set]

*Возвращает и задаёт путь сохранения.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс конфигурации приложения.

Наследует **TrackableObject**.

См. определение в файле **AppConfiguration.cs** строка **13**

### Полный список свойств

#### CultureInfo ViewModel.Technicals.AppConfiguration.ActualLocalization[get], [set]

Возвращает и задаёт актуальную локализацию.

См. определение в файле **AppConfiguration.cs** строка **33**

#### object ViewModel.Technicals.AppConfiguration.ActualTheme[get], [set]

Возвращает и задаёт актуальную тему.

См. определение в файле **AppConfiguration.cs** строка **48**

#### IEnumerable<CultureInfo> ViewModel.Technicals.AppConfiguration.Localizations[get], [set]

**Инициализатор**

=

Enumerable.Empty<CultureInfo>()

Возвращает и задаёт локализации.

См. определение в файле **AppConfiguration.cs** строка **42**

#### string ViewModel.Technicals.AppConfiguration.SavePath[get], [set]

Возвращает и задаёт путь сохранения.

См. определение в файле **AppConfiguration.cs** строка **62**

#### IEnumerable<object> ViewModel.Technicals.AppConfiguration.Themes = Enumerable.Empty<object>()[get], [set]

Возвращает и задаёт темы.

См. определение в файле **AppConfiguration.cs** строка **57**

## Класс ViewModel.Implementations.AppStates.Settings.AppConfigurationFormat

Класс формата конфигурации приложении.

### Свойства

* CultureInfo? **Localization** [get, set]

*Возвращает и задаёт локализацию.*

* string? **Theme** [get, set]

*Возвращает и задаёт тему.*

* string? **SavePath** [get, set]

*Возвращает и задаёт путь к сохранению.*

### Подробное описание

Класс формата конфигурации приложении.

См. определение в файле **AppConfigurationFormat.cs** строка **8**

### Полный список свойств

#### CultureInfo? ViewModel.Implementations.AppStates.Settings.AppConfigurationFormat.Localization[get], [set]

Возвращает и задаёт локализацию.

См. определение в файле **AppConfigurationFormat.cs** строка **13**

#### string? ViewModel.Implementations.AppStates.Settings.AppConfigurationFormat.SavePath[get], [set]

Возвращает и задаёт путь к сохранению.

См. определение в файле **AppConfigurationFormat.cs** строка **23**

#### string? ViewModel.Implementations.AppStates.Settings.AppConfigurationFormat.Theme[get], [set]

Возвращает и задаёт тему.

См. определение в файле **AppConfigurationFormat.cs** строка **18**

## Класс View.iOS.AppDelegate

Класс делегата приложения.

Граф наследования:View.iOS.AppDelegate:

### Открытые члены

* **AppDelegate** ()

*Создаёт экземпляр класса*  ***AppDelegate*** *по умолчанию.*

### Защищенные члены

* override AppBuilder **CustomizeAppBuilder** (AppBuilder builder)
* override AppBuilder **CreateAppBuilder** ()

### События

* EventHandler **AppDeactivated**

#### События унаследованные от ViewModel.Interfaces.AppStates.IAppLifeState

* EventHandler **AppDeactivated**

*Событие, которое возникает, когда приложение деактивируется.*

### Подробное описание

Класс делегата приложения.

Наследует AvaloniaAppDelegate<App>. Реализует **IAppLifeState**.

См. определение в файле **AppDelegate.cs** строка **24**

### Конструктор(ы)

#### View.iOS.AppDelegate.AppDelegate ()

Создаёт экземпляр класса  **AppDelegate** по умолчанию.

См. определение в файле **AppDelegate.cs** строка **32**

### Методы

#### override AppBuilder View.iOS.AppDelegate.CreateAppBuilder ()[protected]

См. определение в файле **AppDelegate.cs** строка **46**

#### override AppBuilder View.iOS.AppDelegate.CustomizeAppBuilder (AppBuilder builder)[protected]

См. определение в файле **AppDelegate.cs** строка **38**

### Cобытия

#### EventHandler View.iOS.AppDelegate.AppDeactivated

См. определение в файле **AppDelegate.cs** строка **27**

## Класс View.iOS.Application

Класс приложения.

### Подробное описание

Класс приложения.

См. определение в файле **Main.cs** строка **8**

## Класс ViewModel.Implementations.AppStates.Settings.AppSettings

Класс настроек приложения.

Граф наследования:ViewModel.Implementations.AppStates.Settings.AppSettings:

### Открытые члены

* **AppSettings** (**IThemeManager** themeManager, **ILocalizationManager** localizationManager, **ISession** session, **IFileService** fileService, **ISerializer** serializer, **ILogger** logger)

*Создаёт экземпляр класса* ***AppSettings****.*

* async Task **Save** ()

*Сохраняет данные в хранилище.*

##### *Возвращает*

Возвращает задачц процесса сохранения данных.

* async Task **Load** ()

*Загружает данные из хранилища.*

##### *Возвращает*

Возвращает задачц процесса загрузки данных.

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* void **InitializeConfiguration** ()

*Инициализирует конфигурацию.*

* void **OnConfigurationUpdated** (**AppConfiguration** oldConfiguration, **AppConfiguration** newConfiguration)

*Вызывается при обновлении конфигурации.*

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Свойства

* string **FilePath** [get, set]

*Возвращает и задаёт путь к файлу.*

* object **Configuration** [get]

*Возвращает конфигурацию.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс настроек приложения.

Наследует **TrackableObject**. Реализует **ISettings**.

См. определение в файле **AppSettings.cs** строка **19**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.Settings.AppSettings.AppSettings (IThemeManager themeManager, ILocalizationManager localizationManager, ISession session, IFileService fileService, ISerializer serializer, ILogger logger)

Создаёт экземпляр класса **AppSettings**.

##### Аргументы

|  |  |
| --- | --- |
| *themeManager* | Менеджер тем. |
| *localizationManager* | Менеджер локализаций. |
| *session* | Сессия. |
| *fileService* | Файловый сервис. |
| *serializer* | Сериализатор. |
| *logger* | Логгирование. |

См. определение в файле **AppSettings.cs** строка **83**

### Методы

#### void ViewModel.Implementations.AppStates.Settings.AppSettings.InitializeConfiguration ()[protected]

Инициализирует конфигурацию.

См. определение в файле **AppSettings.cs** строка **158**

#### async Task ViewModel.Implementations.AppStates.Settings.AppSettings.Load ()

Загружает данные из хранилища.

##### Возвращает

Возвращает задачц процесса загрузки данных.

Замещает **ViewModel.Interfaces.AppStates.IStorageService** (*стр.283*).

См. определение в файле **AppSettings.cs** строка **125**

#### void ViewModel.Implementations.AppStates.Settings.AppSettings.OnConfigurationUpdated (AppConfiguration oldConfiguration, AppConfiguration newConfiguration)[protected]

Вызывается при обновлении конфигурации.

##### Аргументы

|  |  |
| --- | --- |
| *oldConfiguration* | Старая конфигурация. |
| *newConfiguration* | Новая конфигурация. |

См. определение в файле **AppSettings.cs** строка **177**

#### async Task ViewModel.Implementations.AppStates.Settings.AppSettings.Save ()

Сохраняет данные в хранилище.

##### Возвращает

Возвращает задачц процесса сохранения данных.

Замещает **ViewModel.Interfaces.AppStates.IStorageService** (*стр.283*).

См. определение в файле **AppSettings.cs** строка **100**

### Полный список свойств

#### object ViewModel.Implementations.AppStates.Settings.AppSettings.Configuration[get]

Возвращает конфигурацию.

Замещает **ViewModel.Interfaces.AppStates.Settings.ISettings** (*стр.282*).

См. определение в файле **AppSettings.cs** строка **67**

#### string ViewModel.Implementations.AppStates.Settings.AppSettings.FilePath[get], [set]

Возвращает и задаёт путь к файлу.

См. определение в файле **AppSettings.cs** строка **64**

## Класс View.Implementations.AvaloniaLocalizationManager

Класс менеджера локализации Avalonia.

Граф наследования:View.Implementations.AvaloniaLocalizationManager:

### Открытые члены

* **AvaloniaLocalizationManager** ()

*Создаёт экземпляр класса* ***AvaloniaLocalizationManager*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Свойства

* IEnumerable< CultureInfo > **Localizations** [get]

*Возвращает локализации.*

* CultureInfo **ActualLocalization** [get, set]

*Возвращает и задаёт актуальную локализацию.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс менеджера локализации Avalonia.

Наследует **TrackableObject**. Реализует **ILocalizationManager**.

См. определение в файле **AvaloniaLocalizationManager.cs** строка **22**

### Конструктор(ы)

#### View.Implementations.AvaloniaLocalizationManager.AvaloniaLocalizationManager ()

Создаёт экземпляр класса **AvaloniaLocalizationManager** по умолчанию.

См. определение в файле **AvaloniaLocalizationManager.cs** строка **56**

### Полный список свойств

#### CultureInfo View.Implementations.AvaloniaLocalizationManager.ActualLocalization[get], [set]

Возвращает и задаёт актуальную локализацию.

Замещает **ViewModel.Interfaces.AppStates.Settings.ILocalizationManager** (*стр.256*).

См. определение в файле **AvaloniaLocalizationManager.cs** строка **47**

#### IEnumerable<CultureInfo> View.Implementations.AvaloniaLocalizationManager.Localizations[get]

Возвращает локализации.

Замещает **ViewModel.Interfaces.AppStates.Settings.ILocalizationManager** (*стр.256*).

См. определение в файле **AvaloniaLocalizationManager.cs** строка **44**

## Класс View.Implementations.AvaloniaResourceService

Класс сервис ресурсов Avalonia.

Граф наследования:View.Implementations.AvaloniaResourceService:

### Открытые члены

* **AvaloniaResourceService** ()

*Создаёт экземпляр класса* ***AvaloniaResourceService*** *по умолчанию.*

* object? **GetResource** (object key)

*Возвращает ресурс по ключу.*

##### *Аргументы*

|  |  |
| --- | --- |
| *key* | Ключ. |

##### *Возвращает*

Возвращает ресурс.

### Подробное описание

Класс сервис ресурсов Avalonia.

Реализует **IResourceService**.

См. определение в файле **AvaloniaResourceService.cs** строка **15**

### Конструктор(ы)

#### View.Implementations.AvaloniaResourceService.AvaloniaResourceService ()

Создаёт экземпляр класса **AvaloniaResourceService** по умолчанию.

См. определение в файле **AvaloniaResourceService.cs** строка **30**

### Методы

#### object? View.Implementations.AvaloniaResourceService.GetResource (object key)

Возвращает ресурс по ключу.

##### Аргументы

|  |  |
| --- | --- |
| *key* | Ключ. |

##### Возвращает

Возвращает ресурс.

Замещает **ViewModel.Interfaces.AppStates.IResourceService** (*стр.272*).

См. определение в файле **AvaloniaResourceService.cs** строка **38**

## Класс View.Implementations.AvaloniaThemeManager

Класс менеджер тем Avalonia.

Граф наследования:View.Implementations.AvaloniaThemeManager:

### Открытые члены

* **AvaloniaThemeManager** ()

*Создаёт экземпляр класса* ***AvaloniaThemeManager*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Свойства

* IEnumerable< object > **Themes** [get]

*Возвращает темы.*

* object **ActualTheme** [get, set]

*Возвращает и задаёт актуальную тему.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс менеджер тем Avalonia.

Наследует **TrackableObject**. Реализует **IThemeManager**.

См. определение в файле **AvaloniaThemeManager.cs** строка **19**

### Конструктор(ы)

#### View.Implementations.AvaloniaThemeManager.AvaloniaThemeManager ()

Создаёт экземпляр класса **AvaloniaThemeManager** по умолчанию.

См. определение в файле **AvaloniaThemeManager.cs** строка **39**

### Полный список свойств

#### object View.Implementations.AvaloniaThemeManager.ActualTheme[get], [set]

Возвращает и задаёт актуальную тему.

Замещает **ViewModel.Interfaces.AppStates.Settings.IThemeManager** (*стр.317*).

См. определение в файле **AvaloniaThemeManager.cs** строка **30**

#### IEnumerable<object> View.Implementations.AvaloniaThemeManager.Themes[get]

Возвращает темы.

Замещает **ViewModel.Interfaces.AppStates.Settings.IThemeManager** (*стр.317*).

См. определение в файле **AvaloniaThemeManager.cs** строка **27**

## Класс MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

Абстрактный класс базовой оценки модели обучения методом кросс-валидации.

Граф наследования:MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator:

### Открытые члены

* **BaseCrossValidationLearningEvaluator** ()

*Создаёт экземпляр класса* ***BaseCrossValidationLearningEvaluator*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* virtual void **OnPropertyChanged< T >** (T oldValue, T newValue)

*Вызывается при изменении свойства.*

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Свойства

* int **NumberOfFolds** [get, set]

*Возвращает и задаёт количество сегментов.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Абстрактный класс базовой оценки модели обучения методом кросс-валидации.

Наследует **TrackableObject**. Реализует **ILearningEvaluator**.

См. определение в файле **BaseCrossValidationEvaluator.cs** строка **13**

### Конструктор(ы)

#### MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator.BaseCrossValidationLearningEvaluator ()

Создаёт экземпляр класса **BaseCrossValidationLearningEvaluator** по умолчанию.

См. определение в файле **BaseCrossValidationEvaluator.cs** строка **34**

### Методы

#### virtual void MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator.OnPropertyChanged< T > (T oldValue, T newValue)[protected], [virtual]

Вызывается при изменении свойства.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### Аргументы

|  |  |
| --- | --- |
| *oldValue* | Старое значение. |
| *newValue* | Новое значение. |

Переопределяется в **MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >** (*стр.133*) и **MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >** (*стр.152*).

См. определение в файле **BaseCrossValidationEvaluator.cs** строка **45**

### Полный список свойств

#### int MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator.NumberOfFolds[get], [set]

Возвращает и задаёт количество сегментов.

См. определение в файле **BaseCrossValidationEvaluator.cs** строка **24**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext

Базовый класс контекста базы данных.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext:

### Открытые члены

* **BaseDbContext** (string connectionString)

*Создаёт экземпляр класса* ***BaseDbContext****.*

### Защищенные члены

* override void **OnModelCreating** (ModelBuilder modelBuilder)
* override void **OnConfiguring** (DbContextOptionsBuilder optionsBuilder)

### Защищенные данные

* string **\_connectionString**

*Строка подключения.*

### Свойства

* DbSet< **TaskEntity** > **Tasks** [get, set]

*Возвращает и задаёт сущности задачи.*

* DbSet< **MetadataEntity** > **Metadata** [get, set]

*Возвращает и задаёт сущности метаданных.*

* DbSet< **TagEntity** > **Tags** [get, set]

*Возвращает и задаёт сущности тега.*

* DbSet< **TaskCompositeEntity** > **TaskComposites** [get, set]

*Возвращает и задаёт сущности составной задачи.*

* DbSet< **TaskElementEntity** > **TaskElements** [get, set]

*Возвращает и задаёт сущности элементарной задачи.*

* DbSet< **TimeIntervalEntity** > **TimeIntervals** [get, set]

*Возвращает и задаёт сущности временного интерала.*

### Подробное описание

Базовый класс контекста базы данных.

Наследует DbContext.

См. определение в файле **BaseDbContext.cs** строка **14**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext.BaseDbContext (string connectionString)

Создаёт экземпляр класса **BaseDbContext**.

##### Аргументы

|  |  |
| --- | --- |
| *connectionString* | Строка подключения. |

См. определение в файле **BaseDbContext.cs** строка **55**

### Методы

#### override void ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext.OnConfiguring (DbContextOptionsBuilder optionsBuilder)[protected]

См. определение в файле **BaseDbContext.cs** строка **69**

#### override void ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext.OnModelCreating (ModelBuilder modelBuilder)[protected]

См. определение в файле **BaseDbContext.cs** строка **61**

### Данные класса

#### string ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext.\_connectionString[protected]

Строка подключения.

См. определение в файле **BaseDbContext.cs** строка **19**

### Полный список свойств

#### DbSet<MetadataEntity> ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext.Metadata[get], [set]

Возвращает и задаёт сущности метаданных.

См. определение в файле **BaseDbContext.cs** строка **29**

#### DbSet<TagEntity> ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext.Tags[get], [set]

Возвращает и задаёт сущности тега.

См. определение в файле **BaseDbContext.cs** строка **34**

#### DbSet<TaskCompositeEntity> ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext.TaskComposites[get], [set]

Возвращает и задаёт сущности составной задачи.

См. определение в файле **BaseDbContext.cs** строка **39**

#### DbSet<TaskElementEntity> ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext.TaskElements[get], [set]

Возвращает и задаёт сущности элементарной задачи.

См. определение в файле **BaseDbContext.cs** строка **44**

#### DbSet<TaskEntity> ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext.Tasks[get], [set]

Возвращает и задаёт сущности задачи.

См. определение в файле **BaseDbContext.cs** строка **24**

#### DbSet<TimeIntervalEntity> ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext.TimeIntervals[get], [set]

Возвращает и задаёт сущности временного интерала.

См. определение в файле **BaseDbContext.cs** строка **49**

## Шаблон класса ViewModel.ViewModels.BaseDialogViewModel< A, R >

Базовый абстрактный класс контроллера диалога.

Граф наследования:ViewModel.ViewModels.BaseDialogViewModel< A, R >:

### Открытые члены

* async Task< R > **Invoke** (**BaseViewModel** parent, A args)

*Вызывает диалог.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Защищенные члены

* void **GetArgs** (A args)

*Получает аргументы, переданные в диалог.*

### Защищенные данные

* TaskCompletionSource< R >? **\_taskSource**

*Источник для завершения задачи, который используется для возврата результата диалога.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

### Свойства

* **BaseViewModel**? **Parent** [get]

*Возвращает родителя.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Базовый абстрактный класс контроллера диалога.

Наследует **BaseViewModel**.

См. определение в файле **BaseDialogViewModel.cs** строка **9**

### Методы

#### void ViewModel.ViewModels.BaseDialogViewModel< A, R >.GetArgs (A args)[abstract], [protected]

Получает аргументы, переданные в диалог.

##### Аргументы

|  |  |
| --- | --- |
| *args* | Аргументы. |

#### async Task< R > ViewModel.ViewModels.BaseDialogViewModel< A, R >.Invoke (BaseViewModel parent, A args)

Вызывает диалог.

##### Аргументы

|  |  |
| --- | --- |
| *parent* | Родитель. |
| *args* | Аргументы. |

##### Возвращает

Возвращает задачу процесса диалога с результатом.

##### Исключения

|  |  |
| --- | --- |
| *InvalidOperationException* | Выбрасывает, если родитель уже есть. |

См. определение в файле **BaseDialogViewModel.cs** строка **41**

### Данные класса

#### TaskCompletionSource<R>? ViewModel.ViewModels.BaseDialogViewModel< A, R >.\_taskSource[protected]

Источник для завершения задачи, который используется для возврата результата диалога.

См. определение в файле **BaseDialogViewModel.cs** строка **14**

### Полный список свойств

#### BaseViewModel? ViewModel.ViewModels.BaseDialogViewModel< A, R >.Parent[get]

Возвращает родителя.

См. определение в файле **BaseDialogViewModel.cs** строка **24**

## Шаблон класса MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

Абстрактный класс базового конвертора данных в данные для предсказания и наоборот.

Граф наследования:MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >:

### Открытые члены

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

##### *Аргументы*

|  |  |
| --- | --- |
| *predicted* | Предсказанное значение. |

##### *Возвращает*

Возвращает конвертированное выходное значение.

* IEnumerable< double > **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные входные данные для предсказания.

### Защищенные члены

* **BaseLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor)

*Создаёт экземпляр класса* ***BaseLearningConverter<R, D, DT, DR>****.*

* IEnumerable< double?> **ExtractFeatures** (DT dataItem)

*Извлекает признаки у элемента данных.*

### Защищенные данные

* readonly **IPrimaryPointDataProcessor** **\_primaryPointDataProcessor**

*Первичный обработчик точечных данных.*

* IEnumerable< int >? **\_removedColumnsIndices**

*Индексы удалённых столбцов.*

* IEnumerable< **IScaler** >? **\_scalers**

*Коллекция масштабирования данных.*

### Подробное описание

Абстрактный класс базового конвертора данных в данные для предсказания и наоборот.

Реализует **ILearningConverter<IEnumerable<double>, R, DT, DR>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *R* | Тип выходных данных для предсказания. |
| *D* | Тип данных. |
| *DT* | Тип входных данных. |
| *DR* | Тип выходных данных. |

См. определение в файле **BaseLearningConverter.cs** строка **16**

### Методы

#### MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >.BaseLearningConverter (IPrimaryPointDataProcessor primaryPointDataProcessor)[protected]

Создаёт экземпляр класса **BaseLearningConverter<R, D, DT, DR>**.

##### Аргументы

|  |  |
| --- | --- |
| *primaryPointDataProcessor* | Первичный обработчик точечных данных. |

См. определение в файле **BaseLearningConverter.cs** строка **38**

#### IEnumerable< double > MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >.ConvertData (DT data)

Конвертирует входные данные под входные данные для предсказания.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает конвертированные входные данные для предсказания.

Замещает **MachineLearning.Interfaces.Generals.ILearningConverter< T, R, DT, DR >** (*стр.253*).

См. определение в файле **BaseLearningConverter.cs** строка **47**

#### DR MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >.ConvertPredicted (R predicted)[abstract]

Конвертирует предсказанное значение в выходное значение.

##### Аргументы

|  |  |
| --- | --- |
| *predicted* | Предсказанное значение. |

##### Возвращает

Возвращает конвертированное выходное значение.

Замещает **MachineLearning.Interfaces.Generals.ILearningConverter< T, R, DT, DR >** (*стр.253*).

#### IEnumerable< double?> MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >.ExtractFeatures (DT dataItem)[abstract], [protected]

Извлекает признаки у элемента данных.

##### Аргументы

|  |  |
| --- | --- |
| *dataItem* | Элемент данных. |

##### Возвращает

Возвращет, извлеченное у элемента данных, целевое значение.

### Данные класса

#### readonly IPrimaryPointDataProcessor MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >.\_primaryPointDataProcessor[protected]

Первичный обработчик точечных данных.

См. определение в файле **BaseLearningConverter.cs** строка **22**

#### IEnumerable<int>? MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >.\_removedColumnsIndices[protected]

Индексы удалённых столбцов.

См. определение в файле **BaseLearningConverter.cs** строка **27**

#### IEnumerable<IScaler>? MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >.\_scalers[protected]

Коллекция масштабирования данных.

См. определение в файле **BaseLearningConverter.cs** строка **32**

## Класс ViewModel.ViewModels.BasePageViewModel

Базовый класс контроллера страницы.

Граф наследования:ViewModel.ViewModels.BasePageViewModel:

### Свойства

* object **Metadata** [get, protected set]

*Возращает метаданные.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

### Подробное описание

Базовый класс контроллера страницы.

Наследует **BaseViewModel**.

См. определение в файле **BasePageViewModel.cs** строка **9**

### Полный список свойств

#### object ViewModel.ViewModels.BasePageViewModel.Metadata[get], [protected set]

Возращает метаданные.

См. определение в файле **BasePageViewModel.cs** строка **14**

## Шаблон класса MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >

Абстрактный класс базовой оценки модели обучения с учителем методом кросс-валидации.

Граф наследования:MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >:

### Открытые члены

* async Task< **ScoreMetricCategory** > **Evaluate** (IEnumerable< T > data, IEnumerable< R > targets)

*Оценивает модель обучения.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |
| *targets* | Целевые значения. |

##### *Возвращает*

Возвращает категорию модели обучения.

#### Открытые члены унаследованные от MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

* **BaseCrossValidationLearningEvaluator** ()

*Создаёт экземпляр класса* ***BaseCrossValidationLearningEvaluator*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* IEnumerable< **ValidationFold** > **GetValidationFolds** (IEnumerable< T > data, IEnumerable< R > targets)

*Возвращает сегменты валидации.*

* virtual **ScoreMetricCategory** **GetScoresCategory** (IEnumerable< double > scores)

*Определяет категорию оценок.*

* override void **OnPropertyChanged< T >** (T oldValue, T newValue)

*Вызывается при изменении свойства.*

##### *Параметры шаблона*

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### *Аргументы*

|  |  |
| --- | --- |
| *oldValue* | Старое значение. |
| *newValue* | Новое значение. |

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Свойства

* ISupervisedLearningModel< T, R > **Model** [get, set]

*Возвращает и задаёт модель обучения с учителем.*

* ISupervisedScoreMetric< R > **ScoreMetric** [get, set]

*Возвращает и задаёт модель обучения с учителем.*

#### Свойства унаследованные от MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

* int **NumberOfFolds** [get, set]

*Возвращает и задаёт количество сегментов.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Абстрактный класс базовой оценки модели обучения с учителем методом кросс-валидации.

Наследует **BaseCrossValidationLearningEvaluator**. Реализует **ISupervisedLearningEvaluator<T, R>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |

См. определение в файле **BaseSupervisedCrossValidationEvaluator.cs** строка **14**

### Методы

#### async Task< ScoreMetricCategory > MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >.Evaluate (IEnumerable< T > data, IEnumerable< R > targets)

Оценивает модель обучения.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |
| *targets* | Целевые значения. |

##### Возвращает

Возвращает категорию модели обучения.

Замещает **MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >** (*стр.290*).

См. определение в файле **BaseSupervisedCrossValidationEvaluator.cs** строка **46**

#### virtual ScoreMetricCategory MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >.GetScoresCategory (IEnumerable< double > scores)[protected], [virtual]

Определяет категорию оценок.

##### Аргументы

|  |  |
| --- | --- |
| *scores* | Оценки. |

##### Возвращает

Возвращает категорию оценок.

#### IEnumerable< ValidationFold > MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >.GetValidationFolds (IEnumerable< T > data, IEnumerable< R > targets)[abstract], [protected]

Возвращает сегменты валидации.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |
| *targets* | Целевые значения. |

##### Возвращает

Возвращает сегменты валидации.

#### override void MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >.OnPropertyChanged< T > (T oldValue, T newValue)[protected], [virtual]

Вызывается при изменении свойства.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### Аргументы

|  |  |
| --- | --- |
| *oldValue* | Старое значение. |
| *newValue* | Новое значение. |

Переопределяет метод предка **MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator** (*стр.117*).

См. определение в файле **BaseSupervisedCrossValidationEvaluator.cs** строка **90**

### Полный список свойств

#### ISupervisedLearningModel<T, R> MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >.Model[get], [set]

Возвращает и задаёт модель обучения с учителем.

Замещает **MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >** (*стр.290*).

См. определение в файле **BaseSupervisedCrossValidationEvaluator.cs** строка **30**

#### ISupervisedScoreMetric<R> MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >.ScoreMetric[get], [set]

Возвращает и задаёт модель обучения с учителем.

Замещает **MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >** (*стр.290*).

См. определение в файле **BaseSupervisedCrossValidationEvaluator.cs** строка **39**

## Шаблон класса ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >

Абстрактный класс базового контроллера обучения модели обучения с учителем.

Граф наследования:ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >:

### Открытые члены

* **BaseSupervisedEvaluatorLearningController** (ISupervisedLearningConverter< T, R, D, DT, DR > converter, ISupervisedLearningModel< T, R > model, ISupervisedLearningEvaluator< T, R > evaluator)

*Создаёт экземпляр классса* ***BaseSupervisedEvaluatorLearningController<T, R, D, DT, DR>****.*

* async Task< bool > **Train** (IEnumerable< D > data)

*Выполняет обучение модели на основе данных.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает задачу процесса обучения с результатом true , если обучение прошло успешно, иначе false .

* DR **Predict** (DT data)

*Выполняет предсказание на основе данных.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает предсказанное значение.

### Свойства

* **ScoreMetricCategory** **MinScoreCategory** [get, set]

*Возвращает и задаёт минимальную категорию метрики оценки.*

* bool **IsValidModel** [get, protected set]

*Возвращает логическое значение, указывающее на достоверность модели обучения.*

### Подробное описание

Абстрактный класс базового контроллера обучения модели обучения с учителем.

Реализует **ILearningController<D, DT, DR>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |
| *D* | Тип данных. |
| *DT* | Тип входных данных. |
| *DR* | Тип выходных данных. |

См. определение в файле **BaseSupervisedEvaluatorLearningController.cs** строка **19**

### Методы

#### ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >.BaseSupervisedEvaluatorLearningController (ISupervisedLearningConverter< T, R, D, DT, DR > converter, ISupervisedLearningModel< T, R > model, ISupervisedLearningEvaluator< T, R > evaluator)

Создаёт экземпляр классса **BaseSupervisedEvaluatorLearningController<T, R, D, DT, DR>**.

##### Аргументы

|  |  |
| --- | --- |
| *converter* | Конвертор данных в данные для предсказания с учителем и наоборот. |
| *model* | Модель обучения c учителем. |
| *evaluator* | Оценка модели обучения с учителем. |

См. определение в файле **BaseSupervisedEvaluatorLearningController.cs** строка **54**

#### DR ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >.Predict (DT data)

Выполняет предсказание на основе данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает предсказанное значение.

Замещает **ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR >** (*стр.251*).

См. определение в файле **BaseSupervisedEvaluatorLearningController.cs** строка **88**

#### async Task< bool > ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >.Train (IEnumerable< D > data)

Выполняет обучение модели на основе данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает задачу процесса обучения с результатом true , если обучение прошло успешно, иначе false .

Замещает **ViewModel.Interfaces.ModelLearning.IModelTeacher< D >** (*стр.260*).

См. определение в файле **BaseSupervisedEvaluatorLearningController.cs** строка **65**

### Полный список свойств

#### bool ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >.IsValidModel[get], [protected set]

Возвращает логическое значение, указывающее на достоверность модели обучения.

Замещает **ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR >** (*стр.251*).

См. определение в файле **BaseSupervisedEvaluatorLearningController.cs** строка **43**

#### ScoreMetricCategory ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >.MinScoreCategory[get], [set]

Возвращает и задаёт минимальную категорию метрики оценки.

См. определение в файле **BaseSupervisedEvaluatorLearningController.cs** строка **40**

## Шаблон класса MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

Абстрактный класс базового конвертора данных в данные для предсказания с учителем и наоборот.

Граф наследования:MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >:

### Открытые члены

* LearningModelData< IEnumerable< double >, R > **FitConvertData** (IEnumerable< D > data)

*Адаптирует конвертацию данных под данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные данные для предсказания.

#### Открытые члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

##### *Аргументы*

|  |  |
| --- | --- |
| *predicted* | Предсказанное значение. |

##### *Возвращает*

Возвращает конвертированное выходное значение.

* IEnumerable< double > **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные входные данные для предсказания.

### Защищенные члены

* **BaseSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors)

*Создаёт экземпляр класса* ***BaseSupervisedLearningConverter<R, D, DT, DR>****.*

* R **ProcessTarget** (D item)

*Обрабатывает целевые значения элемента.*

* IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)

*Создаёт масштабирование данных.*

* DataProcessorResult< IEnumerable< double > > **ProcessFeaturesData** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* void **CreateScalers** (double[][] featuresArray)

*Создаёт коллекцию масштабирования данных.*

* IEnumerable< int > **NormalizeRemovedIndices** (IEnumerable< IEnumerable< int > > removedIndicesGroups)

*Нормализует индексы удалённых индексов.*

* IEnumerable< R > **ProcessTargets** (IEnumerable< D > data, IEnumerable< int > removedRowsIndices)

*Обрабатывает целевые значения данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* **BaseLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor)

*Создаёт экземпляр класса* ***BaseLearningConverter<R, D, DT, DR>****.*

* IEnumerable< double?> **ExtractFeatures** (DT dataItem)

*Извлекает признаки у элемента данных.*

### Защищенные данные

* readonly IEnumerable< **IPointDataProcessor** > **\_pointDataProcessors**

*Обработчики точечных данных.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* readonly **IPrimaryPointDataProcessor** **\_primaryPointDataProcessor**

*Первичный обработчик точечных данных.*

* IEnumerable< int >? **\_removedColumnsIndices**

*Индексы удалённых столбцов.*

* IEnumerable< **IScaler** >? **\_scalers**

*Коллекция масштабирования данных.*

### Подробное описание

Абстрактный класс базового конвертора данных в данные для предсказания с учителем и наоборот.

Наследует **BaseLearningConverter<R, D, DT, DR>**. Реализует **ISupervisedLearningConverter<IEnumerable<double>, R, D, DT, DR>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *R* | Тип выходных данных для предсказания. |
| *D* | Тип данных. |
| *DT* | Тип входных данных. |
| *DR* | Тип выходных данных. |

См. определение в файле **BaseSupervisedLearningConverter.cs** строка **20**

### Методы

#### MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >.BaseSupervisedLearningConverter (IPrimaryPointDataProcessor primaryPointDataProcessor, IEnumerable< IPointDataProcessor > pointDataProcessors)[protected]

Создаёт экземпляр класса **BaseSupervisedLearningConverter<R, D, DT, DR>**.

##### Аргументы

|  |  |
| --- | --- |
| *primaryPointDataProcessor* | Первичный обработчик точечных данных. |
| *pointDataProcessors* | Обработчики точечных данных. |

См. определение в файле **BaseSupervisedLearningConverter.cs** строка **34**

#### IScaler MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >.CreateScaler (int index, IEnumerable< int > removedColumnsIndices)[abstract], [protected]

Создаёт масштабирование данных.

##### Аргументы

|  |  |
| --- | --- |
| *index* | Индекс. |
| *removedColumnsIndices* | Индексы удалённых столбцов. |

##### Возвращает

Возвращает масштабирование данных.

#### void MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >.CreateScalers (double featuresArray[][])[protected]

Создаёт коллекцию масштабирования данных.

##### Аргументы

|  |  |
| --- | --- |
| *featuresArray* | Массив признаков. |

См. определение в файле **BaseSupervisedLearningConverter.cs** строка **114**

#### LearningModelData< IEnumerable< double >, R > MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >.FitConvertData (IEnumerable< D > data)

Адаптирует конвертацию данных под данные для предсказания.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает конвертированные данные для предсказания.

Замещает **MachineLearning.Interfaces.Generals.ISupervisedLearningConverter< T, R, D, DT, DR >** (*стр.288*).

См. определение в файле **BaseSupervisedLearningConverter.cs** строка **42**

#### IEnumerable< int > MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >.NormalizeRemovedIndices (IEnumerable< IEnumerable< int > > removedIndicesGroups)[protected]

Нормализует индексы удалённых индексов.

##### Аргументы

|  |  |
| --- | --- |
| *removedIndicesGroups* | Группы индексов удалённых индексов. |

##### Возвращает

Возвращает нормализованные удалённые индексы.

См. определение в файле **BaseSupervisedLearningConverter.cs** строка **134**

#### IEnumerable< IEnumerable< double?> > MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >.ProcessFeatures (IEnumerable< D > data)[abstract], [protected]

Обрабатывает признаков данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает обработанные признаков данных.

#### DataProcessorResult< IEnumerable< double > > MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >.ProcessFeaturesData (IEnumerable< D > data)[protected]

Обрабатывает признаков данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает данные о преобразовании признаков данных.

См. определение в файле **BaseSupervisedLearningConverter.cs** строка **84**

#### R MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >.ProcessTarget (D item)[abstract], [protected]

Обрабатывает целевые значения элемента.

##### Аргументы

|  |  |
| --- | --- |
| *item* | Элемента. |

##### Возвращает

Возвращет обработанное целевое значение элемента.

#### IEnumerable< R > MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >.ProcessTargets (IEnumerable< D > data, IEnumerable< int > removedRowsIndices)[protected]

Обрабатывает целевые значения данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |
| *removedRowsIndices* | Индексы удалённых строк. |

##### Возвращает

Возвращает обработанные целевые значения данных.

См. определение в файле **BaseSupervisedLearningConverter.cs** строка **166**

### Данные класса

#### readonly IEnumerable<IPointDataProcessor> MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >.\_pointDataProcessors[protected]

Обработчики точечных данных.

См. определение в файле **BaseSupervisedLearningConverter.cs** строка **27**

## Шаблон класса ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >

Абстрактный класс базового конвертора элементраных задач в данные для предсказания с учителем и наоборот.

Граф наследования:ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >:

### Защищенные члены

* **BaseTaskElementSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors, IFactory< **IScaler** > scalerFactory, IDataTransformer< **TaskMetadata**, int?> metadataCategoriesTransformer, IDataTransformer< **TaskMetadata**, IEnumerable< int > > metadataTagsITransformer)

*Создаёт экземпляр класса* ***BaseTaskElementSupervisedLearningConverter<R, DR>****.*

* override IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< **ITaskElement** > data)
* override IEnumerable< double?> **ExtractFeatures** (**ITaskElement** dataItem)
* override **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)
* List< double?> **ExtractPrimaryFeatures** (**ITaskElement** dataItem)

*Извлекает первичные признаки у элемента данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* **BaseSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors)

*Создаёт экземпляр класса* ***BaseSupervisedLearningConverter<R, D, DT, DR>****.*

* R **ProcessTarget** (D item)

*Обрабатывает целевые значения элемента.*

* IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)

*Создаёт масштабирование данных.*

* DataProcessorResult< IEnumerable< double > > **ProcessFeaturesData** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* void **CreateScalers** (double[][] featuresArray)

*Создаёт коллекцию масштабирования данных.*

* IEnumerable< int > **NormalizeRemovedIndices** (IEnumerable< IEnumerable< int > > removedIndicesGroups)

*Нормализует индексы удалённых индексов.*

* IEnumerable< R > **ProcessTargets** (IEnumerable< D > data, IEnumerable< int > removedRowsIndices)

*Обрабатывает целевые значения данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* **BaseLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor)

*Создаёт экземпляр класса* ***BaseLearningConverter<R, D, DT, DR>****.*

* IEnumerable< double?> **ExtractFeatures** (DT dataItem)

*Извлекает признаки у элемента данных.*

### Защищенные данные

* readonly IFactory< **IScaler** > **\_scalerFactory**

*Фабрика, создающая масштабирования данных.*

* readonly IDataTransformer< **TaskMetadata**, int?> **\_metadataCategoriesTransformer**

*Преобразование категории метаданных в данные для предсказания.*

* readonly IDataTransformer< **TaskMetadata**, IEnumerable< int > > **\_metadataTagsTransformer**

*Преобразование теги метаданных в данные для предсказания.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* readonly IEnumerable< **IPointDataProcessor** > **\_pointDataProcessors**

*Обработчики точечных данных.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* readonly **IPrimaryPointDataProcessor** **\_primaryPointDataProcessor**

*Первичный обработчик точечных данных.*

* IEnumerable< int >? **\_removedColumnsIndices**

*Индексы удалённых столбцов.*

* IEnumerable< **IScaler** >? **\_scalers**

*Коллекция масштабирования данных.*

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* LearningModelData< IEnumerable< double >, R > **FitConvertData** (IEnumerable< D > data)

*Адаптирует конвертацию данных под данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные данные для предсказания.

#### Открытые члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

##### *Аргументы*

|  |  |
| --- | --- |
| *predicted* | Предсказанное значение. |

##### *Возвращает*

Возвращает конвертированное выходное значение.

* IEnumerable< double > **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные входные данные для предсказания.

### Подробное описание

Абстрактный класс базового конвертора элементраных задач в данные для предсказания с учителем и наоборот.

Наследует **BaseSupervisedLearningConverter<R, ITaskElement, ITaskElement, DR>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *R* | Тип выходных данных для предсказания. |
| *DR* | Тип выходных данных. |

См. определение в файле **BaseTaskElementSupervisedLearningConverter.cs** строка **22**

### Методы

#### ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >.BaseTaskElementSupervisedLearningConverter (IPrimaryPointDataProcessor primaryPointDataProcessor, IEnumerable< IPointDataProcessor > pointDataProcessors, IFactory< IScaler > scalerFactory, IDataTransformer< TaskMetadata, int?> metadataCategoriesTransformer, IDataTransformer< TaskMetadata, IEnumerable< int > > metadataTagsITransformer)[protected]

Создаёт экземпляр класса **BaseTaskElementSupervisedLearningConverter<R, DR>**.

##### Аргументы

|  |  |
| --- | --- |
| *primaryPointDataProcessor* | Первичный обработчик точечных данных. |
| *pointDataProcessors* | Обработчики точечных данных. |
| *scalerFactory* | Фабрика, создающая масштабирования данных. |
| *metadataCategoriesTransformer* | Преобразование категории метаданных в данные для предсказания. |
| *metadataTagsITransformer* | Преобразование теги метаданных в данные для предсказания. |

См. определение в файле **BaseTaskElementSupervisedLearningConverter.cs** строка **52**

#### override IEnumerable< double?> ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >.ExtractFeatures (ITaskElement dataItem)[protected]

См. определение в файле **BaseTaskElementSupervisedLearningConverter.cs** строка **84**

#### List< double?> ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >.ExtractPrimaryFeatures (ITaskElement dataItem)[abstract], [protected]

Извлекает первичные признаки у элемента данных.

##### Аргументы

|  |  |
| --- | --- |
| *dataItem* | Элемент данных. |

##### Возвращает

Возвращает список значений.

#### override IEnumerable< IEnumerable< double?> > ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >.ProcessFeatures (IEnumerable< ITaskElement > data)[protected]

См. определение в файле **BaseTaskElementSupervisedLearningConverter.cs** строка **66**

### Данные класса

#### readonly IDataTransformer<TaskMetadata, int?> ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >.\_metadataCategoriesTransformer[protected]

Преобразование категории метаданных в данные для предсказания.

См. определение в файле **BaseTaskElementSupervisedLearningConverter.cs** строка **33**

#### readonly IDataTransformer<TaskMetadata, IEnumerable<int> > ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >.\_metadataTagsTransformer[protected]

Преобразование теги метаданных в данные для предсказания.

См. определение в файле **BaseTaskElementSupervisedLearningConverter.cs** строка **38**

#### readonly IFactory<IScaler> ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >.\_scalerFactory[protected]

Фабрика, создающая масштабирования данных.

См. определение в файле **BaseTaskElementSupervisedLearningConverter.cs** строка **28**

## Шаблон класса MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >

Абстрактный класс базовой оценки модели обучения без учителя методом кросс-валидации.

Граф наследования:MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >:

### Открытые члены

* async Task< **ScoreMetricCategory** > **Evaluate** (IEnumerable< T > data)

*Оценивает модель обучения.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает категорию модели обучения.

#### Открытые члены унаследованные от MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

* **BaseCrossValidationLearningEvaluator** ()

*Создаёт экземпляр класса* ***BaseCrossValidationLearningEvaluator*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* IEnumerable< **ValidationFold** > **GetValidationFolds** (IEnumerable< T > data)

*Возвращает сегменты валидации.*

* virtual **ScoreMetricCategory** **GetScoresCategory** (IEnumerable< double > scores)

*Определяет категорию оценок.*

* override void **OnPropertyChanged< T >** (T oldValue, T newValue)

*Вызывается при изменении свойства.*

##### *Параметры шаблона*

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### *Аргументы*

|  |  |
| --- | --- |
| *oldValue* | Старое значение. |
| *newValue* | Новое значение. |

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Свойства

* IUnsupervisedLearningModel< T, R > **Model** [get, set]

*Возвращает и задаёт модель обучения без учителя.*

* IUnsupervisedScoreMetric< R, T > **ScoreMetric** [get, set]

*Возвращает и задаёт модель обучения без учителя.*

#### Свойства унаследованные от MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

* int **NumberOfFolds** [get, set]

*Возвращает и задаёт количество сегментов.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Абстрактный класс базовой оценки модели обучения без учителя методом кросс-валидации.

Наследует **BaseCrossValidationLearningEvaluator**. Реализует **IUnsupervisedLearningEvaluator<T, R>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |

См. определение в файле **BaseUnsupervisedCrossValidationEvaluator.cs** строка **14**

### Методы

#### async Task< ScoreMetricCategory > MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >.Evaluate (IEnumerable< T > data)

Оценивает модель обучения.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает категорию модели обучения.

Замещает **MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R >** (*стр.330*).

См. определение в файле **BaseUnsupervisedCrossValidationEvaluator.cs** строка **46**

#### virtual ScoreMetricCategory MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >.GetScoresCategory (IEnumerable< double > scores)[protected], [virtual]

Определяет категорию оценок.

##### Аргументы

|  |  |
| --- | --- |
| *scores* | Оценки. |

##### Возвращает

Возвращает категорию оценок.

#### IEnumerable< ValidationFold > MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >.GetValidationFolds (IEnumerable< T > data)[abstract], [protected]

Возвращает сегменты валидации.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает сегменты валидации.

#### override void MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >.OnPropertyChanged< T > (T oldValue, T newValue)[protected], [virtual]

Вызывается при изменении свойства.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### Аргументы

|  |  |
| --- | --- |
| *oldValue* | Старое значение. |
| *newValue* | Новое значение. |

Переопределяет метод предка **MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator** (*стр.117*).

См. определение в файле **BaseUnsupervisedCrossValidationEvaluator.cs** строка **80**

### Полный список свойств

#### IUnsupervisedLearningModel<T, R> MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >.Model[get], [set]

Возвращает и задаёт модель обучения без учителя.

Замещает **MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R >** (*стр.330*).

См. определение в файле **BaseUnsupervisedCrossValidationEvaluator.cs** строка **30**

#### IUnsupervisedScoreMetric<R, T> MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >.ScoreMetric[get], [set]

Возвращает и задаёт модель обучения без учителя.

Замещает **MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R >** (*стр.330*).

См. определение в файле **BaseUnsupervisedCrossValidationEvaluator.cs** строка **39**

## Шаблон класса MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >

Абстрактный класс базового конвертора данных в данные для предсказания без учителя и наоборот.

Граф наследования:MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >:

### Открытые члены

* IEnumerable< IEnumerable< double > > **FitConvertData** (IEnumerable< D > data)

*Адаптирует конвертацию данных под данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные входных данных для предсказания.

#### Открытые члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

##### *Аргументы*

|  |  |
| --- | --- |
| *predicted* | Предсказанное значение. |

##### *Возвращает*

Возвращает конвертированное выходное значение.

* IEnumerable< double > **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные входные данные для предсказания.

### Защищенные члены

* **BaseUnsupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors)

*Создаёт экземпляр класса* ***BaseUnsupervisedLearningConverter<R, D, DT, DR>****.*

* IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)

*Создаёт масштабирование данных.*

* DataProcessorResult< IEnumerable< double > > **ProcessFeaturesData** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* void **CreateScalers** (double[][] featuresArray)

*Создаёт коллекцию масштабирования данных.*

* IEnumerable< int > **NormalizeRemovedIndices** (IEnumerable< IEnumerable< int > > removedIndicesGroups)

*Нормализует индексы удалённых индексов.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* **BaseLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor)

*Создаёт экземпляр класса* ***BaseLearningConverter<R, D, DT, DR>****.*

* IEnumerable< double?> **ExtractFeatures** (DT dataItem)

*Извлекает признаки у элемента данных.*

### Защищенные данные

* readonly IEnumerable< **IPointDataProcessor** > **\_pointDataProcessors**

*Обработчики точечных данных.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* readonly **IPrimaryPointDataProcessor** **\_primaryPointDataProcessor**

*Первичный обработчик точечных данных.*

* IEnumerable< int >? **\_removedColumnsIndices**

*Индексы удалённых столбцов.*

* IEnumerable< **IScaler** >? **\_scalers**

*Коллекция масштабирования данных.*

### Подробное описание

Абстрактный класс базового конвертора данных в данные для предсказания без учителя и наоборот.

Наследует **BaseLearningConverter<R, D, DT, DR>**. Реализует **IUnsupervisedLearningConverter<IEnumerable<double>, R, D, DT, DR>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *R* | Тип выходных данных для предсказания. |
| *D* | Тип данных. |
| *DT* | Тип входных данных. |
| *DR* | Тип выходных данных. |

См. определение в файле **BaseUnsupervisedLearningConverter.cs** строка **19**

### Методы

#### MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >.BaseUnsupervisedLearningConverter (IPrimaryPointDataProcessor primaryPointDataProcessor, IEnumerable< IPointDataProcessor > pointDataProcessors)[protected]

Создаёт экземпляр класса **BaseUnsupervisedLearningConverter<R, D, DT, DR>**.

##### Аргументы

|  |  |
| --- | --- |
| *primaryPointDataProcessor* | Первичный обработчик точечных данных. |
| *pointDataProcessors* | Обработчики точечных данных. |

См. определение в файле **BaseUnsupervisedLearningConverter.cs** строка **33**

#### IScaler MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >.CreateScaler (int index, IEnumerable< int > removedColumnsIndices)[abstract], [protected]

Создаёт масштабирование данных.

##### Аргументы

|  |  |
| --- | --- |
| *index* | Индекс. |
| *removedColumnsIndices* | Индексы удалённых столбцов. |

##### Возвращает

Возвращает масштабирование данных.

#### void MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >.CreateScalers (double featuresArray[][])[protected]

Создаёт коллекцию масштабирования данных.

##### Аргументы

|  |  |
| --- | --- |
| *featuresArray* | Массив признаков. |

См. определение в файле **BaseUnsupervisedLearningConverter.cs** строка **103**

#### IEnumerable< IEnumerable< double > > MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >.FitConvertData (IEnumerable< D > data)

Адаптирует конвертацию данных под данные для предсказания.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает конвертированные входных данных для предсказания.

Замещает **MachineLearning.Interfaces.Generals.IUnsupervisedLearningConverter< T, R, D, DT, DR >** (*стр.328*).

См. определение в файле **BaseUnsupervisedLearningConverter.cs** строка **41**

#### IEnumerable< int > MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >.NormalizeRemovedIndices (IEnumerable< IEnumerable< int > > removedIndicesGroups)[protected]

Нормализует индексы удалённых индексов.

##### Аргументы

|  |  |
| --- | --- |
| *removedIndicesGroups* | Группы индексов удалённых индексов. |

##### Возвращает

Возвращает нормализованные удалённые индексы.

См. определение в файле **BaseUnsupervisedLearningConverter.cs** строка **123**

#### IEnumerable< IEnumerable< double?> > MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >.ProcessFeatures (IEnumerable< D > data)[abstract], [protected]

Обрабатывает признаков данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает обработанные признаков данных.

#### DataProcessorResult< IEnumerable< double > > MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >.ProcessFeaturesData (IEnumerable< D > data)[protected]

Обрабатывает признаков данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает данные о преобразовании признаков данных.

См. определение в файле **BaseUnsupervisedLearningConverter.cs** строка **73**

### Данные класса

#### readonly IEnumerable<IPointDataProcessor> MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >.\_pointDataProcessors[protected]

Обработчики точечных данных.

См. определение в файле **BaseUnsupervisedLearningConverter.cs** строка **26**

## Класс ViewModel.ViewModels.BaseViewModel

Базовый класс для контроллеров.

Граф наследования:ViewModel.ViewModels.BaseViewModel:

### Открытые члены

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Защищенные данные

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

### Свойства

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Базовый класс для контроллеров.

Наследует ReactiveObject. Реализует IActivatableViewModel.

См. определение в файле **BaseViewModel.cs** строка **14**

### Конструктор(ы)

#### ViewModel.ViewModels.BaseViewModel.BaseViewModel ()

Создаёт экемпляр класса **BaseViewModel** по умолчанию.

См. определение в файле **BaseViewModel.cs** строка **52**

### Методы

#### async Task< R > ViewModel.ViewModels.BaseViewModel.AddDialog< A, R > (BaseDialogViewModel< A, R > dialog, A args)

Добавляет и вызывает диалог.

##### Параметры шаблона

|  |  |
| --- | --- |
| *A* | >Тип данных аргументов. |
| *R* | Тип данных результата. |

##### Аргументы

|  |  |
| --- | --- |
| *dialog* | Диалог. |
| *args* | Аргументы. |

##### Возвращает

Возвращает задачу процесса диалога с результатом.

См. определение в файле **BaseViewModel.cs** строка **65**

#### async Task< R > ViewModel.ViewModels.BaseViewModel.AddModal< A, R > (BaseDialogViewModel< A, R > modal, A args)

Добавляет и вызывает модальный диалог.

##### Параметры шаблона

|  |  |
| --- | --- |
| *A* | Тип данных аргументов. |
| *R* | Тип данных результата. |

##### Аргументы

|  |  |
| --- | --- |
| *modal* | Модальное окно. |
| *args* | Аргументы. |

##### Возвращает

Возвращает задачу процесса модального диалога с результатом.

См. определение в файле **BaseViewModel.cs** строка **81**

### Данные класса

#### readonly IObservable<bool> ViewModel.ViewModels.BaseViewModel.\_modalsObservable[protected]

Наблюдатель, который отслеживает модальные диалоги.

Возвращает true , если нет модальных диалогов, и false , если есть.

См. определение в файле **BaseViewModel.cs** строка **22**

### Полный список свойств

#### ViewModelActivator ViewModel.ViewModels.BaseViewModel.Activator = new ViewModelActivator()[get]

Возвращает активатор.

См. определение в файле **BaseViewModel.cs** строка **47**

#### ObservableCollection<BaseViewModel> ViewModel.ViewModels.BaseViewModel.Dialogs[get]

Возвращает диалоги.

См. определение в файле **BaseViewModel.cs** строка **37**

#### ObservableCollection<BaseViewModel> ViewModel.ViewModels.BaseViewModel.Modals[get]

Возвращает модальные диалоги.

См. определение в файле **BaseViewModel.cs** строка **42**

## Класс ViewModel.Technicals.CalendarInterval

Класс интервала календаря.

### Открытые члены

* **CalendarInterval** (**ITimeIntervalElement** timeInterval, **ITaskElement** taskElement)

*Создаёт экземпляр класса* ***CalendarInterval****.*

### Свойства

* **ITimeIntervalElement** **TimeInterval** [get]

*Возвращает временной интервал.*

* **ITaskElement** **TaskElement** [get]

*Возвращает элементарную задачу.*

### Подробное описание

Класс интервала календаря.

См. определение в файле **CalendarInterval.cs** строка **8**

### Конструктор(ы)

#### ViewModel.Technicals.CalendarInterval.CalendarInterval (ITimeIntervalElement timeInterval, ITaskElement taskElement)

Создаёт экземпляр класса **CalendarInterval**.

##### Аргументы

|  |  |
| --- | --- |
| *timeInterval* | Временной интервал. |
| *taskElement* | Элементарная задача. |

См. определение в файле **CalendarInterval.cs** строка **25**

### Полный список свойств

#### ITaskElement ViewModel.Technicals.CalendarInterval.TaskElement[get]

Возвращает элементарную задачу.

См. определение в файле **CalendarInterval.cs** строка **18**

#### ITimeIntervalElement ViewModel.Technicals.CalendarInterval.TimeInterval[get]

Возвращает временной интервал.

См. определение в файле **CalendarInterval.cs** строка **13**

## Класс MachineLearning.LearningEvaluators.ClassificationCrossValidationEvaluator

Класс оценки модели обучения классификации методом кросс-валидации.

Граф наследования:MachineLearning.LearningEvaluators.ClassificationCrossValidationEvaluator:

### Открытые члены

* **ClassificationCrossValidationEvaluator** ()

*Создаёт экземпляр класса* ***ClassificationCrossValidationEvaluator*** *по умолчанию.*

#### Открытые члены унаследованные от MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >

* async Task< **ScoreMetricCategory** > **Evaluate** (IEnumerable< T > data, IEnumerable< R > targets)

*Оценивает модель обучения.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |
| *targets* | Целевые значения. |

##### *Возвращает*

Возвращает категорию модели обучения.

#### Открытые члены унаследованные от MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

* **BaseCrossValidationLearningEvaluator** ()

*Создаёт экземпляр класса* ***BaseCrossValidationLearningEvaluator*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* override IEnumerable< **ValidationFold** > **GetValidationFolds** (IEnumerable< IEnumerable< double > > data, IEnumerable< int > targets)

#### Защищенные члены унаследованные от MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >

* IEnumerable< **ValidationFold** > **GetValidationFolds** (IEnumerable< T > data, IEnumerable< R > targets)

*Возвращает сегменты валидации.*

* virtual **ScoreMetricCategory** **GetScoresCategory** (IEnumerable< double > scores)

*Определяет категорию оценок.*

* override void **OnPropertyChanged< T >** (T oldValue, T newValue)

*Вызывается при изменении свойства.*

##### *Параметры шаблона*

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### *Аргументы*

|  |  |
| --- | --- |
| *oldValue* | Старое значение. |
| *newValue* | Новое значение. |

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Дополнительные унаследованные члены

#### Свойства унаследованные от MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >

* ISupervisedLearningModel< T, R > **Model** [get, set]

*Возвращает и задаёт модель обучения с учителем.*

* ISupervisedScoreMetric< R > **ScoreMetric** [get, set]

*Возвращает и задаёт модель обучения с учителем.*

#### Свойства унаследованные от MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

* int **NumberOfFolds** [get, set]

*Возвращает и задаёт количество сегментов.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс оценки модели обучения классификации методом кросс-валидации.

Наследует **BaseSupervisedCrossValidationEvaluator<IEnumerable<double>, int>**. Реализует **IRegressionEvaluator**.

См. определение в файле **ClassificationCrossValidationEvaluator.cs** строка **13**

### Конструктор(ы)

#### MachineLearning.LearningEvaluators.ClassificationCrossValidationEvaluator.ClassificationCrossValidationEvaluator ()

Создаёт экземпляр класса **ClassificationCrossValidationEvaluator** по умолчанию.

См. определение в файле **ClassificationCrossValidationEvaluator.cs** строка **20**

### Методы

#### override IEnumerable< ValidationFold > MachineLearning.LearningEvaluators.ClassificationCrossValidationEvaluator.GetValidationFolds (IEnumerable< IEnumerable< double > > data, IEnumerable< int > targets)[protected]

См. определение в файле **ClassificationCrossValidationEvaluator.cs** строка **26**

## Класс MachineLearning.LearningEvaluators.ClusteringCrossValidationEvaluator

Класс оценки модели обучения кластеризации методом кросс-валидации.

Граф наследования:MachineLearning.LearningEvaluators.ClusteringCrossValidationEvaluator:

### Открытые члены

* **ClusteringCrossValidationEvaluator** ()

*Создаёт экземпляр класса* ***ClusteringCrossValidationEvaluator*** *по умолчанию.*

#### Открытые члены унаследованные от MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >

* async Task< **ScoreMetricCategory** > **Evaluate** (IEnumerable< T > data)

*Оценивает модель обучения.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает категорию модели обучения.

#### Открытые члены унаследованные от MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

* **BaseCrossValidationLearningEvaluator** ()

*Создаёт экземпляр класса* ***BaseCrossValidationLearningEvaluator*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* override IEnumerable< **ValidationFold** > **GetValidationFolds** (IEnumerable< IEnumerable< double > > data)

#### Защищенные члены унаследованные от MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >

* IEnumerable< **ValidationFold** > **GetValidationFolds** (IEnumerable< T > data)

*Возвращает сегменты валидации.*

* virtual **ScoreMetricCategory** **GetScoresCategory** (IEnumerable< double > scores)

*Определяет категорию оценок.*

* override void **OnPropertyChanged< T >** (T oldValue, T newValue)

*Вызывается при изменении свойства.*

##### *Параметры шаблона*

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### *Аргументы*

|  |  |
| --- | --- |
| *oldValue* | Старое значение. |
| *newValue* | Новое значение. |

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Дополнительные унаследованные члены

#### Свойства унаследованные от MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >

* IUnsupervisedLearningModel< T, R > **Model** [get, set]

*Возвращает и задаёт модель обучения без учителя.*

* IUnsupervisedScoreMetric< R, T > **ScoreMetric** [get, set]

*Возвращает и задаёт модель обучения без учителя.*

#### Свойства унаследованные от MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

* int **NumberOfFolds** [get, set]

*Возвращает и задаёт количество сегментов.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс оценки модели обучения кластеризации методом кросс-валидации.

Наследует **BaseUnsupervisedCrossValidationEvaluator<IEnumerable<double>, int>**. Реализует **IClusteringEvaluator**.

См. определение в файле **ClusteringCrossValidationEvaluator.cs** строка **13**

### Конструктор(ы)

#### MachineLearning.LearningEvaluators.ClusteringCrossValidationEvaluator.ClusteringCrossValidationEvaluator ()

Создаёт экземпляр класса **ClusteringCrossValidationEvaluator** по умолчанию.

См. определение в файле **ClusteringCrossValidationEvaluator.cs** строка **19**

### Методы

#### override IEnumerable< ValidationFold > MachineLearning.LearningEvaluators.ClusteringCrossValidationEvaluator.GetValidationFolds (IEnumerable< IEnumerable< double > > data)[protected]

См. определение в файле **ClusteringCrossValidationEvaluator.cs** строка **25**

## Класс View.Views.Modals.CopyTasksView

Класс пользовательского элемента диалога копирования задач.

Граф наследования:View.Views.Modals.CopyTasksView:

### Открытые члены

* **CopyTasksView** ()

*Создаёт экземпляр класса* ***CopyTasksView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента диалога копирования задач.

Наследует ReactiveUserControl<CopyTasksViewModel>.

См. определение в файле **CopyTasksView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Modals.CopyTasksView.CopyTasksView ()

Создаёт экземпляр класса **CopyTasksView** по умолчанию.

См. определение в файле **CopyTasksView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Modals.CopyTasksViewModel

Класс диалога копирования задач.

Граф наследования:ViewModel.ViewModels.Modals.CopyTasksViewModel:

### Защищенные члены

* override void **GetArgs** (**ItemsTasksViewModelArgs** args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.Modals.TasksViewModel< A, R >

* override void **GetArgs** (A args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* void **GetArgs** (A args)

*Получает аргументы, переданные в диалог.*

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.ViewModels.Modals.TasksViewModel< A, R >

* **TasksViewModel** ()

*Создаёт экземпляр класса* ***TasksViewModel<A, R>*** *по умолчанию.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* async Task< R > **Invoke** (**BaseViewModel** parent, A args)

*Вызывает диалог.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.Modals.TasksViewModel< A, R >

* IObservable< bool > **\_canExecuteGoToPrevious**

*Наблюдатель, который отслеживает возможность выполнения GoToPrevious.*

* IObservable< bool > **\_canExecuteGo**

*Наблюдатель, который отслеживает возможность выполнения Go.*

* IEnumerable< **ITask** > **\_mainList**

*Основной список.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* TaskCompletionSource< R >? **\_taskSource**

*Источник для завершения задачи, который используется для возврата результата диалога.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* **BaseViewModel**? **Parent** [get]

*Возвращает родителя.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс диалога копирования задач.

Наследует **TasksViewModel<ItemsTasksViewModelArgs, CopyTasksViewModelResult?>**.

См. определение в файле **CopyTasksViewModel.cs** строка **14**

### Методы

#### override void ViewModel.ViewModels.Modals.CopyTasksViewModel.GetArgs (ItemsTasksViewModelArgs args)[protected]

См. определение в файле **CopyTasksViewModel.cs** строка **30**

## Класс ViewModel.ViewModels.Modals.CopyTasksViewModelResult

Класс результата диалога **CopyTasksViewModel**.

### Открытые члены

* **CopyTasksViewModelResult** (IEnumerable< **ITask** > list, int count)

*Создаёт экземпляр класса* ***CopyTasksViewModelResult****.*

### Свойства

* IEnumerable< **ITask** > **List** [get]

*Возвращает список.*

* int **Count** [get]

*Возвращает количество.*

### Подробное описание

Класс результата диалога **CopyTasksViewModel**.

См. определение в файле **CopyTasksViewModelResult.cs** строка **8**

### Конструктор(ы)

#### ViewModel.ViewModels.Modals.CopyTasksViewModelResult.CopyTasksViewModelResult (IEnumerable< ITask > list, int count)

Создаёт экземпляр класса **CopyTasksViewModelResult**.

##### Аргументы

|  |  |
| --- | --- |
| *list* | Список. |
| *count* | Количество. |

См. определение в файле **CopyTasksViewModelResult.cs** строка **25**

### Полный список свойств

#### int ViewModel.ViewModels.Modals.CopyTasksViewModelResult.Count[get]

Возвращает количество.

См. определение в файле **CopyTasksViewModelResult.cs** строка **18**

#### IEnumerable<ITask> ViewModel.ViewModels.Modals.CopyTasksViewModelResult.List[get]

Возвращает список.

См. определение в файле **CopyTasksViewModelResult.cs** строка **13**

## Класс MachineLearning.DataProcessors.CorrelationColumnProcessor

Класс обработчика столбцов для устранения корреляции.

Граф наследования:MachineLearning.DataProcessors.CorrelationColumnProcessor:

### Открытые члены

* DataProcessorResult< IEnumerable< double > > **Process** (IEnumerable< IEnumerable< double > > data)

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IDataProcessor< T, R >

* DataProcessorResult< R > **Process** (IEnumerable< T > data)

*Преобразует данные.*

### Подробное описание

Класс обработчика столбцов для устранения корреляции.

Реализует **IPointDataProcessor**.

См. определение в файле **CorrelationColumnProcessor.cs** строка **14**

### Методы

#### DataProcessorResult< IEnumerable< double > > MachineLearning.DataProcessors.CorrelationColumnProcessor.Process (IEnumerable< IEnumerable< double > > data)

См. определение в файле **CorrelationColumnProcessor.cs** строка **22**

## Шаблон структуры MachineLearning.DataProcessorResult< T >

Структура хранения данных результата преобразования.

### Открытые члены

* **DataProcessorResult** (IEnumerable< T > result, IEnumerable< int >? removedColumnsIndices=null, IEnumerable< int >? removedRowsIndices=null)

*Создаёт экземпляр класса* ***DataProcessorResult<T>****.*

### Свойства

* IEnumerable< T > **Result** [get]

*Возвращает и задаёт результат преобразования.*

* IEnumerable< int > **RemovedColumnsIndices** [get]

*Возвращает и задаёт индексы удалённых столбцов.*

* IEnumerable< int > **RemovedRowsIndices** [get]

*Возвращает и задаёт индексы удалённых строк.*

### Подробное описание

Структура хранения данных результата преобразования.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип данных. |

См. определение в файле **DataProcessorResult.cs** строка **7**

### Методы

#### MachineLearning.DataProcessorResult< T >.DataProcessorResult (IEnumerable< T > result, IEnumerable< int >? removedColumnsIndices = null, IEnumerable< int >? removedRowsIndices = null)

Создаёт экземпляр класса **DataProcessorResult<T>**.

##### Аргументы

|  |  |
| --- | --- |
| *result* | Результат преобразования. |
| *removedColumnsIndices* | Индексы удалённых столбцов. |
| *removedRowsIndices* | Индексы удалённых строк. |

См. определение в файле **DataProcessorResult.cs** строка **30**

### Полный список свойств

#### IEnumerable<int> MachineLearning.DataProcessorResult< T >.RemovedColumnsIndices[get]

Возвращает и задаёт индексы удалённых столбцов.

См. определение в файле **DataProcessorResult.cs** строка **17**

#### IEnumerable<int> MachineLearning.DataProcessorResult< T >.RemovedRowsIndices[get]

Возвращает и задаёт индексы удалённых строк.

См. определение в файле **DataProcessorResult.cs** строка **22**

#### IEnumerable<T> MachineLearning.DataProcessorResult< T >.Result[get]

Возвращает и задаёт результат преобразования.

См. определение в файле **DataProcessorResult.cs** строка **12**

## Класс View.Controls.DateTimePicker

Класс пользовательского элемента для выбора даты и времени.

Граф наследования:View.Controls.DateTimePicker:

### Открытые члены

* **DateTimePicker** ()

*Создаёт экземпляр класса* ***DateTimePicker*** *по умолчанию.*

### Статические открытые данные

* static readonly StyledProperty< **DateTime**?> **DateTimeProperty**

*Свойство стиля* ***DateTime****.*

* static readonly StyledProperty< **TimeSpan**?> **TimeSpanProperty**

*Свойство стиля* ***TimeSpan****.*

### Свойства

* DateTime? **DateTime** [get, set]

*Возвращает и задаёт дату.*

* TimeSpan? **TimeSpan** [get, set]

*Возвращает и задаёт время.*

### Подробное описание

Класс пользовательского элемента для выбора даты и времени.

Наследует UserControl.

См. определение в файле **DateTimePicker.axaml.cs** строка **14**

### Конструктор(ы)

#### View.Controls.DateTimePicker.DateTimePicker ()

Создаёт экземпляр класса **DateTimePicker** по умолчанию.

См. определение в файле **DateTimePicker.axaml.cs** строка **51**

### Данные класса

#### readonly StyledProperty<DateTime?> View.Controls.DateTimePicker.DateTimeProperty[static]

**Инициализатор**

=

AvaloniaProperty.Register<DateTimePicker, DateTime?>(nameof(DateTime),

defaultBindingMode: BindingMode.TwoWay)

Свойство стиля **DateTime**.

См. определение в файле **DateTimePicker.axaml.cs** строка **19**

#### readonly StyledProperty<TimeSpan?> View.Controls.DateTimePicker.TimeSpanProperty[static]

**Инициализатор**

=

AvaloniaProperty.Register<DateTimePicker, TimeSpan?>(nameof(TimeSpan),

defaultBindingMode: BindingMode.TwoWay)

Свойство стиля **TimeSpan**.

См. определение в файле **DateTimePicker.axaml.cs** строка **26**

### Полный список свойств

#### DateTime? View.Controls.DateTimePicker.DateTime[get], [set]

Возвращает и задаёт дату.

См. определение в файле **DateTimePicker.axaml.cs** строка **33**

#### TimeSpan? View.Controls.DateTimePicker.TimeSpan[get], [set]

Возвращает и задаёт время.

См. определение в файле **DateTimePicker.axaml.cs** строка **42**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.DbContextFactory

Класс фабрики, создающая контексты базы данных.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.DbContextFactory:

### Открытые члены

* **BaseDbContext** **Create** ()

*Создает новый экземпляр объекта.*

##### *Возвращает*

Возвращает новый экземпляр объекта.

### Свойства

* string **ConnectionString** [get, set]

*Возвращает и задаёт строку подключения.*

### Подробное описание

Класс фабрики, создающая контексты базы данных.

Реализует <see cref="IDbContextFactory{**BaseDbContext**}/>.

См. определение в файле **DbContextFactory.cs** строка **9**

### Методы

#### BaseDbContext ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.DbContextFactory.Create ()

Создает новый экземпляр объекта.

##### Возвращает

Возвращает новый экземпляр объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >** (*стр.245*).

### Полный список свойств

#### string ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.DbContextFactory.ConnectionString[get], [set]

Возвращает и задаёт строку подключения.

Замещает **ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.IDbContextFactory< T >** (*стр.238*).

См. определение в файле **DbContextFactory.cs** строка **12**

## Класс ViewModel.Implementations.AppStates.Sessions.DbSession

Класс сессии базы данных.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.DbSession:

### Открытые члены

* **DbSession** (IDbContextFactory< **BaseDbContext** > contextFactory, IMapper< **TaskEntity**, **ITask** > taskMapper, IMapper< **TimeIntervalEntity**, **ITimeIntervalElement** > timeIntervalMapper, **IFileService** fileService)

*Создаёт экземпляр класса* ***DbSession****.*

* async Task **Load** ()

*Загружает данные из хранилища.*

##### *Возвращает*

Возвращает задачц процесса загрузки данных.

* async Task **Save** ()

*Сохраняет данные в хранилище.*

##### *Возвращает*

Возвращает задачц процесса сохранения данных.

* void **AddTasks** (IEnumerable< **ITask** > tasks, **ITaskComposite**? parentTask)

*Добавляет задачи.*

##### *Аргументы*

|  |  |
| --- | --- |
| *tasks* | Задачи. |
| *parentTask* | Родительская задача. |

* void **EditTask** (**ITask** task)

*Изменяет задачу.*

##### *Аргументы*

|  |  |
| --- | --- |
| *task* | Задача. |

* void **RemoveTasks** (IEnumerable< **ITask** > tasks)

*Удаляет задачи.*

##### *Аргументы*

|  |  |
| --- | --- |
| *tasks* | Задачи. |

* void **MoveTasks** (IEnumerable< **ITask** > tasks, **ITaskComposite**? parentTask)

*Перемещает задачи.*

##### *Аргументы*

|  |  |
| --- | --- |
| *tasks* | Задачи. |
| *parentTask* | Родительская задача. |

* void **AddTimeInterval** (**ITimeIntervalElement** timeIntervalElement, **ITaskElement** taskElement)

*Добавляет временной интервал.*

##### *Аргументы*

|  |  |
| --- | --- |
| *timeIntervalElement* | Элементарный временной интервал. |
| *taskElement* | Элементарная задача. |

* void **EditTimeInterval** (**ITimeIntervalElement** timeIntervalElement)

*Изменяет временной интервал.*

##### *Аргументы*

|  |  |
| --- | --- |
| *timeIntervalElement* | Элементарный временной интервал. |

* void **RemoveTimeInterval** (**ITimeIntervalElement** timeIntervalElement, **ITaskElement** taskElement)

*Удаляет временной интервал.*

##### *Аргументы*

|  |  |
| --- | --- |
| *timeIntervalElement* | Элементарный временной интервал. |
| *taskElement* | Элементарная задача. |

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Свойства

* string **SavePath** [get, set]

*Возвращает и задаёт путь сохранения.*

* IEnumerable< **ITask** > **Tasks** [get]

*Возвращает задачи.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### События

* EventHandler< **ItemsUpdatedEventArgs** > **ItemsUpdated**

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

#### События унаследованные от ViewModel.Interfaces.AppStates.Sessions.ISession

* EventHandler< **ItemsUpdatedEventArgs** > **ItemsUpdated**

*Событие, которое возникает при обновлении данных.*

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Подробное описание

Класс сессии базы данных.

Наследует **TrackableObject**. Реализует **ISession**.

См. определение в файле **DbSession.cs** строка **23**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.Sessions.DbSession.DbSession (IDbContextFactory< BaseDbContext > contextFactory, IMapper< TaskEntity, ITask > taskMapper, IMapper< TimeIntervalEntity, ITimeIntervalElement > timeIntervalMapper, IFileService fileService)

Создаёт экземпляр класса **DbSession**.

##### Аргументы

|  |  |
| --- | --- |
| *contextFactory* | Фабрика, создающая контексты базы данных. |
| *taskMapper* | Преобразование значений между сущностью задачи и задачей. |
| *timeIntervalMapper* | Преобразование значений между сущностью временного интервала и элементарным временным интервалом. |
| *fileService* | Файловый сервис. |

См. определение в файле **DbSession.cs** строка **90**

### Методы

#### void ViewModel.Implementations.AppStates.Sessions.DbSession.AddTasks (IEnumerable< ITask > tasks, ITaskComposite? parentTask)

Добавляет задачи.

##### Аргументы

|  |  |
| --- | --- |
| *tasks* | Задачи. |
| *parentTask* | Родительская задача. |

Замещает **ViewModel.Interfaces.AppStates.Sessions.ISession** (*стр.279*).

См. определение в файле **DbSession.cs** строка **126**

#### void ViewModel.Implementations.AppStates.Sessions.DbSession.AddTimeInterval (ITimeIntervalElement timeIntervalElement, ITaskElement taskElement)

Добавляет временной интервал.

##### Аргументы

|  |  |
| --- | --- |
| *timeIntervalElement* | Элементарный временной интервал. |
| *taskElement* | Элементарная задача. |

Замещает **ViewModel.Interfaces.AppStates.Sessions.ISession** (*стр.279*).

См. определение в файле **DbSession.cs** строка **203**

#### void ViewModel.Implementations.AppStates.Sessions.DbSession.EditTask (ITask task)

Изменяет задачу.

##### Аргументы

|  |  |
| --- | --- |
| *task* | Задача. |

Замещает **ViewModel.Interfaces.AppStates.Sessions.ISession** (*стр.279*).

См. определение в файле **DbSession.cs** строка **140**

#### void ViewModel.Implementations.AppStates.Sessions.DbSession.EditTimeInterval (ITimeIntervalElement timeIntervalElement)

Изменяет временной интервал.

##### Аргументы

|  |  |
| --- | --- |
| *timeIntervalElement* | Элементарный временной интервал. |

Замещает **ViewModel.Interfaces.AppStates.Sessions.ISession** (*стр.280*).

См. определение в файле **DbSession.cs** строка **219**

#### async Task ViewModel.Implementations.AppStates.Sessions.DbSession.Load ()

Загружает данные из хранилища.

##### Возвращает

Возвращает задачц процесса загрузки данных.

Замещает **ViewModel.Interfaces.AppStates.IStorageService** (*стр.283*).

См. определение в файле **DbSession.cs** строка **102**

#### void ViewModel.Implementations.AppStates.Sessions.DbSession.MoveTasks (IEnumerable< ITask > tasks, ITaskComposite? parentTask)

Перемещает задачи.

##### Аргументы

|  |  |
| --- | --- |
| *tasks* | Задачи. |
| *parentTask* | Родительская задача. |

Замещает **ViewModel.Interfaces.AppStates.Sessions.ISession** (*стр.280*).

См. определение в файле **DbSession.cs** строка **181**

#### void ViewModel.Implementations.AppStates.Sessions.DbSession.RemoveTasks (IEnumerable< ITask > tasks)

Удаляет задачи.

##### Аргументы

|  |  |
| --- | --- |
| *tasks* | Задачи. |

Замещает **ViewModel.Interfaces.AppStates.Sessions.ISession** (*стр.280*).

См. определение в файле **DbSession.cs** строка **153**

#### void ViewModel.Implementations.AppStates.Sessions.DbSession.RemoveTimeInterval (ITimeIntervalElement timeIntervalElement, ITaskElement taskElement)

Удаляет временной интервал.

##### Аргументы

|  |  |
| --- | --- |
| *timeIntervalElement* | Элементарный временной интервал. |
| *taskElement* | Элементарная задача. |

Замещает **ViewModel.Interfaces.AppStates.Sessions.ISession** (*стр.280*).

См. определение в файле **DbSession.cs** строка **228**

#### async Task ViewModel.Implementations.AppStates.Sessions.DbSession.Save ()

Сохраняет данные в хранилище.

##### Возвращает

Возвращает задачц процесса сохранения данных.

Замещает **ViewModel.Interfaces.AppStates.IStorageService** (*стр.283*).

См. определение в файле **DbSession.cs** строка **120**

### Полный список свойств

#### string ViewModel.Implementations.AppStates.Sessions.DbSession.SavePath[get], [set]

Возвращает и задаёт путь сохранения.

Замещает **ViewModel.Interfaces.AppStates.Sessions.ISession** (*стр.280*).

См. определение в файле **DbSession.cs** строка **62**

#### IEnumerable<ITask> ViewModel.Implementations.AppStates.Sessions.DbSession.Tasks[get]

Возвращает задачи.

Замещает **ViewModel.Interfaces.AppStates.Sessions.ISession** (*стр.281*).

См. определение в файле **DbSession.cs** строка **69**

### Cобытия

#### EventHandler<ItemsUpdatedEventArgs> ViewModel.Implementations.AppStates.Sessions.DbSession.ItemsUpdated

См. определение в файле **DbSession.cs** строка **76**

## Класс ViewModel.Implementations.ModelLearning.DeadlineTaskElementEvaluatorLearningController

Класс контроллера обучения модели обучения с учителем для предсказания срока элементарной задачи.

Граф наследования:ViewModel.Implementations.ModelLearning.DeadlineTaskElementEvaluatorLearningController:

### Открытые члены

* **DeadlineTaskElementEvaluatorLearningController** (**DeadlineTaskElementLearningConverter** converter, **IRegressionModel** model, **IRegressionEvaluator** evaluator)

*Создаёт экземпляр классса* ***DeadlineTaskElementEvaluatorLearningController****.*

#### Открытые члены унаследованные от ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >

* **BaseSupervisedEvaluatorLearningController** (ISupervisedLearningConverter< T, R, D, DT, DR > converter, ISupervisedLearningModel< T, R > model, ISupervisedLearningEvaluator< T, R > evaluator)

*Создаёт экземпляр классса* ***BaseSupervisedEvaluatorLearningController<T, R, D, DT, DR>****.*

* async Task< bool > **Train** (IEnumerable< D > data)

*Выполняет обучение модели на основе данных.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает задачу процесса обучения с результатом true , если обучение прошло успешно, иначе false .

* DR **Predict** (DT data)

*Выполняет предсказание на основе данных.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает предсказанное значение.

### Дополнительные унаследованные члены

#### Свойства унаследованные от ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >

* **ScoreMetricCategory** **MinScoreCategory** [get, set]

*Возвращает и задаёт минимальную категорию метрики оценки.*

* bool **IsValidModel** [get, protected set]

*Возвращает логическое значение, указывающее на достоверность модели обучения.*

### Подробное описание

Класс контроллера обучения модели обучения с учителем для предсказания срока элементарной задачи.

Наследует **BaseSupervisedEvaluatorLearningController<IEnumerable<double>, double, ITaskElement, ITaskElement, DateTime?>**.

См. определение в файле **DeadlineTaskElementEvaluatorLearningController.cs** строка **17**

### Конструктор(ы)

#### ViewModel.Implementations.ModelLearning.DeadlineTaskElementEvaluatorLearningController.DeadlineTaskElementEvaluatorLearningController (DeadlineTaskElementLearningConverter converter, IRegressionModel model, IRegressionEvaluator evaluator)

Создаёт экземпляр классса **DeadlineTaskElementEvaluatorLearningController**.

##### Аргументы

|  |  |
| --- | --- |
| *converter* | Конвертор данных в данные для предсказания с учителем и наоборот. |
| *model* | Модель обучения c учителем. |
| *evaluator* | Оценка модели обучения с учителем. |

См. определение в файле **DeadlineTaskElementEvaluatorLearningController.cs** строка **29**

## Класс ViewModel.Implementations.ModelLearning.Converters.DeadlineTaskElementLearningConverter

Класс конвертора элементраных задач в данные для предсказания срока с учителем и наоборот.

Граф наследования:ViewModel.Implementations.ModelLearning.Converters.DeadlineTaskElementLearningConverter:

### Открытые члены

* **DeadlineTaskElementLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors, IFactory< **IScaler** > scalerFactory, IDataTransformer< **TaskMetadata**, int?> metadataCategoriesTransformer, IDataTransformer< **TaskMetadata**, IEnumerable< int > > metadataTagsITransformer)

*Создаёт экземпляр класса* ***DeadlineTaskElementLearningConverter****.*

* override? DateTime **ConvertPredicted** (double predicted)

#### Открытые члены унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* LearningModelData< IEnumerable< double >, R > **FitConvertData** (IEnumerable< D > data)

*Адаптирует конвертацию данных под данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные данные для предсказания.

#### Открытые члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

##### *Аргументы*

|  |  |
| --- | --- |
| *predicted* | Предсказанное значение. |

##### *Возвращает*

Возвращает конвертированное выходное значение.

* IEnumerable< double > **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные входные данные для предсказания.

### Защищенные члены

* override List< double?> **ExtractPrimaryFeatures** (**ITaskElement** dataItem)
* override double **ProcessTarget** (**ITaskElement** item)

#### Защищенные члены унаследованные от ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >

* **BaseTaskElementSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors, IFactory< **IScaler** > scalerFactory, IDataTransformer< **TaskMetadata**, int?> metadataCategoriesTransformer, IDataTransformer< **TaskMetadata**, IEnumerable< int > > metadataTagsITransformer)

*Создаёт экземпляр класса* ***BaseTaskElementSupervisedLearningConverter<R, DR>****.*

* override IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< **ITaskElement** > data)
* override IEnumerable< double?> **ExtractFeatures** (**ITaskElement** dataItem)
* override **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)
* List< double?> **ExtractPrimaryFeatures** (**ITaskElement** dataItem)

*Извлекает первичные признаки у элемента данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* **BaseSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors)

*Создаёт экземпляр класса* ***BaseSupervisedLearningConverter<R, D, DT, DR>****.*

* R **ProcessTarget** (D item)

*Обрабатывает целевые значения элемента.*

* IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)

*Создаёт масштабирование данных.*

* DataProcessorResult< IEnumerable< double > > **ProcessFeaturesData** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* void **CreateScalers** (double[][] featuresArray)

*Создаёт коллекцию масштабирования данных.*

* IEnumerable< int > **NormalizeRemovedIndices** (IEnumerable< IEnumerable< int > > removedIndicesGroups)

*Нормализует индексы удалённых индексов.*

* IEnumerable< R > **ProcessTargets** (IEnumerable< D > data, IEnumerable< int > removedRowsIndices)

*Обрабатывает целевые значения данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* **BaseLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor)

*Создаёт экземпляр класса* ***BaseLearningConverter<R, D, DT, DR>****.*

* IEnumerable< double?> **ExtractFeatures** (DT dataItem)

*Извлекает признаки у элемента данных.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >

* readonly IFactory< **IScaler** > **\_scalerFactory**

*Фабрика, создающая масштабирования данных.*

* readonly IDataTransformer< **TaskMetadata**, int?> **\_metadataCategoriesTransformer**

*Преобразование категории метаданных в данные для предсказания.*

* readonly IDataTransformer< **TaskMetadata**, IEnumerable< int > > **\_metadataTagsTransformer**

*Преобразование теги метаданных в данные для предсказания.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* readonly IEnumerable< **IPointDataProcessor** > **\_pointDataProcessors**

*Обработчики точечных данных.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* readonly **IPrimaryPointDataProcessor** **\_primaryPointDataProcessor**

*Первичный обработчик точечных данных.*

* IEnumerable< int >? **\_removedColumnsIndices**

*Индексы удалённых столбцов.*

* IEnumerable< **IScaler** >? **\_scalers**

*Коллекция масштабирования данных.*

### Подробное описание

Класс конвертора элементраных задач в данные для предсказания срока с учителем и наоборот.

Наследует **BaseTaskElementSupervisedLearningConverter<double, DateTime?>**.

См. определение в файле **DeadlineTaskElementLearningConverter.cs** строка **17**

### Конструктор(ы)

#### ViewModel.Implementations.ModelLearning.Converters.DeadlineTaskElementLearningConverter.DeadlineTaskElementLearningConverter (IPrimaryPointDataProcessor primaryPointDataProcessor, IEnumerable< IPointDataProcessor > pointDataProcessors, IFactory< IScaler > scalerFactory, IDataTransformer< TaskMetadata, int?> metadataCategoriesTransformer, IDataTransformer< TaskMetadata, IEnumerable< int > > metadataTagsITransformer)

Создаёт экземпляр класса **DeadlineTaskElementLearningConverter**.

##### Аргументы

|  |  |
| --- | --- |
| *primaryPointDataProcessor* | Первичный обработчик точечных данных. |
| *pointDataProcessors* | Обработчики точечных данных. |
| *scalerFactory* | Фабрика, создающая масштабирования данных. |
| *metadataCategoriesTransformer* | Преобразование категории метаданных в данные для предсказания. |
| *metadataTagsITransformer* | Преобразование теги метаданных в данные для предсказания. |

См. определение в файле **DeadlineTaskElementLearningConverter.cs** строка **32**

## Класс View.Extensions.DesignExtension

Класс расширений синтаксиса разметки для дизайнера, который использует контейнер зависимости.

Граф наследования:View.Extensions.DesignExtension:

### Открытые члены

* **DesignExtension** (Type type)

*Создаёт экземпляр класса* ***EnumsExtension****.*

* override object **ProvideValue** (IServiceProvider serviceProvider)

### Подробное описание

Класс расширений синтаксиса разметки для дизайнера, который использует контейнер зависимости.

Наследует MarkupExtension.

См. определение в файле **DesignExtension.cs** строка **18**

### Конструктор(ы)

#### View.Extensions.DesignExtension.DesignExtension (Type type)

Создаёт экземпляр класса **EnumsExtension**.

##### Аргументы

|  |  |
| --- | --- |
| *type* | Тип. |

##### Исключения

|  |  |
| --- | --- |
| *ArgumentException* | Выбрасывает, если тип отсутствует. |

См. определение в файле **DesignExtension.cs** строка **37**

## Класс View.Desktop.Linux.DesktopAppLifeState

Класс для управления состоянием жизненного цикла desktop-приложения.

Граф наследования:View.Desktop.Linux.DesktopAppLifeState:

### Открытые члены

* **DesktopAppLifeState** ()

*Создаёт экземпляр класса* ***DesktopAppLifeState*** *по умолчанию.*

### События

* EventHandler **AppDeactivated**

#### События унаследованные от ViewModel.Interfaces.AppStates.IAppLifeState

* EventHandler **AppDeactivated**

*Событие, которое возникает, когда приложение деактивируется.*

### Подробное описание

Класс для управления состоянием жизненного цикла desktop-приложения.

Реализует **IAppLifeState**.

См. определение в файле **DesktopAppLifeState.cs** строка **13**

### Конструктор(ы)

#### View.Desktop.Linux.DesktopAppLifeState.DesktopAppLifeState ()

Создаёт экземпляр класса **DesktopAppLifeState** по умолчанию.

См. определение в файле **DesktopAppLifeState.cs** строка **21**

### Cобытия

#### EventHandler View.Desktop.Linux.DesktopAppLifeState.AppDeactivated

См. определение в файле **DesktopAppLifeState.cs** строка **16**

## Класс View.Desktop.macOS.DesktopAppLifeState

Класс для управления состоянием жизненного цикла desktop-приложения.

Граф наследования:View.Desktop.macOS.DesktopAppLifeState:

### Открытые члены

* **DesktopAppLifeState** ()

*Создаёт экземпляр класса* ***DesktopAppLifeState*** *по умолчанию.*

### События

* EventHandler **AppDeactivated**

#### События унаследованные от ViewModel.Interfaces.AppStates.IAppLifeState

* EventHandler **AppDeactivated**

*Событие, которое возникает, когда приложение деактивируется.*

### Подробное описание

Класс для управления состоянием жизненного цикла desktop-приложения.

Реализует **IAppLifeState**.

См. определение в файле **DesktopAppLifeState.cs** строка **13**

### Конструктор(ы)

#### View.Desktop.macOS.DesktopAppLifeState.DesktopAppLifeState ()

Создаёт экземпляр класса **DesktopAppLifeState** по умолчанию.

См. определение в файле **DesktopAppLifeState.cs** строка **21**

### Cобытия

#### EventHandler View.Desktop.macOS.DesktopAppLifeState.AppDeactivated

См. определение в файле **DesktopAppLifeState.cs** строка **16**

## Класс View.Desktop.Windows.DesktopAppLifeState

Класс для управления состоянием жизненного цикла desktop-приложения.

Граф наследования:View.Desktop.Windows.DesktopAppLifeState:

### Открытые члены

* **DesktopAppLifeState** ()

*Создаёт экземпляр класса* ***DesktopAppLifeState*** *по умолчанию.*

### События

* EventHandler **AppDeactivated**

#### События унаследованные от ViewModel.Interfaces.AppStates.IAppLifeState

* EventHandler **AppDeactivated**

*Событие, которое возникает, когда приложение деактивируется.*

### Подробное описание

Класс для управления состоянием жизненного цикла desktop-приложения.

Реализует **IAppLifeState**.

См. определение в файле **DesktopAppLifeState.cs** строка **13**

### Конструктор(ы)

#### View.Desktop.Windows.DesktopAppLifeState.DesktopAppLifeState ()

Создаёт экземпляр класса **DesktopAppLifeState** по умолчанию.

См. определение в файле **DesktopAppLifeState.cs** строка **21**

### Cобытия

#### EventHandler View.Desktop.Windows.DesktopAppLifeState.AppDeactivated

См. определение в файле **DesktopAppLifeState.cs** строка **16**

## Класс MachineLearning.DataProcessors.DuplicatesRowProcessor

Класс обработчика строк для устранения дубликатов.

Граф наследования:MachineLearning.DataProcessors.DuplicatesRowProcessor:

### Открытые члены

* DataProcessorResult< IEnumerable< double > > **Process** (IEnumerable< IEnumerable< double > > data)

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IDataProcessor< T, R >

* DataProcessorResult< R > **Process** (IEnumerable< T > data)

*Преобразует данные.*

### Подробное описание

Класс обработчика строк для устранения дубликатов.

Реализует **IPointDataProcessor**.

См. определение в файле **DublicatesRowProcessor.cs** строка **11**

### Методы

#### DataProcessorResult< IEnumerable< double > > MachineLearning.DataProcessors.DuplicatesRowProcessor.Process (IEnumerable< IEnumerable< double > > data)

См. определение в файле **DublicatesRowProcessor.cs** строка **14**

## Класс View.Views.Pages.EditorView

Класс пользовательского элемента страницы изменения задач.

Граф наследования:View.Views.Pages.EditorView:

### Открытые члены

* **EditorView** ()

*Создаёт экземпляр класса* ***EditorView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента страницы изменения задач.

Наследует ReactiveUserControl<EditorViewModel>.

См. определение в файле **EditorView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Pages.EditorView.EditorView ()

Создаёт экземпляр класса **EditorView** по умолчанию.

См. определение в файле **EditorView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Pages.EditorViewModel

Класс контроллера страницы изменения задач.

Граф наследования:ViewModel.ViewModels.Pages.EditorViewModel:

### Открытые члены

* **EditorViewModel** (**ISession** session, IResourceService resourceService, BaseDialogViewModel< IList< **ITask** >, bool > removeTasksDialog, BaseDialogViewModel< **ITask**, bool > addTasksDialog, BaseDialogViewModel< object, bool > editTaskDialog, BaseDialogViewModel< **ItemsTasksViewModelArgs**, IEnumerable< **ITask** >?> moveTasksDialog, BaseDialogViewModel< **ItemsTasksViewModelArgs**, **CopyTasksViewModelResult**?> copyTasksDialog, IFactory< **ITaskComposite** > taskCompositeFactory, IFactory< **ITaskElementProxy** > taskElementProxyFactory, **ITasksEditorProxy** tasksEditorProxy, **ITaskElementsEditorProxy** taskElementsEditorProxy)

*Создаёт экземпляр класса* ***EditorViewModel****.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BasePageViewModel

* object **Metadata** [get, protected set]

*Возращает метаданные.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс контроллера страницы изменения задач.

Наследует **BasePageViewModel**.

См. определение в файле **EditorViewModel.cs** строка **23**

### Конструктор(ы)

#### ViewModel.ViewModels.Pages.EditorViewModel.EditorViewModel (ISession session, IResourceService resourceService, BaseDialogViewModel< IList< ITask >, bool > removeTasksDialog, BaseDialogViewModel< ITask, bool > addTasksDialog, BaseDialogViewModel< object, bool > editTaskDialog, BaseDialogViewModel< ItemsTasksViewModelArgs, IEnumerable< ITask >?> moveTasksDialog, BaseDialogViewModel< ItemsTasksViewModelArgs, CopyTasksViewModelResult?> copyTasksDialog, IFactory< ITaskComposite > taskCompositeFactory, IFactory< ITaskElementProxy > taskElementProxyFactory, ITasksEditorProxy tasksEditorProxy, ITaskElementsEditorProxy taskElementsEditorProxy)

Создаёт экземпляр класса **EditorViewModel**.

##### Аргументы

|  |  |
| --- | --- |
| *session* | Сессия. |
| *resourceService* | Сервис ресурсов. |
| *removeTasksDialog* | Диалог удаления задач. |
| *addTasksDialog* | Диалог добавления задач. |
| *editTaskDialog* | Диалог изменения задачи. |
| *moveTasksDialog* | Диалог перемещения задач. |
| *copyTasksDialog* | Диалог копирования задач. |
| *taskCompositeFactory* | Фабрика, создающая составную задачу. |
| *taskElementProxyFactory* | Фабрика, создающая элементарную задачу. |
| *tasksEditorProxy* | Заместитель задач для редактирования. |
| *taskElementsEditorProxy* | Заместитель элементарных задач для редактирования. |

См. определение в файле **EditorViewModel.cs** строка **144**

## Класс View.Views.Modals.EditTaskView

Класс пользовательского элемента диалога изменения задачи.

Граф наследования:View.Views.Modals.EditTaskView:

### Открытые члены

* **EditTaskView** ()

*Создаёт экземпляр класса* ***EditTaskView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента диалога изменения задачи.

Наследует ReactiveUserControl<EditTaskViewModel>.

См. определение в файле **EditTaskView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Modals.EditTaskView.EditTaskView ()

Создаёт экземпляр класса **EditTaskView** по умолчанию.

См. определение в файле **EditTaskView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Modals.EditTaskViewModel

Класс диалога изменения задачи.

Граф наследования:ViewModel.ViewModels.Modals.EditTaskViewModel:

### Открытые члены

* **EditTaskViewModel** ()

*Создаёт экземпляр класса* ***EditTaskViewModel*** *по умолчанию.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* async Task< R > **Invoke** (**BaseViewModel** parent, A args)

*Вызывает диалог.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Защищенные члены

* override void **GetArgs** (object args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* void **GetArgs** (A args)

*Получает аргументы, переданные в диалог.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* TaskCompletionSource< R >? **\_taskSource**

*Источник для завершения задачи, который используется для возврата результата диалога.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* **BaseViewModel**? **Parent** [get]

*Возвращает родителя.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс диалога изменения задачи.

Наследует **BaseDialogViewModel<object, bool>**.

См. определение в файле **EditTaskViewModel.cs** строка **14**

### Конструктор(ы)

#### ViewModel.ViewModels.Modals.EditTaskViewModel.EditTaskViewModel ()

Создаёт экземпляр класса **EditTaskViewModel** по умолчанию.

См. определение в файле **EditTaskViewModel.cs** строка **30**

## Класс View.Views.Modals.EditTimeIntervalView

Класс пользовательского элемента диалога изменения временного интервала.

Граф наследования:View.Views.Modals.EditTimeIntervalView:

### Открытые члены

* **EditTimeIntervalView** ()

*Создаёт экземпляр класса* ***EditTimeIntervalView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента диалога изменения временного интервала.

Наследует ReactiveUserControl<EditTimeIntervalViewModel>.

См. определение в файле **EditTimeIntervalView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Modals.EditTimeIntervalView.EditTimeIntervalView ()

Создаёт экземпляр класса **EditTimeIntervalView** по умолчанию.

См. определение в файле **EditTimeIntervalView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Modals.EditTimeIntervalViewModel

Класс диалога изменения временного интервала.

Граф наследования:ViewModel.ViewModels.Modals.EditTimeIntervalViewModel:

### Открытые члены

* **EditTimeIntervalViewModel** ()

*Создаёт экземпляр класса* ***EditTimeIntervalViewModel*** *по умолчанию.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* async Task< R > **Invoke** (**BaseViewModel** parent, A args)

*Вызывает диалог.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Защищенные члены

* override void **GetArgs** (**ITimeIntervalElement** args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* void **GetArgs** (A args)

*Получает аргументы, переданные в диалог.*

### Свойства

* **ITimeIntervalElement** **TimeIntervalElement** [get]

*Возвращает элементарнай временной интервал.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* **BaseViewModel**? **Parent** [get]

*Возвращает родителя.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* TaskCompletionSource< R >? **\_taskSource**

*Источник для завершения задачи, который используется для возврата результата диалога.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

### Подробное описание

Класс диалога изменения временного интервала.

Наследует **BaseDialogViewModel<ITimeIntervalElement, bool>**.

См. определение в файле **EditTimeIntervalViewModel.cs** строка **16**

### Конструктор(ы)

#### ViewModel.ViewModels.Modals.EditTimeIntervalViewModel.EditTimeIntervalViewModel ()

Создаёт экземпляр класса **EditTimeIntervalViewModel** по умолчанию.

См. определение в файле **EditTimeIntervalViewModel.cs** строка **31**

### Методы

#### override void ViewModel.ViewModels.Modals.EditTimeIntervalViewModel.GetArgs (ITimeIntervalElement args)[protected]

См. определение в файле **EditTimeIntervalViewModel.cs** строка **46**

### Полный список свойств

#### ITimeIntervalElement ViewModel.ViewModels.Modals.EditTimeIntervalViewModel.TimeIntervalElement[get]

Возвращает элементарнай временной интервал.

См. определение в файле **EditTimeIntervalViewModel.cs** строка **26**

## Класс View.Extensions.EnumsExtension

Класс расширений синтаксиса разметки для перечислений.

Граф наследования:View.Extensions.EnumsExtension:

### Открытые члены

* **EnumsExtension** (Type type)

*Создаёт экземпляр класса* ***EnumsExtension****.*

* override object **ProvideValue** (IServiceProvider serviceProvider)

### Подробное описание

Класс расширений синтаксиса разметки для перечислений.

Наследует MarkupExtension.

См. определение в файле **EnumsExtension.cs** строка **12**

### Конструктор(ы)

#### View.Extensions.EnumsExtension.EnumsExtension (Type type)

Создаёт экземпляр класса **EnumsExtension**.

##### Аргументы

|  |  |
| --- | --- |
| *type* | Тип перечисления. |

##### Исключения

|  |  |
| --- | --- |
| *ArgumentException* | Выбрасывает, если тип не является перечислением или отсутствует. |

См. определение в файле **EnumsExtension.cs** строка **26**

## Класс MachineLearning.DistanceMetrics.EuclideanDistanceMetric

Класс метрики Евклидового расстояния.

Граф наследования:MachineLearning.DistanceMetrics.EuclideanDistanceMetric:

### Открытые члены

* double **CalculateDistance** (IEnumerable< double > object1, IEnumerable< double > object2)

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IDistanceMetric< T >

* double **CalculateDistance** (T object1, T object2)

*Вычисляет расстояние между двумя объектами.*

### Подробное описание

Класс метрики Евклидового расстояния.

Реализует **IPointDistanceMetric**.

См. определение в файле **EuclideanDistanceMetric.cs** строка **11**

### Методы

#### double MachineLearning.DistanceMetrics.EuclideanDistanceMetric.CalculateDistance (IEnumerable< double > object1, IEnumerable< double > object2)

См. определение в файле **EuclideanDistanceMetric.cs** строка **14**

## Класс ViewModel.Implementations.ModelLearning.ExecutionChanceTaskElementEvaluatorLearningController

Класс контроллера обучения модели обучения с учителем для предсказания шанса выполнения элементарной задачи.

Граф наследования:ViewModel.Implementations.ModelLearning.ExecutionChanceTaskElementEvaluatorLearningController:

### Открытые члены

* **ExecutionChanceTaskElementEvaluatorLearningController** (**ExecutionChanceTaskElementLearningConverter** converter, **IRegressionModel** model, **IRegressionEvaluator** evaluator)

*Создаёт экземпляр классса* ***ExecutionChanceTaskElementEvaluatorLearningController****.*

#### Открытые члены унаследованные от ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >

* **BaseSupervisedEvaluatorLearningController** (ISupervisedLearningConverter< T, R, D, DT, DR > converter, ISupervisedLearningModel< T, R > model, ISupervisedLearningEvaluator< T, R > evaluator)

*Создаёт экземпляр классса* ***BaseSupervisedEvaluatorLearningController<T, R, D, DT, DR>****.*

* async Task< bool > **Train** (IEnumerable< D > data)

*Выполняет обучение модели на основе данных.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает задачу процесса обучения с результатом true , если обучение прошло успешно, иначе false .

* DR **Predict** (DT data)

*Выполняет предсказание на основе данных.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает предсказанное значение.

### Дополнительные унаследованные члены

#### Свойства унаследованные от ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >

* **ScoreMetricCategory** **MinScoreCategory** [get, set]

*Возвращает и задаёт минимальную категорию метрики оценки.*

* bool **IsValidModel** [get, protected set]

*Возвращает логическое значение, указывающее на достоверность модели обучения.*

### Подробное описание

Класс контроллера обучения модели обучения с учителем для предсказания шанса выполнения элементарной задачи.

Наследует **BaseSupervisedEvaluatorLearningController<IEnumerable<double>, double, ITaskElement, ITaskElement, double>**.

См. определение в файле **ExecutionChanceTaskElementEvaluatorLearningController.cs** строка **17**

### Конструктор(ы)

#### ViewModel.Implementations.ModelLearning.ExecutionChanceTaskElementEvaluatorLearningController.ExecutionChanceTaskElementEvaluatorLearningController (ExecutionChanceTaskElementLearningConverter converter, IRegressionModel model, IRegressionEvaluator evaluator)

Создаёт экземпляр классса **ExecutionChanceTaskElementEvaluatorLearningController**.

##### Аргументы

|  |  |
| --- | --- |
| *converter* | Конвертор данных в данные для предсказания с учителем и наоборот. |
| *model* | Модель обучения c учителем. |
| *evaluator* | Оценка модели обучения с учителем. |

См. определение в файле **ExecutionChanceTaskElementEvaluatorLearningController.cs** строка **30**

## Класс ViewModel.Implementations.ModelLearning.Converters.ExecutionChanceTaskElementLearningConverter

Класс конвертора элементраных задач в данные для предсказания шанс выполнения с учителем и наоборот.

Граф наследования:ViewModel.Implementations.ModelLearning.Converters.ExecutionChanceTaskElementLearningConverter:

### Открытые члены

* **ExecutionChanceTaskElementLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors, IFactory< **IScaler** > scalerFactory, IDataTransformer< **TaskMetadata**, int?> metadataCategoriesTransformer, IDataTransformer< **TaskMetadata**, IEnumerable< int > > metadataTagsITransformer)

*Создаёт экземпляр класса* ***ExecutionChanceTaskElementLearningConverter****.*

* override double **ConvertPredicted** (double predicted)

#### Открытые члены унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* LearningModelData< IEnumerable< double >, R > **FitConvertData** (IEnumerable< D > data)

*Адаптирует конвертацию данных под данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные данные для предсказания.

#### Открытые члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

##### *Аргументы*

|  |  |
| --- | --- |
| *predicted* | Предсказанное значение. |

##### *Возвращает*

Возвращает конвертированное выходное значение.

* IEnumerable< double > **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные входные данные для предсказания.

### Защищенные члены

* override List< double?> **ExtractPrimaryFeatures** (**ITaskElement** dataItem)
* override double **ProcessTarget** (**ITaskElement** item)

#### Защищенные члены унаследованные от ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >

* **BaseTaskElementSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors, IFactory< **IScaler** > scalerFactory, IDataTransformer< **TaskMetadata**, int?> metadataCategoriesTransformer, IDataTransformer< **TaskMetadata**, IEnumerable< int > > metadataTagsITransformer)

*Создаёт экземпляр класса* ***BaseTaskElementSupervisedLearningConverter<R, DR>****.*

* override IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< **ITaskElement** > data)
* override IEnumerable< double?> **ExtractFeatures** (**ITaskElement** dataItem)
* override **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)
* List< double?> **ExtractPrimaryFeatures** (**ITaskElement** dataItem)

*Извлекает первичные признаки у элемента данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* **BaseSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors)

*Создаёт экземпляр класса* ***BaseSupervisedLearningConverter<R, D, DT, DR>****.*

* R **ProcessTarget** (D item)

*Обрабатывает целевые значения элемента.*

* IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)

*Создаёт масштабирование данных.*

* DataProcessorResult< IEnumerable< double > > **ProcessFeaturesData** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* void **CreateScalers** (double[][] featuresArray)

*Создаёт коллекцию масштабирования данных.*

* IEnumerable< int > **NormalizeRemovedIndices** (IEnumerable< IEnumerable< int > > removedIndicesGroups)

*Нормализует индексы удалённых индексов.*

* IEnumerable< R > **ProcessTargets** (IEnumerable< D > data, IEnumerable< int > removedRowsIndices)

*Обрабатывает целевые значения данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* **BaseLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor)

*Создаёт экземпляр класса* ***BaseLearningConverter<R, D, DT, DR>****.*

* IEnumerable< double?> **ExtractFeatures** (DT dataItem)

*Извлекает признаки у элемента данных.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >

* readonly IFactory< **IScaler** > **\_scalerFactory**

*Фабрика, создающая масштабирования данных.*

* readonly IDataTransformer< **TaskMetadata**, int?> **\_metadataCategoriesTransformer**

*Преобразование категории метаданных в данные для предсказания.*

* readonly IDataTransformer< **TaskMetadata**, IEnumerable< int > > **\_metadataTagsTransformer**

*Преобразование теги метаданных в данные для предсказания.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* readonly IEnumerable< **IPointDataProcessor** > **\_pointDataProcessors**

*Обработчики точечных данных.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* readonly **IPrimaryPointDataProcessor** **\_primaryPointDataProcessor**

*Первичный обработчик точечных данных.*

* IEnumerable< int >? **\_removedColumnsIndices**

*Индексы удалённых столбцов.*

* IEnumerable< **IScaler** >? **\_scalers**

*Коллекция масштабирования данных.*

### Подробное описание

Класс конвертора элементраных задач в данные для предсказания шанс выполнения с учителем и наоборот.

Наследует **BaseTaskElementSupervisedLearningConverter<double, double>**.

См. определение в файле **ExecutionChanceTaskElementLearningConverter.cs** строка **20**

### Конструктор(ы)

#### ViewModel.Implementations.ModelLearning.Converters.ExecutionChanceTaskElementLearningConverter.ExecutionChanceTaskElementLearningConverter (IPrimaryPointDataProcessor primaryPointDataProcessor, IEnumerable< IPointDataProcessor > pointDataProcessors, IFactory< IScaler > scalerFactory, IDataTransformer< TaskMetadata, int?> metadataCategoriesTransformer, IDataTransformer< TaskMetadata, IEnumerable< int > > metadataTagsITransformer)

Создаёт экземпляр класса **ExecutionChanceTaskElementLearningConverter**.

##### Аргументы

|  |  |
| --- | --- |
| *primaryPointDataProcessor* | Первичный обработчик точечных данных. |
| *pointDataProcessors* | Обработчики точечных данных. |
| *scalerFactory* | Фабрика, создающая масштабирования данных. |
| *metadataCategoriesTransformer* | Преобразование категории метаданных в данные для предсказания. |
| *metadataTagsITransformer* | Преобразование теги метаданных в данные для предсказания. |

См. определение в файле **ExecutionChanceTaskElementLearningConverter.cs** строка **35**

## Класс MachineLearning.ScoreMetrics.F1ScoreMetric

Класс метрики F1 оценки для модели обучения классификации.

Граф наследования:MachineLearning.ScoreMetrics.F1ScoreMetric:

### Открытые члены

* double **CalculateScore** (IEnumerable< int > actual, IEnumerable< int > predicted)
* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

##### *Аргументы*

|  |  |
| --- | --- |
| *score* | Оценка. |

##### *Возвращает*

Возвращает категорию оценки.

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T >

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< T > predicted)

*Вычисляет оценку на основе фактических и предсказанных данных.*

### Подробное описание

Класс метрики F1 оценки для модели обучения классификации.

Реализует **IClassificationScoreMetric**.

См. определение в файле **F1ScoreMetric.cs** строка **13**

### Методы

#### double MachineLearning.ScoreMetrics.F1ScoreMetric.CalculateScore (IEnumerable< int > actual, IEnumerable< int > predicted)

См. определение в файле **F1ScoreMetric.cs** строка **16**

#### ScoreMetricCategory MachineLearning.ScoreMetrics.F1ScoreMetric.GetScoreCategory (double score)

Определяет категорию оценки.

##### Аргументы

|  |  |
| --- | --- |
| *score* | Оценка. |

##### Возвращает

Возвращает категорию оценки.

Замещает **MachineLearning.Interfaces.Generals.IScoreMetric** (*стр.275*).

## Класс ViewModel.Implementations.AppStates.FileLogger

Класс сервиса логгирования сообщений в файл.

Граф наследования:ViewModel.Implementations.AppStates.FileLogger:

### Открытые члены

* **FileLogger** (**IFileService** fileService)

*Создаёт экземпляр класса* ***FileLogger****.*

* void **Log** (string message)

*Записывает сообщение в лог.*

##### *Аргументы*

|  |  |
| --- | --- |
| *message* | Сообщение. |

### Свойства

* string **FilePath** [get, set]

*Возвращает и задаёт путь к файлу.*

### Подробное описание

Класс сервиса логгирования сообщений в файл.

Реализует **ILogger**.

См. определение в файле **FileLogger.cs** строка **11**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.FileLogger.FileLogger (IFileService fileService)

Создаёт экземпляр класса **FileLogger**.

##### Аргументы

|  |  |
| --- | --- |
| *fileService* | Файловый сервис. |

См. определение в файле **FileLogger.cs** строка **37**

### Методы

#### void ViewModel.Implementations.AppStates.FileLogger.Log (string message)

Записывает сообщение в лог.

##### Аргументы

|  |  |
| --- | --- |
| *message* | Сообщение. |

Замещает **ViewModel.Interfaces.AppStates.ILogger** (*стр.257*).

См. определение в файле **FileLogger.cs** строка **46**

### Полный список свойств

#### string ViewModel.Implementations.AppStates.FileLogger.FilePath[get], [set]

Возвращает и задаёт путь к файлу.

См. определение в файле **FileLogger.cs** строка **21**

## Класс ViewModel.Implementations.AppStates.FileService

Класс файлового сервиса.

Граф наследования:ViewModel.Implementations.AppStates.FileService:

### Открытые члены

* Task< byte[]> **Load** (string path)

*Загружает данные из файла по пути.*

##### *Аргументы*

|  |  |
| --- | --- |
| *path* | Путь. |

##### *Возвращает*

Возвращает данные.

* Task **Save** (string path, byte[] data)

*Загружает данные из файла по пути.*

##### *Аргументы*

|  |  |
| --- | --- |
| *path* | Путь. |
| *data* | Данные. |

##### *Возвращает*

* Stream **CreateStream** (string path, FileMode mode, FileAccess access, FileShare share)

*Создаёт поток.*

##### *Аргументы*

|  |  |
| --- | --- |
| *path* | Путь. |
| *mode* | Режим. |
| *access* | Доступ к использованию этого потока. |
| *share* | Доступ к использованию другим потокам. |

##### *Возвращает*

Возвращает поток.

* void **CreateDirectory** (string path)

*Создаёт директорию по пути.*

##### *Аргументы*

|  |  |
| --- | --- |
| *path* | Путь. |

* string? **GetDirectoryPath** (string path)

*Получает путь к директории.*

##### *Аргументы*

|  |  |
| --- | --- |
| *path* | Путь. |

##### *Возвращает*

* string **CombinePath** (string path1, string path2)

*Объединяет пути.*

##### *Аргументы*

|  |  |
| --- | --- |
| *path1* | Первый путь. |
| *path2* | Второй путь. |

##### *Возвращает*

Возвращает объединённый путь.

* bool **IsPathExists** (string path)

*Проверяет, существует ли заданный путь.*

##### *Аргументы*

|  |  |
| --- | --- |
| *path* | Путь. |

##### *Возвращает*

Возвращает true , если путь существует, иначе false .

### Свойства

* string **PersonalDirectoryPath** [get]

*Возвращает путь к персональной директории.*

### Подробное описание

Класс файлового сервиса.

Реализует **IFileService**.

См. определение в файле **FileService.cs** строка **11**

### Методы

#### string ViewModel.Implementations.AppStates.FileService.CombinePath (string path1, string path2)

Объединяет пути.

##### Аргументы

|  |  |
| --- | --- |
| *path1* | Первый путь. |
| *path2* | Второй путь. |

##### Возвращает

Возвращает объединённый путь.

Замещает **ViewModel.Interfaces.AppStates.IFileService** (*стр.248*).

#### void ViewModel.Implementations.AppStates.FileService.CreateDirectory (string path)

Создаёт директорию по пути.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |

Замещает **ViewModel.Interfaces.AppStates.IFileService** (*стр.248*).

#### Stream ViewModel.Implementations.AppStates.FileService.CreateStream (string path, FileMode mode, FileAccess access, FileShare share)

Создаёт поток.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |
| *mode* | Режим. |
| *access* | Доступ к использованию этого потока. |
| *share* | Доступ к использованию другим потокам. |

##### Возвращает

Возвращает поток.

Замещает **ViewModel.Interfaces.AppStates.IFileService** (*стр.248*).

#### string? ViewModel.Implementations.AppStates.FileService.GetDirectoryPath (string path)

Получает путь к директории.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |

##### Возвращает

Замещает **ViewModel.Interfaces.AppStates.IFileService** (*стр.248*).

#### bool ViewModel.Implementations.AppStates.FileService.IsPathExists (string path)

Проверяет, существует ли заданный путь.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |

##### Возвращает

Возвращает true , если путь существует, иначе false .

Замещает **ViewModel.Interfaces.AppStates.IFileService** (*стр.249*).

#### Task< byte[]> ViewModel.Implementations.AppStates.FileService.Load (string path)

Загружает данные из файла по пути.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |

##### Возвращает

Возвращает данные.

Замещает **ViewModel.Interfaces.AppStates.IFileService** (*стр.249*).

#### Task ViewModel.Implementations.AppStates.FileService.Save (string path, byte[] data)

Загружает данные из файла по пути.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |
| *data* | Данные. |

##### Возвращает

Замещает **ViewModel.Interfaces.AppStates.IFileService** (*стр.249*).

### Полный список свойств

#### string ViewModel.Implementations.AppStates.FileService.PersonalDirectoryPath[get]

Возвращает путь к персональной директории.

Замещает **ViewModel.Interfaces.AppStates.IFileService** (*стр.249*).

См. определение в файле **FileService.cs** строка **21**

## Класс View.Converters.FormattableToStringConverter

Класс конвертора форматируемого объекта в строку.

Граф наследования:View.Converters.FormattableToStringConverter:

### Открытые члены

* object? **Convert** (object? value, Type targetType, object? parameter, CultureInfo culture)
* object? **ConvertBack** (object? value, Type targetType, object? parameter, CultureInfo culture)

### Подробное описание

Класс конвертора форматируемого объекта в строку.

Реализует IValueConverter.

См. определение в файле **FormattableToStringConverter.cs** строка **13**

### Методы

#### object? View.Converters.FormattableToStringConverter.Convert (object? value, Type targetType, object? parameter, CultureInfo culture)

См. определение в файле **FormattableToStringConverter.cs** строка **16**

## Интерфейс ViewModel.Interfaces.AppStates.IAppLifeState

Интерфейс для управления состоянием жизненного цикла приложения.

Граф наследования:ViewModel.Interfaces.AppStates.IAppLifeState:

### События

* EventHandler **AppDeactivated**

*Событие, которое возникает, когда приложение деактивируется.*

### Подробное описание

Интерфейс для управления состоянием жизненного цикла приложения.

См. определение в файле **IAppLifeState.cs** строка **6**

### Cобытия

#### EventHandler ViewModel.Interfaces.AppStates.IAppLifeState.AppDeactivated

Событие, которое возникает, когда приложение деактивируется.

См. определение в файле **IAppLifeState.cs** строка **11**

## Интерфейс MachineLearning.Interfaces.IClassificationEvaluator

Интерфейс оценки модели обучения классификации.

Граф наследования:MachineLearning.Interfaces.IClassificationEvaluator:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >

* Task< **ScoreMetricCategory** > **Evaluate** (IEnumerable< T > data, IEnumerable< R > targets)

*Оценивает модель обучения.*

#### Свойства унаследованные от MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >

* ISupervisedLearningModel< T, R > **Model** [get, set]

*Возвращает и задаёт модель обучения с учителем.*

* ISupervisedScoreMetric< R > **ScoreMetric** [get, set]

*Возвращает и задаёт метрику оценки обучения с учителем.*

### Подробное описание

Интерфейс оценки модели обучения классификации.

Наследует **ISupervisedLearningEvaluator<IEnumerable<double>, int>**.

См. определение в файле **IClassificationEvaluator.cs** строка **11**

## Интерфейс MachineLearning.Interfaces.IClassificationModel

Интерфейс модели обучения классификации.

Граф наследования:MachineLearning.Interfaces.IClassificationModel:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedLearningModel< T, R >

* Task **Train** (IEnumerable< T > data, IEnumerable< R > targets)

*Обучает модель на основе данных и целевых значений.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ILearningModel< T, R >

* R **Predict** (T data)

*Предсказывает значение на основе данных.*

### Подробное описание

Интерфейс модели обучения классификации.

Наследует **ISupervisedLearningModel<IEnumerable<double>, int>**.

См. определение в файле **IClassificationModel.cs** строка **11**

## Интерфейс MachineLearning.Interfaces.IClassificationScoreMetric

Интерфейс метрики оценки для модели обучения классификации.

Граф наследования:MachineLearning.Interfaces.IClassificationScoreMetric:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T >

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< T > predicted)

*Вычисляет оценку на основе фактических и предсказанных данных.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IScoreMetric

* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

### Подробное описание

Интерфейс метрики оценки для модели обучения классификации.

Наследует **ISupervisedScoreMetric<int>**.

См. определение в файле **IClassificationScoreMetric.cs** строка **11**

## Интерфейс MachineLearning.Interfaces.IClusteringEvaluator

Интерфейс оценки модели обучения кластеризации.

Граф наследования:MachineLearning.Interfaces.IClusteringEvaluator:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R >

* Task< **ScoreMetricCategory** > **Evaluate** (IEnumerable< T > data)

*Оценивает модель обучения.*

#### Свойства унаследованные от MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R >

* IUnsupervisedLearningModel< T, R > **Model** [get, set]

*Возвращает и задаёт модель обучения без учителя.*

* IUnsupervisedScoreMetric< R, T > **ScoreMetric** [get, set]

*Возвращает и задаёт метрику оценки обучения с учителем.*

### Подробное описание

Интерфейс оценки модели обучения кластеризации.

Наследует **IUnsupervisedLearningEvaluator<IEnumerable<double>, int>**.

См. определение в файле **IClusteringEvaluator.cs** строка **11**

## Интерфейс MachineLearning.Interfaces.IClusteringModel

Интерфейс модели обучения кластеризации.

Граф наследования:MachineLearning.Interfaces.IClusteringModel:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IUnsupervisedLearningModel< T, R >

* Task **Train** (IEnumerable< T > data)

*Обучает модель на основе данных.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ILearningModel< T, R >

* R **Predict** (T data)

*Предсказывает значение на основе данных.*

### Подробное описание

Интерфейс модели обучения кластеризации.

Наследует **IUnsupervisedLearningModel<IEnumerable<double>, int>**.

См. определение в файле **IClusteringModel.cs** строка **11**

## Интерфейс MachineLearning.Interfaces.IClusteringScoreMetric

Интерфейс метрики оценки для модели обучения кластеризации.

Граф наследования:MachineLearning.Interfaces.IClusteringScoreMetric:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IUnsupervisedScoreMetric< T, D >

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< D > data)

*Вычисляет оценку на основе фактических данных и данных для предсказания.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IScoreMetric

* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

### Подробное описание

Интерфейс метрики оценки для модели обучения кластеризации.

Наследует **IUnsupervisedScoreMetric<int, IEnumerable<double>>**.

См. определение в файле **IClusteringScoreMetric.cs** строка **11**

## Шаблон интерфейса MachineLearning.Interfaces.Generals.IDataProcessor< T, R >

Интерфейс обработчика данных.

Граф наследования:MachineLearning.Interfaces.Generals.IDataProcessor< T, R >:

### Открытые члены

* DataProcessorResult< R > **Process** (IEnumerable< T > data)

*Преобразует данные.*

### Подробное описание

Интерфейс обработчика данных.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных. |
| *R* | Тип выходных данных. |

См. определение в файле **IDataProcessor.cs** строка **8**

### Методы

#### DataProcessorResult< R > MachineLearning.Interfaces.Generals.IDataProcessor< T, R >.Process (IEnumerable< T > data)

Преобразует данные.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает данные о преобразовании.

## Шаблон интерфейса ViewModel.Interfaces.ModelLearning.IDataTransformer< T, R >

Интерфейс для преобразования данных из одного типа в другой.

Граф наследования:ViewModel.Interfaces.ModelLearning.IDataTransformer< T, R >:

### Открытые члены

* IEnumerable< R > **FitTransform** (IEnumerable< T > data)

*Адаптирует преобразование под данные.*

* R **Transform** (T data)

*Преобразует данные.*

### Подробное описание

Интерфейс для преобразования данных из одного типа в другой.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных. |
| *R* | Тип выходных данных. |

См. определение в файле **IDataTransformer.cs** строка **8**

### Методы

#### IEnumerable< R > ViewModel.Interfaces.ModelLearning.IDataTransformer< T, R >.FitTransform (IEnumerable< T > data)

Адаптирует преобразование под данные.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает преобразованные данные.

#### R ViewModel.Interfaces.ModelLearning.IDataTransformer< T, R >.Transform (T data)

Преобразует данные.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает преобразованные данные.

## Шаблон интерфейса ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.IDbContextFactory< T >

Интерфейс фабрики, создающая контексты базы данных.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.IDbContextFactory< T >:

### Свойства

* string **ConnectionString** [get, set]

*Возвращает и задаёт строку подключения.*

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >

* T **Create** ()

*Создает новый экземпляр объекта.*

### Подробное описание

Интерфейс фабрики, создающая контексты базы данных.

Наследует **IFactory<T>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип данных. |

См. определение в файле **IDbContextFactory.cs** строка **12**

### Полный список свойств

#### string ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.IDbContextFactory< T >.ConnectionString[get], [set]

Возвращает и задаёт строку подключения.

Замещается в **ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.DbContextFactory** (*стр.183*).

См. определение в файле **IDbContextFactory.cs** строка **17**

## Шаблон интерфейса MachineLearning.Interfaces.Generals.IDistanceMetric< T >

Интерфейс метрики расстояния.

Граф наследования:MachineLearning.Interfaces.Generals.IDistanceMetric< T >:

### Открытые члены

* double **CalculateDistance** (T object1, T object2)

*Вычисляет расстояние между двумя объектами.*

### Подробное описание

Интерфейс метрики расстояния.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных. |

См. определение в файле **IDistanceMetric.cs** строка **7**

### Методы

#### double MachineLearning.Interfaces.Generals.IDistanceMetric< T >.CalculateDistance (T object1, T object2)

Вычисляет расстояние между двумя объектами.

##### Аргументы

|  |  |
| --- | --- |
| *object1* | Первый объект. |
| *object2* | Второй объект. |

##### Возвращает

Возвращает расстояние.

## Интерфейс ViewModel.Implementations.AppStates.Sessions.Database.Domains.IDomain

Индерфейс доменной модели.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.Domains.IDomain:

### Свойства

* object **EntityId** [get]

*Возвращает индетификатор связанной сущности.*

### Подробное описание

Индерфейс доменной модели.

См. определение в файле **IDomain.cs** строка **6**

### Полный список свойств

#### object ViewModel.Implementations.AppStates.Sessions.Database.Domains.IDomain.EntityId[get]

Возвращает индетификатор связанной сущности.

Замещается в **ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain** (*стр.438*), **ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskElementDomain** (*стр.461*) и **ViewModel.Implementations.AppStates.Sessions.Database.Domains.TimeIntervalElementDomain** (*стр.496*).

См. определение в файле **IDomain.cs** строка **11**

## Шаблон интерфейса ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T >

Интерфейс заменителя для изменения данных.

Граф наследования:ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T >:

### Свойства

* new T **Target** [get, set]

*Возвращает и задаёт заменяемого объекта.*

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.Interfaces.DataManagers.Generals.IEditorService

* void **ApplyChanges** ()

*Применения изменений.*

### Подробное описание

Интерфейс заменителя для изменения данных.

Наследует **IEditorService** и **IProxy<T>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип заменяемого данных. |

См. определение в файле **IEditorProxy.cs** строка **10**

### Полный список свойств

#### new T ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T >.Target[get], [set]

Возвращает и задаёт заменяемого объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IProxy< out T >** (*стр.268*).

Замещается в **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.485*) и **ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy** (*стр.500*).

См. определение в файле **IEditorProxy.cs** строка **15**

## Интерфейс ViewModel.Interfaces.DataManagers.Generals.IEditorService

Интерфейс сервиса изменения.

Граф наследования:ViewModel.Interfaces.DataManagers.Generals.IEditorService:

### Открытые члены

* void **ApplyChanges** ()

*Применения изменений.*

### Подробное описание

Интерфейс сервиса изменения.

См. определение в файле **IEditorService.cs** строка **6**

### Методы

#### void ViewModel.Interfaces.DataManagers.Generals.IEditorService.ApplyChanges ()

Применения изменений.

Замещается в **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.470*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.483*) и **ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy** (*стр.499*).

## Шаблон интерфейса ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >

Интерфейс фабрики, создающая объекты.

Граф наследования:ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >:

### Открытые члены

* T **Create** ()

*Создает новый экземпляр объекта.*

### Подробное описание

Интерфейс фабрики, создающая объекты.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип данных создаваемых объектов. |

См. определение в файле **IFactory.cs** строка **7**

### Методы

#### T ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >.Create ()

Создает новый экземпляр объекта.

##### Возвращает

Возвращает новый экземпляр объекта.

Замещается в **ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.DbContextFactory** (*стр.182*), **ViewModel.Implementations.DataManagers.Factories.TaskCompositeFactory** (*стр.441*), **ViewModel.Implementations.DataManagers.Factories.TaskElementCreatorProxyFactory** (*стр.458*), **ViewModel.Implementations.DataManagers.Factories.TaskElementFactory** (*стр.466*), **ViewModel.Implementations.DataManagers.Factories.TaskMetadataFactory** (*стр.481*), **ViewModel.Implementations.DataManagers.Factories.TimeIntervalElementFactory** (*стр.497*) и **ViewModel.Implementations.ModelLearning.Converters.ScalerFactory** (*стр.409*).

## Интерфейс ViewModel.Interfaces.AppStates.IFileService

Интерфейс сервиса файлов.

Граф наследования:ViewModel.Interfaces.AppStates.IFileService:

### Открытые члены

* Task **Save** (string path, byte[] data)

*Загружает данные из файла по пути.*

* Task< byte[]> **Load** (string path)

*Загружает данные из файла по пути.*

* Stream **CreateStream** (string path, FileMode mode, FileAccess access, FileShare share)

*Создаёт поток.*

* void **CreateDirectory** (string path)

*Создаёт директорию по пути.*

* string **CombinePath** (string path1, string path2)

*Объединяет пути.*

* string? **GetDirectoryPath** (string path)

*Получает путь к директории.*

* bool **IsPathExists** (string path)

*Проверяет, существует ли заданный путь.*

### Свойства

* string **PersonalDirectoryPath** [get]

*Возвращает путь к персональной директории.*

### Подробное описание

Интерфейс сервиса файлов.

См. определение в файле **IFileService.cs** строка **6**

### Методы

#### string ViewModel.Interfaces.AppStates.IFileService.CombinePath (string path1, string path2)

Объединяет пути.

##### Аргументы

|  |  |
| --- | --- |
| *path1* | Первый путь. |
| *path2* | Второй путь. |

##### Возвращает

Возвращает объединённый путь.

Замещается в **ViewModel.Implementations.AppStates.FileService** (*стр.224*).

#### void ViewModel.Interfaces.AppStates.IFileService.CreateDirectory (string path)

Создаёт директорию по пути.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |

Замещается в **ViewModel.Implementations.AppStates.FileService** (*стр.225*).

#### Stream ViewModel.Interfaces.AppStates.IFileService.CreateStream (string path, FileMode mode, FileAccess access, FileShare share)

Создаёт поток.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |
| *mode* | Режим. |
| *access* | Доступ к использованию этого потока. |
| *share* | Доступ к использованию другим потокам. |

##### Возвращает

Возвращает поток.

Замещается в **ViewModel.Implementations.AppStates.FileService** (*стр.225*).

#### string? ViewModel.Interfaces.AppStates.IFileService.GetDirectoryPath (string path)

Получает путь к директории.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |

##### Возвращает

Замещается в **ViewModel.Implementations.AppStates.FileService** (*стр.225*).

#### bool ViewModel.Interfaces.AppStates.IFileService.IsPathExists (string path)

Проверяет, существует ли заданный путь.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |

##### Возвращает

Возвращает true , если путь существует, иначе false .

Замещается в **ViewModel.Implementations.AppStates.FileService** (*стр.225*).

#### Task< byte[]> ViewModel.Interfaces.AppStates.IFileService.Load (string path)

Загружает данные из файла по пути.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |

##### Возвращает

Возвращает данные.

Замещается в **ViewModel.Implementations.AppStates.FileService** (*стр.226*).

#### Task ViewModel.Interfaces.AppStates.IFileService.Save (string path, byte[] data)

Загружает данные из файла по пути.

##### Аргументы

|  |  |
| --- | --- |
| *path* | Путь. |
| *data* | Данные. |

##### Возвращает

Замещается в **ViewModel.Implementations.AppStates.FileService** (*стр.226*).

### Полный список свойств

#### string ViewModel.Interfaces.AppStates.IFileService.PersonalDirectoryPath[get]

Возвращает путь к персональной директории.

Замещается в **ViewModel.Implementations.AppStates.FileService** (*стр.226*).

См. определение в файле **IFileService.cs** строка **11**

## Шаблон интерфейса ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR >

Интерфейс контроллера обучения модели обучения.

Граф наследования:ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR >:

### Открытые члены

* DR **Predict** (DT data)

*Выполняет предсказание на основе данных.*

#### Открытые члены унаследованные от ViewModel.Interfaces.ModelLearning.IModelTeacher< D >

* Task< bool > **Train** (IEnumerable< D > data)

*Выполняет обучение модели на основе данных.*

### Свойства

* bool **IsValidModel** [get]

*Возвращает логическое значение, указывающее на достоверность модели обучения.*

### Подробное описание

Интерфейс контроллера обучения модели обучения.

Наследует **IModelTeacher<D>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *D* | Тип данных. |
| *DT* | Тип входных данных. |
| *DR* | Тип выходных данных. |

См. определение в файле **ILearningController.cs** строка **12**

### Методы

#### DR ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR >.Predict (DT data)

Выполняет предсказание на основе данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает предсказанное значение.

Замещается в **ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >** (*стр.136*).

### Полный список свойств

#### bool ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR >.IsValidModel[get]

Возвращает логическое значение, указывающее на достоверность модели обучения.

Замещается в **ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >** (*стр.137*).

См. определение в файле **ILearningController.cs** строка **17**

## Шаблон интерфейса MachineLearning.Interfaces.Generals.ILearningConverter< T, R, DT, DR >

Интерфейс конвертора данных в данные для предсказания и наоборот.

Граф наследования:MachineLearning.Interfaces.Generals.ILearningConverter< T, R, DT, DR >:

### Открытые члены

* T **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

### Подробное описание

Интерфейс конвертора данных в данные для предсказания и наоборот.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |
| *DT* | Тип входных данных. |
| *DR* | Тип выходных данных. |

См. определение в файле **ILearningConverter.cs** строка **10**

### Методы

#### T MachineLearning.Interfaces.Generals.ILearningConverter< T, R, DT, DR >.ConvertData (DT data)

Конвертирует входные данные под входные данные для предсказания.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает конвертированные входные данные для предсказания.

Замещается в **MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >** (*стр.126*).

#### DR MachineLearning.Interfaces.Generals.ILearningConverter< T, R, DT, DR >.ConvertPredicted (R predicted)

Конвертирует предсказанное значение в выходное значение.

##### Аргументы

|  |  |
| --- | --- |
| *predicted* | Предсказанное значение. |

##### Возвращает

Возвращает конвертированное выходное значение.

Замещается в **MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >** (*стр.127*).

## Интерфейс MachineLearning.Interfaces.Generals.ILearningEvaluator

Интерфейс оценки модели обучения.

Граф наследования:MachineLearning.Interfaces.Generals.ILearningEvaluator:

### Подробное описание

Интерфейс оценки модели обучения.

См. определение в файле **ILearningEvaluator.cs** строка **6**

## Шаблон интерфейса MachineLearning.Interfaces.Generals.ILearningModel< T, R >

Интерфейс модели обучения.

Граф наследования:MachineLearning.Interfaces.Generals.ILearningModel< T, R >:

### Открытые члены

* R **Predict** (T data)

*Предсказывает значение на основе данных.*

### Подробное описание

Интерфейс модели обучения.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |

См. определение в файле **ILearningModel.cs** строка **8**

### Методы

#### R MachineLearning.Interfaces.Generals.ILearningModel< T, R >.Predict (T data)

Предсказывает значение на основе данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает предсказанное значение.

## Интерфейс ViewModel.Interfaces.AppStates.Settings.ILocalizationManager

Интерфейс менеджера локализаций.

Граф наследования:ViewModel.Interfaces.AppStates.Settings.ILocalizationManager:

### Свойства

* IEnumerable< CultureInfo > **Localizations** [get]

*Возвращает локализации.*

* CultureInfo **ActualLocalization** [get, set]

*Возвращает и задаёт актуальную локализацию.*

### Подробное описание

Интерфейс менеджера локализаций.

См. определение в файле **ILocalizationManager.cs** строка **8**

### Полный список свойств

#### CultureInfo ViewModel.Interfaces.AppStates.Settings.ILocalizationManager.ActualLocalization[get], [set]

Возвращает и задаёт актуальную локализацию.

Замещается в **View.Implementations.AvaloniaLocalizationManager** (*стр.111*) и **ViewModel.Implementations.Mocks.MockLocalizationManager** (*стр.368*).

См. определение в файле **ILocalizationManager.cs** строка **18**

#### IEnumerable<CultureInfo> ViewModel.Interfaces.AppStates.Settings.ILocalizationManager.Localizations[get]

Возвращает локализации.

Замещается в **View.Implementations.AvaloniaLocalizationManager** (*стр.111*) и **ViewModel.Implementations.Mocks.MockLocalizationManager** (*стр.368*).

См. определение в файле **ILocalizationManager.cs** строка **13**

## Интерфейс ViewModel.Interfaces.AppStates.ILogger

Интерфейс сервиса логирования сообщений.

Граф наследования:ViewModel.Interfaces.AppStates.ILogger:

### Открытые члены

* void **Log** (string message)

*Записывает сообщение в лог.*

### Подробное описание

Интерфейс сервиса логирования сообщений.

См. определение в файле **ILogger.cs** строка **6**

### Методы

#### void ViewModel.Interfaces.AppStates.ILogger.Log (string message)

Записывает сообщение в лог.

##### Аргументы

|  |  |
| --- | --- |
| *message* | Сообщение. |

Замещается в **ViewModel.Implementations.AppStates.FileLogger** (*стр.222*).

## Шаблон интерфейса ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 >

Интерфейс для преобразования значений между двумя предметными областями.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 >:

### Открытые члены

* T2 **Map** (T1 value)

*Преобразует исходные данные в целевые данные.*

* T1 **MapBack** (T2 value)

*Преобразует целевые данные в исходные данные.*

### Подробное описание

Интерфейс для преобразования значений между двумя предметными областями.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T1* | Тип исходных данных. |
| *T2* | Тип целевых данных. |

См. определение в файле **IMapper.cs** строка **8**

### Методы

#### T2 ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 >.Map (T1 value)

Преобразует исходные данные в целевые данные.

##### Аргументы

|  |  |
| --- | --- |
| *value* | Исходное значение. |

##### Возвращает

Возвращает целевое значение.

#### T1 ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 >.MapBack (T2 value)

Преобразует целевые данные в исходные данные.

##### Аргументы

|  |  |
| --- | --- |
| *value* | Целевое значение. |

##### Возвращает

Возвращает исходное значение.

## Шаблон интерфейса ViewModel.Interfaces.ModelLearning.IModelTeacher< D >

Интерфейс для обучения модели на основе данных.

Граф наследования:ViewModel.Interfaces.ModelLearning.IModelTeacher< D >:

### Открытые члены

* Task< bool > **Train** (IEnumerable< D > data)

*Выполняет обучение модели на основе данных.*

### Подробное описание

Интерфейс для обучения модели на основе данных.

##### Параметры шаблона

|  |  |
| --- | --- |
| *D* | Тип данных. |

См. определение в файле **IModelTeacher.cs** строка **7**

### Методы

#### Task< bool > ViewModel.Interfaces.ModelLearning.IModelTeacher< D >.Train (IEnumerable< D > data)

Выполняет обучение модели на основе данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает задачу процесса обучения с результатом true , если обучение прошло успешно, иначе false .

Замещается в **ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >** (*стр.137*).

## Интерфейс ViewModel.Interfaces.AppStates.INotificationManager

Интерфейс менеджера уведомлений.

Граф наследования:ViewModel.Interfaces.AppStates.INotificationManager:

### Открытые члены

* void **SendNotification** (string description, string title)

*Отправляет уведомление.*

### Подробное описание

Интерфейс менеджера уведомлений.

См. определение в файле **INotificationManager.cs** строка **6**

### Методы

#### void ViewModel.Interfaces.AppStates.INotificationManager.SendNotification (string description, string title)

Отправляет уведомление.

##### Аргументы

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

Замещается в **View.Android.AndroidNotificationManager** (*стр.96*), **View.Desktop.Linux.LinuxNotificationManager** (*стр.345*), **View.Desktop.macOS.MacOsNotificationManager** (*стр.347*), **View.Desktop.Windows.WindowsNotificationManager** (*стр.536*), **View.iOS.IOsNotificationManager** (*стр.264*) и **ViewModel.Implementations.Mocks.MockNotificationManager** (*стр.369*).

## Класс MachineLearning.DataProcessors.InvalidValuesColumnProcessor

Класс обработчика столбцов для устранения некорректных значений.

Граф наследования:MachineLearning.DataProcessors.InvalidValuesColumnProcessor:

### Открытые члены

* DataProcessorResult< double > **Process** (IEnumerable< double?> data)
* DataProcessorResult< IEnumerable< double > > **Process** (IEnumerable< IEnumerable< double?> > data)

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IDataProcessor< T, R >

* DataProcessorResult< R > **Process** (IEnumerable< T > data)

*Преобразует данные.*

### Защищенные члены

* virtual bool **IsInvalidValue** (double? value)

*Проверяет, является ли значение не корректным.*

* virtual double **CalculateReplacementValue** (IEnumerable< double > column)

*Рассчитывает значение для замещения на основе столбца.*

### Защищенные данные

* double **\_replacementInvalidValue** = -1

*Значение для замены некорректных значений.*

### Подробное описание

Класс обработчика столбцов для устранения некорректных значений.

Реализует **IPrimaryPointDataProcessor**.

См. определение в файле **InvalidValuesColumnProcessor.cs** строка **13**

### Методы

#### virtual double MachineLearning.DataProcessors.InvalidValuesColumnProcessor.CalculateReplacementValue (IEnumerable< double > column)[protected], [virtual]

Рассчитывает значение для замещения на основе столбца.

##### Аргументы

|  |  |
| --- | --- |
| *column* | Столбец. |

##### Возвращает

Возвращает значение для замещения.

#### virtual bool MachineLearning.DataProcessors.InvalidValuesColumnProcessor.IsInvalidValue (double? value)[protected], [virtual]

Проверяет, является ли значение не корректным.

##### Аргументы

|  |  |
| --- | --- |
| *value* | Значение. |

##### Возвращает

Возвращает true , если значение не корректно, иначе false .

#### DataProcessorResult< double > MachineLearning.DataProcessors.InvalidValuesColumnProcessor.Process (IEnumerable< double?> data)

См. определение в файле **InvalidValuesColumnProcessor.cs** строка **21**

#### DataProcessorResult< IEnumerable< double > > MachineLearning.DataProcessors.InvalidValuesColumnProcessor.Process (IEnumerable< IEnumerable< double?> > data)

См. определение в файле **InvalidValuesColumnProcessor.cs** строка **28**

### Данные класса

#### double MachineLearning.DataProcessors.InvalidValuesColumnProcessor.\_replacementInvalidValue = -1[protected]

Значение для замены некорректных значений.

См. определение в файле **InvalidValuesColumnProcessor.cs** строка **18**

## Класс View.iOS.IOsNotificationManager

Класс менеджера уведомлений **iOS**.

Граф наследования:View.iOS.IOsNotificationManager:

### Открытые члены

* void **SendNotification** (string description, string title)

*Отправляет уведомление.*

##### *Аргументы*

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

### Подробное описание

Класс менеджера уведомлений **iOS**.

Реализует **INotificationManager**.

См. определение в файле **IOsNotificationManager.cs** строка **13**

### Методы

#### void View.iOS.IOsNotificationManager.SendNotification (string description, string title)

Отправляет уведомление.

##### Аргументы

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

Замещает **ViewModel.Interfaces.AppStates.INotificationManager** (*стр.261*).

См. определение в файле **IOsNotificationManager.cs** строка **21**

## Интерфейс MachineLearning.Interfaces.IPointDataProcessor

Интерфейс обработчика точечных данных.

Граф наследования:MachineLearning.Interfaces.IPointDataProcessor:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IDataProcessor< T, R >

* DataProcessorResult< R > **Process** (IEnumerable< T > data)

*Преобразует данные.*

### Подробное описание

Интерфейс обработчика точечных данных.

Наследует **IDataProcessor<IEnumerable<double>, IEnumerable<double>>**.

См. определение в файле **IPointDataProcessor.cs** строка **11**

## Интерфейс MachineLearning.Interfaces.IPointDistanceMetric

Интерфейс метрики расстояния для точечных данных.

Граф наследования:MachineLearning.Interfaces.IPointDistanceMetric:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IDistanceMetric< T >

* double **CalculateDistance** (T object1, T object2)

*Вычисляет расстояние между двумя объектами.*

### Подробное описание

Интерфейс метрики расстояния для точечных данных.

Наследует **IDistanceMetric<IEnumerable<double>>**.

См. определение в файле **IPointDistanceMetric.cs** строка **11**

## Интерфейс MachineLearning.Interfaces.IPrimaryPointDataProcessor

Интерфейс первичного обработчика точечных данных.

Граф наследования:MachineLearning.Interfaces.IPrimaryPointDataProcessor:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IDataProcessor< T, R >

* DataProcessorResult< R > **Process** (IEnumerable< T > data)

*Преобразует данные.*

### Подробное описание

Интерфейс первичного обработчика точечных данных.

Наследует **IDataProcessor<IEnumerable<Nullable<double>>, IEnumerable<double>>** и **IDataProcessor<Nullable<double>, double>**.

См. определение в файле **IPrimaryPointDataProcessor.cs** строка **12**

## Шаблон интерфейса ViewModel.Interfaces.DataManagers.Generals.IProxy< out T >

Интерфейс заместителя.

Граф наследования:ViewModel.Interfaces.DataManagers.Generals.IProxy< out T >:

### Свойства

* T **Target** [get]

*Возвращает заменяемого объекта.*

### Подробное описание

Интерфейс заместителя.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип заменяемого данных. |

См. определение в файле **IProxy.cs** строка **7**

### Полный список свойств

#### T ViewModel.Interfaces.DataManagers.Generals.IProxy< out T >.Target[get]

Возвращает заменяемого объекта.

Замещается в **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.485*), **ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy** (*стр.500*), **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.456*) и **ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T >** (*стр.242*).

См. определение в файле **IProxy.cs** строка **12**

## Интерфейс MachineLearning.Interfaces.IRegressionEvaluator

Интерфейс оценки модели обучения классификации.

Граф наследования:MachineLearning.Interfaces.IRegressionEvaluator:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >

* Task< **ScoreMetricCategory** > **Evaluate** (IEnumerable< T > data, IEnumerable< R > targets)

*Оценивает модель обучения.*

#### Свойства унаследованные от MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >

* ISupervisedLearningModel< T, R > **Model** [get, set]

*Возвращает и задаёт модель обучения с учителем.*

* ISupervisedScoreMetric< R > **ScoreMetric** [get, set]

*Возвращает и задаёт метрику оценки обучения с учителем.*

### Подробное описание

Интерфейс оценки модели обучения классификации.

Наследует **ISupervisedLearningEvaluator<IEnumerable<double>, double>**.

См. определение в файле **IRegressionEvaluator.cs** строка **11**

## Интерфейс MachineLearning.Interfaces.IRegressionModel

Интерфейс модели обучения регресии.

Граф наследования:MachineLearning.Interfaces.IRegressionModel:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedLearningModel< T, R >

* Task **Train** (IEnumerable< T > data, IEnumerable< R > targets)

*Обучает модель на основе данных и целевых значений.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ILearningModel< T, R >

* R **Predict** (T data)

*Предсказывает значение на основе данных.*

### Подробное описание

Интерфейс модели обучения регресии.

Наследует **ISupervisedLearningModel<IEnumerable<double>, double>**.

См. определение в файле **IRegressionModel.cs** строка **11**

## Интерфейс MachineLearning.Interfaces.IRegressionScoreMetric

Интерфейс метрики оценки для модели обучения регрессии.

Граф наследования:MachineLearning.Interfaces.IRegressionScoreMetric:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T >

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< T > predicted)

*Вычисляет оценку на основе фактических и предсказанных данных.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IScoreMetric

* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

### Подробное описание

Интерфейс метрики оценки для модели обучения регрессии.

Наследует **ISupervisedScoreMetric<double>**.

См. определение в файле **IRegressionScoreMetric.cs** строка **11**

## Интерфейс ViewModel.Interfaces.AppStates.IResourceService

Интерфейс сервиса ресурсов.

Граф наследования:ViewModel.Interfaces.AppStates.IResourceService:

### Открытые члены

* object? **GetResource** (object key)

*Возвращает ресурс по ключу.*

### Подробное описание

Интерфейс сервиса ресурсов.

См. определение в файле **IResourceService.cs** строка **6**

### Методы

#### object? ViewModel.Interfaces.AppStates.IResourceService.GetResource (object key)

Возвращает ресурс по ключу.

##### Аргументы

|  |  |
| --- | --- |
| *key* | Ключ. |

##### Возвращает

Возвращает ресурс.

Замещается в **View.Implementations.AvaloniaResourceService** (*стр.112*) и **ViewModel.Implementations.Mocks.MockResourceService** (*стр.371*).

## Интерфейс MachineLearning.Interfaces.IScaler

Интерфейс масштабирования данных.

Граф наследования:MachineLearning.Interfaces.IScaler:

### Открытые члены

* IEnumerable< double > **FitTransform** (IEnumerable< double > data)

*Адаптирует преобразования под заданные данные.*

* double **Transform** (double value)

*Преобразует значение.*

### Подробное описание

Интерфейс масштабирования данных.

См. определение в файле **IScaler.cs** строка **6**

### Методы

#### IEnumerable< double > MachineLearning.Interfaces.IScaler.FitTransform (IEnumerable< double > data)

Адаптирует преобразования под заданные данные.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает коллекцию нормализованных данных.

Замещается в **MachineLearning.Scalers.MinMaxScaler** (*стр.366*).

#### double MachineLearning.Interfaces.IScaler.Transform (double value)

Преобразует значение.

##### Аргументы

|  |  |
| --- | --- |
| *value* | Значение. |

##### Возвращает

Возвращает нормализованное значение.

Замещается в **MachineLearning.Scalers.MinMaxScaler** (*стр.366*).

## Интерфейс MachineLearning.Interfaces.Generals.IScoreMetric

Интерфейс метрики оценки.

Граф наследования:MachineLearning.Interfaces.Generals.IScoreMetric:

### Открытые члены

* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

### Подробное описание

Интерфейс метрики оценки.

См. определение в файле **IScoreMetric.cs** строка **6**

### Методы

#### ScoreMetricCategory MachineLearning.Interfaces.Generals.IScoreMetric.GetScoreCategory (double score)

Определяет категорию оценки.

##### Аргументы

|  |  |
| --- | --- |
| *score* | Оценка. |

##### Возвращает

Возвращает категорию оценки.

Замещается в **MachineLearning.ScoreMetrics.AccuracyScoreMetric** (*стр.86*), **MachineLearning.ScoreMetrics.AdjustedRandIndexScoreMetric** (*стр.95*), **MachineLearning.ScoreMetrics.F1ScoreMetric** (*стр.220*), **MachineLearning.ScoreMetrics.RSquaredScoreMetric** (*стр.408*), **MachineLearning.ScoreMetrics.SilhouetteScoreMetric** (*стр.414*) и **MachineLearning.ScoreMetrics.SmapeScoreMetric** (*стр.416*).

## Интерфейс ViewModel.Interfaces.AppStates.ISerializer

Интерфейс сериализатора.

Граф наследования:ViewModel.Interfaces.AppStates.ISerializer:

### Открытые члены

* byte[] **Serialize** (object value)

*Сериализует объект.*

* T? **Deserialize< T >** (byte[] data)

*Десериализует объект.*

### Подробное описание

Интерфейс сериализатора.

См. определение в файле **ISerializer.cs** строка **6**

### Методы

#### T? ViewModel.Interfaces.AppStates.ISerializer.Deserialize< T > (byte[] data)

Десериализует объект.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает десериализованный объект.

Замещается в **ViewModel.Implementations.AppStates.JsonSerializer** (*стр.335*).

#### byte[] ViewModel.Interfaces.AppStates.ISerializer.Serialize (object value)

Сериализует объект.

##### Аргументы

|  |  |
| --- | --- |
| *value* | Объект. |

##### Возвращает

Возвращает массив байтов, представляющий сериализованные данные.

Замещается в **ViewModel.Implementations.AppStates.JsonSerializer** (*стр.336*).

## Интерфейс ViewModel.Interfaces.AppStates.Sessions.ISession

Интерфейс сессии для хранения и изменения данных.

Граф наследования:ViewModel.Interfaces.AppStates.Sessions.ISession:

### Открытые члены

* void **AddTasks** (IEnumerable< **ITask** > tasks, **ITaskComposite**? parentTask)

*Добавляет задачи.*

* void **EditTask** (**ITask** task)

*Изменяет задачу.*

* void **RemoveTasks** (IEnumerable< **ITask** > tasks)

*Удаляет задачи.*

* void **MoveTasks** (IEnumerable< **ITask** > tasks, **ITaskComposite**? parentTask)

*Перемещает задачи.*

* void **AddTimeInterval** (**ITimeIntervalElement** timeIntervalElement, **ITaskElement** taskElement)

*Добавляет временной интервал.*

* void **EditTimeInterval** (**ITimeIntervalElement** timeIntervalElement)

*Изменяет временной интервал.*

* void **RemoveTimeInterval** (**ITimeIntervalElement** timeIntervalElement, **ITaskElement** taskElement)

*Удаляет временной интервал.*

#### Открытые члены унаследованные от ViewModel.Interfaces.AppStates.IStorageService

* Task **Save** ()

*Сохраняет данные в хранилище.*

* Task **Load** ()

*Загружает данные из хранилища.*

### Свойства

* string **SavePath** [get, set]

*Возвращает и задаёт путь сохранения.*

* IEnumerable< **ITask** > **Tasks** [get]

*Возвращает задачи.*

### События

* EventHandler< **ItemsUpdatedEventArgs** > **ItemsUpdated**

*Событие, которое возникает при обновлении данных.*

### Подробное описание

Интерфейс сессии для хранения и изменения данных.

Наследует **IStorageService**.

См. определение в файле **ISession.cs** строка **11**

### Методы

#### void ViewModel.Interfaces.AppStates.Sessions.ISession.AddTasks (IEnumerable< ITask > tasks, ITaskComposite? parentTask)

Добавляет задачи.

##### Аргументы

|  |  |
| --- | --- |
| *tasks* | Задачи. |
| *parentTask* | Родительская задача. |

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.187*).

#### void ViewModel.Interfaces.AppStates.Sessions.ISession.AddTimeInterval (ITimeIntervalElement timeIntervalElement, ITaskElement taskElement)

Добавляет временной интервал.

##### Аргументы

|  |  |
| --- | --- |
| *timeIntervalElement* | Элементарный временной интервал. |
| *taskElement* | Элементарная задача. |

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.187*).

#### void ViewModel.Interfaces.AppStates.Sessions.ISession.EditTask (ITask task)

Изменяет задачу.

##### Аргументы

|  |  |
| --- | --- |
| *task* | Задача. |

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.187*).

#### void ViewModel.Interfaces.AppStates.Sessions.ISession.EditTimeInterval (ITimeIntervalElement timeIntervalElement)

Изменяет временной интервал.

##### Аргументы

|  |  |
| --- | --- |
| *timeIntervalElement* | Элементарный временной интервал. |

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.188*).

#### void ViewModel.Interfaces.AppStates.Sessions.ISession.MoveTasks (IEnumerable< ITask > tasks, ITaskComposite? parentTask)

Перемещает задачи.

##### Аргументы

|  |  |
| --- | --- |
| *tasks* | Задачи. |
| *parentTask* | Родительская задача. |

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.188*).

#### void ViewModel.Interfaces.AppStates.Sessions.ISession.RemoveTasks (IEnumerable< ITask > tasks)

Удаляет задачи.

##### Аргументы

|  |  |
| --- | --- |
| *tasks* | Задачи. |

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.188*).

#### void ViewModel.Interfaces.AppStates.Sessions.ISession.RemoveTimeInterval (ITimeIntervalElement timeIntervalElement, ITaskElement taskElement)

Удаляет временной интервал.

##### Аргументы

|  |  |
| --- | --- |
| *timeIntervalElement* | Элементарный временной интервал. |
| *taskElement* | Элементарная задача. |

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.188*).

### Полный список свойств

#### string ViewModel.Interfaces.AppStates.Sessions.ISession.SavePath[get], [set]

Возвращает и задаёт путь сохранения.

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.189*).

См. определение в файле **ISession.cs** строка **16**

#### IEnumerable<ITask> ViewModel.Interfaces.AppStates.Sessions.ISession.Tasks[get]

Возвращает задачи.

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.189*).

См. определение в файле **ISession.cs** строка **21**

### Cобытия

#### EventHandler<ItemsUpdatedEventArgs> ViewModel.Interfaces.AppStates.Sessions.ISession.ItemsUpdated

Событие, которое возникает при обновлении данных.

См. определение в файле **ISession.cs** строка **26**

## Интерфейс ViewModel.Interfaces.AppStates.Settings.ISettings

Интерфейс настроек.

Граф наследования:ViewModel.Interfaces.AppStates.Settings.ISettings:

### Свойства

* object **Configuration** [get]

*Возвращает конфигурацию.*

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.Interfaces.AppStates.IStorageService

* Task **Save** ()

*Сохраняет данные в хранилище.*

* Task **Load** ()

*Загружает данные из хранилища.*

### Подробное описание

Интерфейс настроек.

Наследует **IStorageService**.

См. определение в файле **ISettings.cs** строка **9**

### Полный список свойств

#### object ViewModel.Interfaces.AppStates.Settings.ISettings.Configuration[get]

Возвращает конфигурацию.

Замещается в **ViewModel.Implementations.AppStates.Settings.AppSettings** (*стр.109*).

См. определение в файле **ISettings.cs** строка **14**

## Интерфейс ViewModel.Interfaces.AppStates.IStorageService

Интерфейс для сервиса хранилища.

Граф наследования:ViewModel.Interfaces.AppStates.IStorageService:

### Открытые члены

* Task **Save** ()

*Сохраняет данные в хранилище.*

* Task **Load** ()

*Загружает данные из хранилища.*

### Подробное описание

Интерфейс для сервиса хранилища.

См. определение в файле **IStorageService.cs** строка **6**

### Методы

#### Task ViewModel.Interfaces.AppStates.IStorageService.Load ()

Загружает данные из хранилища.

##### Возвращает

Возвращает задачц процесса загрузки данных.

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.188*) и **ViewModel.Implementations.AppStates.Settings.AppSettings** (*стр.108*).

#### Task ViewModel.Interfaces.AppStates.IStorageService.Save ()

Сохраняет данные в хранилище.

##### Возвращает

Возвращает задачц процесса сохранения данных.

Замещается в **ViewModel.Implementations.AppStates.Sessions.DbSession** (*стр.189*) и **ViewModel.Implementations.AppStates.Settings.AppSettings** (*стр.109*).

## Шаблон интерфейса ViewModel.Interfaces.ModelLearning.ISupervisedLearningController< T, R, D, DT, DR >

Интерфейс контроллера обучения модели обучения с учителем.

Граф наследования:ViewModel.Interfaces.ModelLearning.ISupervisedLearningController< T, R, D, DT, DR >:

### Свойства

* ISupervisedLearningModel< T, R > **LearningModel** [get, set]

*Возвращает и задаёт модель обучение с учителем.*

* ISupervisedLearningEvaluator< T, R > **LearningEvaluator** [get, set]

*Возвращает и задаёт оцениватель модели обучения с учителем.*

#### Свойства унаследованные от ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR >

* bool **IsValidModel** [get]

*Возвращает логическое значение, указывающее на достоверность модели обучения.*

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR >

* DR **Predict** (DT data)

*Выполняет предсказание на основе данных.*

#### Открытые члены унаследованные от ViewModel.Interfaces.ModelLearning.IModelTeacher< D >

* Task< bool > **Train** (IEnumerable< D > data)

*Выполняет обучение модели на основе данных.*

### Подробное описание

Интерфейс контроллера обучения модели обучения с учителем.

Наследует **ILearningController<D, DT, DR>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |
| *D* | Тип данных. |
| *DT* | Тип входных данных. |
| *DR* | Тип выходных данных. |

См. определение в файле **ISupervisedLearningModel.cs** строка **16**

### Полный список свойств

#### ISupervisedLearningEvaluator<T, R> ViewModel.Interfaces.ModelLearning.ISupervisedLearningController< T, R, D, DT, DR >.LearningEvaluator[get], [set]

Возвращает и задаёт оцениватель модели обучения с учителем.

См. определение в файле **ISupervisedLearningModel.cs** строка **27**

#### ISupervisedLearningModel<T, R> ViewModel.Interfaces.ModelLearning.ISupervisedLearningController< T, R, D, DT, DR >.LearningModel[get], [set]

Возвращает и задаёт модель обучение с учителем.

См. определение в файле **ISupervisedLearningModel.cs** строка **22**

## Шаблон интерфейса MachineLearning.Interfaces.Generals.ISupervisedLearningConverter< T, R, D, DT, DR >

Интерфейс конвертора данных в данные для предсказания с учителем и наоборот.

Граф наследования:MachineLearning.Interfaces.Generals.ISupervisedLearningConverter< T, R, D, DT, DR >:

### Открытые члены

* LearningModelData< T, R > **FitConvertData** (IEnumerable< D > data)

*Адаптирует конвертацию данных под данные для предсказания.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ILearningConverter< T, R, DT, DR >

* T **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

### Подробное описание

Интерфейс конвертора данных в данные для предсказания с учителем и наоборот.

Наследует **ILearningConverter<T, R, DT, DR>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |
| *D* | Тип данных. |
| *DT* | Тип входных данных. |
| *DR* | Тип выходных данных. |

См. определение в файле **ISupervisedLearningConverter.cs** строка **14**

### Методы

#### LearningModelData< T, R > MachineLearning.Interfaces.Generals.ISupervisedLearningConverter< T, R, D, DT, DR >.FitConvertData (IEnumerable< D > data)

Адаптирует конвертацию данных под данные для предсказания.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает конвертированные данные для предсказания.

Замещается в **MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >** (*стр.141*).

## Шаблон интерфейса MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >

Интерфейс оценки модели обучения с учителем.

Граф наследования:MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >:

### Открытые члены

* Task< **ScoreMetricCategory** > **Evaluate** (IEnumerable< T > data, IEnumerable< R > targets)

*Оценивает модель обучения.*

### Свойства

* ISupervisedLearningModel< T, R > **Model** [get, set]

*Возвращает и задаёт модель обучения с учителем.*

* ISupervisedScoreMetric< R > **ScoreMetric** [get, set]

*Возвращает и задаёт метрику оценки обучения с учителем.*

### Подробное описание

Интерфейс оценки модели обучения с учителем.

Наследует **ILearningEvaluator**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |

См. определение в файле **ISupervisedLearningEvaluator.cs** строка **11**

### Методы

#### Task< ScoreMetricCategory > MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >.Evaluate (IEnumerable< T > data, IEnumerable< R > targets)

Оценивает модель обучения.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |
| *targets* | Целевые значения. |

##### Возвращает

Возвращает категорию модели обучения.

Замещается в **MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >** (*стр.132*).

### Полный список свойств

#### ISupervisedLearningModel<T, R> MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >.Model[get], [set]

Возвращает и задаёт модель обучения с учителем.

Замещается в **MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >** (*стр.133*).

См. определение в файле **ISupervisedLearningEvaluator.cs** строка **16**

#### ISupervisedScoreMetric<R> MachineLearning.Interfaces.Generals.ISupervisedLearningEvaluator< T, R >.ScoreMetric[get], [set]

Возвращает и задаёт метрику оценки обучения с учителем.

Замещается в **MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >** (*стр.133*).

См. определение в файле **ISupervisedLearningEvaluator.cs** строка **21**

## Шаблон интерфейса MachineLearning.Interfaces.Generals.ISupervisedLearningModel< T, R >

Интерфейс модели обучения c учителем.

Граф наследования:MachineLearning.Interfaces.Generals.ISupervisedLearningModel< T, R >:

### Открытые члены

* Task **Train** (IEnumerable< T > data, IEnumerable< R > targets)

*Обучает модель на основе данных и целевых значений.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ILearningModel< T, R >

* R **Predict** (T data)

*Предсказывает значение на основе данных.*

### Подробное описание

Интерфейс модели обучения c учителем.

Наследует **ILearningModel<T, R>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |

См. определение в файле **ISupervisedLearningModel.cs** строка **11**

### Методы

#### Task MachineLearning.Interfaces.Generals.ISupervisedLearningModel< T, R >.Train (IEnumerable< T > data, IEnumerable< R > targets)

Обучает модель на основе данных и целевых значений.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |
| *targets* | Целевые значения. |

##### Возвращает

Возвращает задачу процесса обучения.

## Шаблон интерфейса MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T >

Интерфейс метрики оценки для модели обучения с учителем.

Граф наследования:MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T >:

### Открытые члены

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< T > predicted)

*Вычисляет оценку на основе фактических и предсказанных данных.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IScoreMetric

* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

### Подробное описание

Интерфейс метрики оценки для модели обучения с учителем.

Наследует **IScoreMetric**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип данных для оценки. |

См. определение в файле **ISupervisedScoreMetric.cs** строка **10**

### Методы

#### double MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T >.CalculateScore (IEnumerable< T > actual, IEnumerable< T > predicted)

Вычисляет оценку на основе фактических и предсказанных данных.

##### Аргументы

|  |  |
| --- | --- |
| *actual* | Коллекция фактических данных. |
| *predicted* | Коллекция предсказанных значений. |

##### Возвращает

Возвращает значение оценки.

## Интерфейс Model.Interfaces.ITask

Интерфейс задачи.

Граф наследования:Model.Interfaces.ITask:

### Свойства

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

* int **Difficult** [get]

*Возращает сложность.*

* int **Priority** [get]

*Возращает приоритет.*

* **TaskStatus** **Status** [get]

*Возращает статус.*

* DateTime? **Deadline** [get]

*Возращает срок.*

* double **Progress** [get]

*Возращает прогресс.*

* TimeSpan **PlannedTime** [get]

*Возращает запланированное время.*

* TimeSpan **SpentTime** [get]

*Возращает потраченное время.*

### Подробное описание

Интерфейс задачи.

См. определение в файле **ITask.cs** строка **6**

### Полный список свойств

#### DateTime? Model.Interfaces.ITask.Deadline[get]

Возращает срок.

Замещается в **Model.Interfaces.ITaskElement** (*стр.302*), **Model.Tasks.TaskComposite** (*стр.432*), **Model.Tasks.TaskElement** (*стр.448*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.470*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.483*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.453*).

См. определение в файле **ITask.cs** строка **36**

#### int Model.Interfaces.ITask.Difficult[get]

Возращает сложность.

Замещается в **Model.Interfaces.ITaskElement** (*стр.302*), **Model.Tasks.TaskComposite** (*стр.432*), **Model.Tasks.TaskElement** (*стр.448*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.471*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.483*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.453*).

См. определение в файле **ITask.cs** строка **21**

#### object? Model.Interfaces.ITask.Metadata[get], [set]

Возращает и задаёт метаданные.

Замещается в **Model.Tasks.TaskComposite** (*стр.432*), **Model.Tasks.TaskElement** (*стр.448*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.471*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.483*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.454*).

См. определение в файле **ITask.cs** строка **16**

#### ITaskComposite? Model.Interfaces.ITask.ParentTask[get], [set]

Возращает и задаёт родительскую задачу.

Замещается в **Model.Tasks.TaskComposite** (*стр.432*), **Model.Tasks.TaskElement** (*стр.448*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.471*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.484*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.454*).

См. определение в файле **ITask.cs** строка **11**

#### TimeSpan Model.Interfaces.ITask.PlannedTime[get]

Возращает запланированное время.

Замещается в **Model.Interfaces.ITaskElement** (*стр.303*), **Model.Tasks.TaskComposite** (*стр.432*), **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.471*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.484*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.455*).

См. определение в файле **ITask.cs** строка **46**

#### int Model.Interfaces.ITask.Priority[get]

Возращает приоритет.

Замещается в **Model.Interfaces.ITaskElement** (*стр.303*), **Model.Tasks.TaskComposite** (*стр.433*), **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.484*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.455*).

См. определение в файле **ITask.cs** строка **26**

#### double Model.Interfaces.ITask.Progress[get]

Возращает прогресс.

Замещается в **Model.Interfaces.ITaskElement** (*стр.303*), **Model.Tasks.TaskComposite** (*стр.433*), **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.484*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.456*).

См. определение в файле **ITask.cs** строка **41**

#### TimeSpan Model.Interfaces.ITask.SpentTime[get]

Возращает потраченное время.

Замещается в **Model.Interfaces.ITaskElement** (*стр.303*), **Model.Tasks.TaskComposite** (*стр.433*), **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.484*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.456*).

См. определение в файле **ITask.cs** строка **51**

#### TaskStatus Model.Interfaces.ITask.Status[get]

Возращает статус.

Замещается в **Model.Interfaces.ITaskElement** (*стр.303*), **Model.Tasks.TaskComposite** (*стр.433*), **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*), **ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy** (*стр.484*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.456*).

См. определение в файле **ITask.cs** строка **31**

## Интерфейс Model.Interfaces.ITaskComposite

Интерфейс составной задачи.

Граф наследования:Model.Interfaces.ITaskComposite:

### Дополнительные унаследованные члены

#### Свойства унаследованные от Model.Interfaces.ITask

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

* int **Difficult** [get]

*Возращает сложность.*

* int **Priority** [get]

*Возращает приоритет.*

* **TaskStatus** **Status** [get]

*Возращает статус.*

* DateTime? **Deadline** [get]

*Возращает срок.*

* double **Progress** [get]

*Возращает прогресс.*

* TimeSpan **PlannedTime** [get]

*Возращает запланированное время.*

* TimeSpan **SpentTime** [get]

*Возращает потраченное время.*

### Подробное описание

Интерфейс составной задачи.

Наследует **ITask**, IList<ITask> и IList.

См. определение в файле **ITaskComposite.cs** строка **11**

## Интерфейс Model.Interfaces.ITaskElement

Интерфейс элементарной задачи.

Граф наследования:Model.Interfaces.ITaskElement:

### Свойства

* new int **Difficult** [get, set]

*Возвращает и задаёт сложность.*

* new int **Priority** [get, set]

*Возвращает и задаёт приоритет.*

* new **TaskStatus** **Status** [get, set]

*Возвращает и задаёт статус.*

* new? DateTime **Deadline** [get, set]

*Возвращает и задаёт срок.*

* new double **Progress** [get, set]

*Возвращает и задаёт прогресс.*

* new TimeSpan **PlannedTime** [get, set]

*Возвращает и задаёт запланированное время.*

* new TimeSpan **SpentTime** [get, set]

*Возвращает и задаёт потраченное время.*

* double **PlannedReal** [get, set]

*Возвращает и задаёт запланированный реальный показатель.*

* double **ExecutedReal** [get, set]

*Возвращает и задаёт выполненный реальный показатель.*

* **ITimeIntervalList** **TimeIntervals** [get]

*Возвращает список временных интервалов.*

#### Свойства унаследованные от Model.Interfaces.ITask

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

### Подробное описание

Интерфейс элементарной задачи.

Наследует **ITask**.

См. определение в файле **ITaskElement.cs** строка **9**

### Полный список свойств

#### new? DateTime Model.Interfaces.ITaskElement.Deadline[get], [set]

Возвращает и задаёт срок.

Замещает **Model.Interfaces.ITask** (*стр.296*).

Замещается в **Model.Tasks.TaskElement** (*стр.448*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.470*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.453*).

См. определение в файле **ITaskElement.cs** строка **29**

#### new int Model.Interfaces.ITaskElement.Difficult[get], [set]

Возвращает и задаёт сложность.

Замещает **Model.Interfaces.ITask** (*стр.296*).

Замещается в **Model.Tasks.TaskElement** (*стр.448*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.471*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.453*).

См. определение в файле **ITaskElement.cs** строка **14**

#### double Model.Interfaces.ITaskElement.ExecutedReal[get], [set]

Возвращает и задаёт выполненный реальный показатель.

Замещается в **Model.Tasks.TaskElement** (*стр.448*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.471*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.454*).

См. определение в файле **ITaskElement.cs** строка **54**

#### double Model.Interfaces.ITaskElement.PlannedReal[get], [set]

Возвращает и задаёт запланированный реальный показатель.

Замещается в **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.471*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.455*).

См. определение в файле **ITaskElement.cs** строка **49**

#### new TimeSpan Model.Interfaces.ITaskElement.PlannedTime[get], [set]

Возвращает и задаёт запланированное время.

Замещает **Model.Interfaces.ITask** (*стр.297*).

Замещается в **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.471*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.455*).

См. определение в файле **ITaskElement.cs** строка **39**

#### new int Model.Interfaces.ITaskElement.Priority[get], [set]

Возвращает и задаёт приоритет.

Замещает **Model.Interfaces.ITask** (*стр.297*).

Замещается в **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.455*).

См. определение в файле **ITaskElement.cs** строка **19**

#### new double Model.Interfaces.ITaskElement.Progress[get], [set]

Возвращает и задаёт прогресс.

Замещает **Model.Interfaces.ITask** (*стр.297*).

Замещается в **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.456*).

См. определение в файле **ITaskElement.cs** строка **34**

#### new TimeSpan Model.Interfaces.ITaskElement.SpentTime[get], [set]

Возвращает и задаёт потраченное время.

Замещает **Model.Interfaces.ITask** (*стр.297*).

Замещается в **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.456*).

См. определение в файле **ITaskElement.cs** строка **44**

#### new TaskStatus Model.Interfaces.ITaskElement.Status[get], [set]

Возвращает и задаёт статус.

Замещает **Model.Interfaces.ITask** (*стр.297*).

Замещается в **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.456*).

См. определение в файле **ITaskElement.cs** строка **24**

#### ITimeIntervalList Model.Interfaces.ITaskElement.TimeIntervals[get]

Возвращает список временных интервалов.

Замещается в **Model.Tasks.TaskElement** (*стр.449*), **ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy** (*стр.472*) и **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.456*).

См. определение в файле **ITaskElement.cs** строка **59**

## Интерфейс ViewModel.Interfaces.DataManagers.ITaskElementProxy

Интерфейс заместителя элементарной задачи.

Граф наследования:ViewModel.Interfaces.DataManagers.ITaskElementProxy:

### Свойства

* DateTime? **PredictedDeadline** [get]

*Возвращает предсказанный срок.*

* TimeSpan **PredictedPlannedTime** [get]

*Возвращает предсказанное запланированное время.*

* double **PredictedPlannedReal** [get]

*Возвращает предсказанный реальный запланированный показатель.*

* bool **IsValidPredictedDeadline** [get]

*Возвращает логическое значение, указывающее на корректность предсказания срок.*

* bool **IsValidPredictedPlannedTime** [get]

*Возвращает логическое значение, указывающее на корректность предсказания запланированного времени.*

* bool **IsValidPredictedPlannedReal** [get]

*Возвращает логическое значение, указывающее на корректность предсказания реального запланированного показателя.*

#### Свойства унаследованные от ViewModel.Interfaces.DataManagers.Generals.IProxy< out T >

* T **Target** [get]

*Возвращает заменяемого объекта.*

#### Свойства унаследованные от Model.Interfaces.ITaskElement

* new int **Difficult** [get, set]

*Возвращает и задаёт сложность.*

* new int **Priority** [get, set]

*Возвращает и задаёт приоритет.*

* new **TaskStatus** **Status** [get, set]

*Возвращает и задаёт статус.*

* new? DateTime **Deadline** [get, set]

*Возвращает и задаёт срок.*

* new double **Progress** [get, set]

*Возвращает и задаёт прогресс.*

* new TimeSpan **PlannedTime** [get, set]

*Возвращает и задаёт запланированное время.*

* new TimeSpan **SpentTime** [get, set]

*Возвращает и задаёт потраченное время.*

* double **PlannedReal** [get, set]

*Возвращает и задаёт запланированный реальный показатель.*

* double **ExecutedReal** [get, set]

*Возвращает и задаёт выполненный реальный показатель.*

* **ITimeIntervalList** **TimeIntervals** [get]

*Возвращает список временных интервалов.*

#### Свойства унаследованные от Model.Interfaces.ITask

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

### Подробное описание

Интерфейс заместителя элементарной задачи.

Наследует **IProxy<ITaskElement>** и **ITaskElement**.

См. определение в файле **ITaskElementProxy.cs** строка **13**

### Полный список свойств

#### bool ViewModel.Interfaces.DataManagers.ITaskElementProxy.IsValidPredictedDeadline[get]

Возвращает логическое значение, указывающее на корректность предсказания срок.

Замещается в **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.454*).

См. определение в файле **ITaskElementProxy.cs** строка **33**

#### bool ViewModel.Interfaces.DataManagers.ITaskElementProxy.IsValidPredictedPlannedReal[get]

Возвращает логическое значение, указывающее на корректность предсказания реального запланированного показателя.

Замещается в **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.454*).

См. определение в файле **ITaskElementProxy.cs** строка **45**

#### bool ViewModel.Interfaces.DataManagers.ITaskElementProxy.IsValidPredictedPlannedTime[get]

Возвращает логическое значение, указывающее на корректность предсказания запланированного времени.

Замещается в **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.454*).

См. определение в файле **ITaskElementProxy.cs** строка **39**

#### DateTime? ViewModel.Interfaces.DataManagers.ITaskElementProxy.PredictedDeadline[get]

Возвращает предсказанный срок.

Замещается в **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.455*).

См. определение в файле **ITaskElementProxy.cs** строка **18**

#### double ViewModel.Interfaces.DataManagers.ITaskElementProxy.PredictedPlannedReal[get]

Возвращает предсказанный реальный запланированный показатель.

Замещается в **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.455*).

См. определение в файле **ITaskElementProxy.cs** строка **28**

#### TimeSpan ViewModel.Interfaces.DataManagers.ITaskElementProxy.PredictedPlannedTime[get]

Возвращает предсказанное запланированное время.

Замещается в **ViewModel.Implementations.DataManagers.TaskElementCreatorProxy** (*стр.455*).

См. определение в файле **ITaskElementProxy.cs** строка **23**

## Интерфейс ViewModel.Interfaces.DataManagers.ITaskElementsEditorProxy

Интерфейс заместителя элементарной задачи для редактирования.

Граф наследования:ViewModel.Interfaces.DataManagers.ITaskElementsEditorProxy:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.Interfaces.DataManagers.Generals.IEditorService

* void **ApplyChanges** ()

*Применения изменений.*

#### Свойства унаследованные от ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T >

* new T **Target** [get, set]

*Возвращает и задаёт заменяемого объекта.*

#### Свойства унаследованные от Model.Interfaces.ITaskElement

* new int **Difficult** [get, set]

*Возвращает и задаёт сложность.*

* new int **Priority** [get, set]

*Возвращает и задаёт приоритет.*

* new **TaskStatus** **Status** [get, set]

*Возвращает и задаёт статус.*

* new? DateTime **Deadline** [get, set]

*Возвращает и задаёт срок.*

* new double **Progress** [get, set]

*Возвращает и задаёт прогресс.*

* new TimeSpan **PlannedTime** [get, set]

*Возвращает и задаёт запланированное время.*

* new TimeSpan **SpentTime** [get, set]

*Возвращает и задаёт потраченное время.*

* double **PlannedReal** [get, set]

*Возвращает и задаёт запланированный реальный показатель.*

* double **ExecutedReal** [get, set]

*Возвращает и задаёт выполненный реальный показатель.*

* **ITimeIntervalList** **TimeIntervals** [get]

*Возвращает список временных интервалов.*

#### Свойства унаследованные от Model.Interfaces.ITask

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

### Подробное описание

Интерфейс заместителя элементарной задачи для редактирования.

Наследует **IEditorProxy<ITaskElement>** и **ITaskElement**.

См. определение в файле **ITaskElementsEditorProxy.cs** строка **13**

## Интерфейс ViewModel.Interfaces.DataManagers.ITasksEditorProxy

Интерфейс заместителя задачи для редактирования.

Граф наследования:ViewModel.Interfaces.DataManagers.ITasksEditorProxy:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.Interfaces.DataManagers.Generals.IEditorService

* void **ApplyChanges** ()

*Применения изменений.*

#### Свойства унаследованные от ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T >

* new T **Target** [get, set]

*Возвращает и задаёт заменяемого объекта.*

#### Свойства унаследованные от Model.Interfaces.ITask

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

* int **Difficult** [get]

*Возращает сложность.*

* int **Priority** [get]

*Возращает приоритет.*

* **TaskStatus** **Status** [get]

*Возращает статус.*

* DateTime? **Deadline** [get]

*Возращает срок.*

* double **Progress** [get]

*Возращает прогресс.*

* TimeSpan **PlannedTime** [get]

*Возращает запланированное время.*

* TimeSpan **SpentTime** [get]

*Возращает потраченное время.*

### Подробное описание

Интерфейс заместителя задачи для редактирования.

Наследует **IEditorProxy<ITask>** и **ITask**.

См. определение в файле **ITasksEditorProxy.cs** строка **13**

## Класс ViewModel.ViewModels.Modals.ItemsTasksViewModelArgs

Класс аргументов диалога элементов списка.

Граф наследования:ViewModel.ViewModels.Modals.ItemsTasksViewModelArgs:

### Открытые члены

* **ItemsTasksViewModelArgs** (IList< **ITask** > items, IEnumerable< **ITask** > list, IEnumerable< **ITask** > mainList)

*Создаёт экземпляр класса* ***ItemsTasksViewModelArgs****.*

#### Открытые члены унаследованные от ViewModel.ViewModels.Modals.TasksViewModelArgs

* **TasksViewModelArgs** (IEnumerable< **ITask** > list, IEnumerable< **ITask** > mainList)

*Создаёт экземпляр класса* ***TasksViewModelArgs****.*

### Свойства

* IList< **ITask** > **Items** [get]

*Возвращает элементы.*

#### Свойства унаследованные от ViewModel.ViewModels.Modals.TasksViewModelArgs

* IEnumerable< **ITask** > **List** [get]

*Возвращает список.*

* IEnumerable< **ITask** > **MainList** [get]

*Возвращает основной список.*

### Подробное описание

Класс аргументов диалога элементов списка.

Наследует **TasksViewModelArgs**.

См. определение в файле **ItemsTasksViewModelArgs.cs** строка **11**

### Конструктор(ы)

#### ViewModel.ViewModels.Modals.ItemsTasksViewModelArgs.ItemsTasksViewModelArgs (IList< ITask > items, IEnumerable< ITask > list, IEnumerable< ITask > mainList)

Создаёт экземпляр класса **ItemsTasksViewModelArgs**.

##### Аргументы

|  |  |
| --- | --- |
| *items* | Элеметоы. |
| *list* | Список. |
| *mainList* | Основной список. |

См. определение в файле **ItemsTasksViewModelArgs.cs** строка **24**

### Полный список свойств

#### IList<ITask> ViewModel.ViewModels.Modals.ItemsTasksViewModelArgs.Items[get]

Возвращает элементы.

См. определение в файле **ItemsTasksViewModelArgs.cs** строка **16**

## Класс ViewModel.Interfaces.AppStates.Sessions.ItemsUpdatedEventArgs

Класс аргумента события при обновлении данных.

### Открытые члены

* **ItemsUpdatedEventArgs** (**UpdateItemsState** state, IEnumerable< object > items, Type itemsType)

*Создаёт экземпляр класса* ***ItemsUpdatedEventArgs****.*

### Свойства

* **UpdateItemsState** **State** [get]

*Возвращает состояние.*

* IEnumerable< object > **Items** [get]

*Возвращает объекты.*

* Type **ItemsType** [get]

*Возвращает тип объектов.*

### Подробное описание

Класс аргумента события при обновлении данных.

См. определение в файле **ItemsUpdatedEventArgs.cs** строка **6**

### Конструктор(ы)

#### ViewModel.Interfaces.AppStates.Sessions.ItemsUpdatedEventArgs.ItemsUpdatedEventArgs (UpdateItemsState state, IEnumerable< object > items, Type itemsType)

Создаёт экземпляр класса **ItemsUpdatedEventArgs**.

##### Аргументы

|  |  |
| --- | --- |
| *state* | Состояние. |
| *items* | Объекты. |
| *itemsType* | Тип объектов. |

См. определение в файле **ItemsUpdatedEventArgs.cs** строка **29**

### Полный список свойств

#### IEnumerable<object> ViewModel.Interfaces.AppStates.Sessions.ItemsUpdatedEventArgs.Items[get]

Возвращает объекты.

См. определение в файле **ItemsUpdatedEventArgs.cs** строка **16**

#### Type ViewModel.Interfaces.AppStates.Sessions.ItemsUpdatedEventArgs.ItemsType[get]

Возвращает тип объектов.

См. определение в файле **ItemsUpdatedEventArgs.cs** строка **21**

#### UpdateItemsState ViewModel.Interfaces.AppStates.Sessions.ItemsUpdatedEventArgs.State[get]

Возвращает состояние.

См. определение в файле **ItemsUpdatedEventArgs.cs** строка **11**

## Интерфейс ViewModel.Interfaces.AppStates.Settings.IThemeManager

Интерфейс менеджера тем.

Граф наследования:ViewModel.Interfaces.AppStates.Settings.IThemeManager:

### Свойства

* IEnumerable< object > **Themes** [get]

*Возвращает темы.*

* object **ActualTheme** [get, set]

*Возвращает и задаёт актуальную тему.*

### Подробное описание

Интерфейс менеджера тем.

См. определение в файле **IThemeManager.cs** строка **6**

### Полный список свойств

#### object ViewModel.Interfaces.AppStates.Settings.IThemeManager.ActualTheme[get], [set]

Возвращает и задаёт актуальную тему.

Замещается в **View.Implementations.AvaloniaThemeManager** (*стр.115*) и **ViewModel.Implementations.Mocks.MockThemeManager** (*стр.373*).

См. определение в файле **IThemeManager.cs** строка **16**

#### IEnumerable<object> ViewModel.Interfaces.AppStates.Settings.IThemeManager.Themes[get]

Возвращает темы.

Замещается в **View.Implementations.AvaloniaThemeManager** (*стр.115*) и **ViewModel.Implementations.Mocks.MockThemeManager** (*стр.373*).

См. определение в файле **IThemeManager.cs** строка **11**

## Интерфейс Model.Interfaces.ITimeInterval

Интерфейс временного интервала.

Граф наследования:Model.Interfaces.ITimeInterval:

### Свойства

* TimeSpan **Duration** [get]

*Возвращает длительность.*

### Подробное описание

Интерфейс временного интервала.

См. определение в файле **ITimeInterval.cs** строка **6**

### Полный список свойств

#### TimeSpan Model.Interfaces.ITimeInterval.Duration[get]

Возвращает длительность.

Замещается в **Model.Times.TimeIntervalElement** (*стр.493*), **Model.Times.TimeIntervalList** (*стр.506*) и **ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy** (*стр.500*).

См. определение в файле **ITimeInterval.cs** строка **11**

## Интерфейс Model.Interfaces.ITimeIntervalElement

Интерфейс элементарного временного интервала.

Граф наследования:Model.Interfaces.ITimeIntervalElement:

### Свойства

* DateTime **Start** [get, set]

*Возвращает и задаёт начало.*

* DateTime **End** [get, set]

*Возвращает и задаёт конец.*

#### Свойства унаследованные от Model.Interfaces.ITimeInterval

* TimeSpan **Duration** [get]

*Возвращает длительность.*

### Подробное описание

Интерфейс элементарного временного интервала.

Наследует **ITimeInterval**.

См. определение в файле **ITimeIntervalElement.cs** строка **9**

### Полный список свойств

#### DateTime Model.Interfaces.ITimeIntervalElement.End[get], [set]

Возвращает и задаёт конец.

Замещается в **Model.Times.TimeIntervalElement** (*стр.493*) и **ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy** (*стр.500*).

См. определение в файле **ITimeIntervalElement.cs** строка **19**

#### DateTime Model.Interfaces.ITimeIntervalElement.Start[get], [set]

Возвращает и задаёт начало.

Замещается в **Model.Times.TimeIntervalElement** (*стр.493*) и **ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy** (*стр.500*).

См. определение в файле **ITimeIntervalElement.cs** строка **14**

## Интерфейс ViewModel.Interfaces.DataManagers.ITimeIntervalElementsEditorProxy

Интерфейс заместителя элементарного временного интервала для редактирования.

Граф наследования:ViewModel.Interfaces.DataManagers.ITimeIntervalElementsEditorProxy:

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.Interfaces.DataManagers.Generals.IEditorService

* void **ApplyChanges** ()

*Применения изменений.*

#### Свойства унаследованные от ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T >

* new T **Target** [get, set]

*Возвращает и задаёт заменяемого объекта.*

#### Свойства унаследованные от Model.Interfaces.ITimeIntervalElement

* DateTime **Start** [get, set]

*Возвращает и задаёт начало.*

* DateTime **End** [get, set]

*Возвращает и задаёт конец.*

#### Свойства унаследованные от Model.Interfaces.ITimeInterval

* TimeSpan **Duration** [get]

*Возвращает длительность.*

### Подробное описание

Интерфейс заместителя элементарного временного интервала для редактирования.

Наследует **IEditorProxy<ITimeIntervalElement>** и **ITimeIntervalElement**.

См. определение в файле **ITimeIntervalElementsEditorProxy.cs** строка **14**

## Интерфейс Model.Interfaces.ITimeIntervalList

Интерфейс списка временных интервалов.

Граф наследования:Model.Interfaces.ITimeIntervalList:

### Дополнительные унаследованные члены

#### Свойства унаследованные от Model.Interfaces.ITimeInterval

* TimeSpan **Duration** [get]

*Возвращает длительность.*

### Подробное описание

Интерфейс списка временных интервалов.

Наследует **ITimeInterval**, IList<ITimeIntervalElement> и IList.

См. определение в файле **ITimeIntervalList.cs** строка **12**

## Интерфейс ViewModel.Interfaces.ITimeScheduler

Интерфейс планировщика времени, который отслеживает достижение временных меток.

Граф наследования:ViewModel.Interfaces.ITimeScheduler:

### Свойства

* IList< DateTime > **Timepoints** [get]

*Возращает список временных меток.*

### События

* EventHandler< DateTime > **TimepointReached**

*Событие, которое возникает при достижении временной метки.*

### Подробное описание

Интерфейс планировщика времени, который отслеживает достижение временных меток.

См. определение в файле **ITimeScheduler.cs** строка **6**

### Полный список свойств

#### IList<DateTime> ViewModel.Interfaces.ITimeScheduler.Timepoints[get]

Возращает список временных меток.

Замещается в **ViewModel.Implementations.AppStates.TimeScheduler** (*стр.512*).

См. определение в файле **ITimeScheduler.cs** строка **11**

### Cобытия

#### EventHandler<DateTime> ViewModel.Interfaces.ITimeScheduler.TimepointReached

Событие, которое возникает при достижении временной метки.

Параметр передаёт время достигнутой метки.

См. определение в файле **ITimeScheduler.cs** строка **19**

## Шаблон интерфейса ViewModel.Interfaces.ModelLearning.IUnsupervisedLearningController< T, R, D, DT, DR >

Интерфейс контроллера обучения модели обучения без учителя.

Граф наследования:ViewModel.Interfaces.ModelLearning.IUnsupervisedLearningController< T, R, D, DT, DR >:

### Свойства

* IUnsupervisedLearningModel< T, R > **LearningModel** [get, set]

*Возвращает и задаёт модель обучение без учителя.*

* IUnsupervisedLearningEvaluator< T, R > **LearningEvaluator** [get, set]

*Возвращает и задаёт оцениватель модели обучения без учителя.*

#### Свойства унаследованные от ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR >

* bool **IsValidModel** [get]

*Возвращает логическое значение, указывающее на достоверность модели обучения.*

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.Interfaces.ModelLearning.ILearningController< D, DT, DR >

* DR **Predict** (DT data)

*Выполняет предсказание на основе данных.*

#### Открытые члены унаследованные от ViewModel.Interfaces.ModelLearning.IModelTeacher< D >

* Task< bool > **Train** (IEnumerable< D > data)

*Выполняет обучение модели на основе данных.*

### Подробное описание

Интерфейс контроллера обучения модели обучения без учителя.

Наследует **ILearningController<D, DT, DR>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |
| *D* | Тип данных. |
| *DT* | Тип входных данных. |
| *DR* | Тип выходных данных. |

См. определение в файле **IUnsupervisedLearningController.cs** строка **16**

### Полный список свойств

#### IUnsupervisedLearningEvaluator<T, R> ViewModel.Interfaces.ModelLearning.IUnsupervisedLearningController< T, R, D, DT, DR >.LearningEvaluator[get], [set]

Возвращает и задаёт оцениватель модели обучения без учителя.

См. определение в файле **IUnsupervisedLearningController.cs** строка **27**

#### IUnsupervisedLearningModel<T, R> ViewModel.Interfaces.ModelLearning.IUnsupervisedLearningController< T, R, D, DT, DR >.LearningModel[get], [set]

Возвращает и задаёт модель обучение без учителя.

См. определение в файле **IUnsupervisedLearningController.cs** строка **22**

## Шаблон интерфейса MachineLearning.Interfaces.Generals.IUnsupervisedLearningConverter< T, R, D, DT, DR >

Интерфейс конвертора данных в данные для предсказания без учителя и наоборот.

Граф наследования:MachineLearning.Interfaces.Generals.IUnsupervisedLearningConverter< T, R, D, DT, DR >:

### Открытые члены

* IEnumerable< T > **FitConvertData** (IEnumerable< D > data)

*Адаптирует конвертацию данных под данные для предсказания.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ILearningConverter< T, R, DT, DR >

* T **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

### Подробное описание

Интерфейс конвертора данных в данные для предсказания без учителя и наоборот.

Наследует **ILearningConverter<T, R, DT, DR>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |
| *D* | Тип данных. |
| *DT* | Тип входных данных. |
| *DR* | Тип выходных данных. |

См. определение в файле **IUnsupervisedLearningConverter.cs** строка **14**

### Методы

#### IEnumerable< T > MachineLearning.Interfaces.Generals.IUnsupervisedLearningConverter< T, R, D, DT, DR >.FitConvertData (IEnumerable< D > data)

Адаптирует конвертацию данных под данные для предсказания.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает конвертированные входных данных для предсказания.

Замещается в **MachineLearning.Converters.BaseUnsupervisedLearningConverter< R, D, DT, DR >** (*стр.156*).

## Шаблон интерфейса MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R >

Интерфейс оценки модели обучения без учителя.

Граф наследования:MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R >:

### Открытые члены

* Task< **ScoreMetricCategory** > **Evaluate** (IEnumerable< T > data)

*Оценивает модель обучения.*

### Свойства

* IUnsupervisedLearningModel< T, R > **Model** [get, set]

*Возвращает и задаёт модель обучения без учителя.*

* IUnsupervisedScoreMetric< R, T > **ScoreMetric** [get, set]

*Возвращает и задаёт метрику оценки обучения с учителем.*

### Подробное описание

Интерфейс оценки модели обучения без учителя.

Наследует **ILearningEvaluator**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |

См. определение в файле **IUnsupervisedLearningEvaluator.cs** строка **11**

### Методы

#### Task< ScoreMetricCategory > MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R >.Evaluate (IEnumerable< T > data)

Оценивает модель обучения.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает категорию модели обучения.

Замещается в **MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >** (*стр.151*).

### Полный список свойств

#### IUnsupervisedLearningModel<T, R> MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R >.Model[get], [set]

Возвращает и задаёт модель обучения без учителя.

Замещается в **MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >** (*стр.152*).

См. определение в файле **IUnsupervisedLearningEvaluator.cs** строка **16**

#### IUnsupervisedScoreMetric<R, T> MachineLearning.Interfaces.Generals.IUnsupervisedLearningEvaluator< T, R >.ScoreMetric[get], [set]

Возвращает и задаёт метрику оценки обучения с учителем.

Замещается в **MachineLearning.LearningEvaluators.BaseUnsupervisedCrossValidationEvaluator< T, R >** (*стр.152*).

См. определение в файле **IUnsupervisedLearningEvaluator.cs** строка **21**

## Шаблон интерфейса MachineLearning.Interfaces.Generals.IUnsupervisedLearningModel< T, R >

Интерфейс модели обучение без учителя.

Граф наследования:MachineLearning.Interfaces.Generals.IUnsupervisedLearningModel< T, R >:

### Открытые члены

* Task **Train** (IEnumerable< T > data)

*Обучает модель на основе данных.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ILearningModel< T, R >

* R **Predict** (T data)

*Предсказывает значение на основе данных.*

### Подробное описание

Интерфейс модели обучение без учителя.

Наследует **ILearningModel<T, R>**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |

См. определение в файле **IUnsupervisedLearningModel.cs** строка **11**

### Методы

#### Task MachineLearning.Interfaces.Generals.IUnsupervisedLearningModel< T, R >.Train (IEnumerable< T > data)

Обучает модель на основе данных.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает задачу процесса обучения.

## Шаблон интерфейса MachineLearning.Interfaces.Generals.IUnsupervisedScoreMetric< T, D >

Интерфейс метрики оценки для модели обучения без учителя.

Граф наследования:MachineLearning.Interfaces.Generals.IUnsupervisedScoreMetric< T, D >:

### Открытые члены

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< D > data)

*Вычисляет оценку на основе фактических данных и данных для предсказания.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IScoreMetric

* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

### Подробное описание

Интерфейс метрики оценки для модели обучения без учителя.

Наследует **IScoreMetric**.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип фактических данных для оценки. |
| *D* | Тип данных для предсказания для оценки. |

См. определение в файле **IUnsupervisedScoreMetric.cs** строка **11**

### Методы

#### double MachineLearning.Interfaces.Generals.IUnsupervisedScoreMetric< T, D >.CalculateScore (IEnumerable< T > actual, IEnumerable< D > data)

Вычисляет оценку на основе фактических данных и данных для предсказания.

##### Аргументы

|  |  |
| --- | --- |
| *actual* | Коллекция фактических данных. |
| *data* | Коллекция данных для предсказания. |

##### Возвращает

Возвращает значение оценки.

## Класс ViewModel.Implementations.AppStates.JsonSerializer

Класс Json-сериализатора.

Граф наследования:ViewModel.Implementations.AppStates.JsonSerializer:

### Открытые члены

* T? **Deserialize< T >** (byte[] data)

*Десериализует объект.*

##### *Параметры шаблона*

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает десериализованный объект.

* byte[] **Serialize** (object value)

*Сериализует объект.*

##### *Аргументы*

|  |  |
| --- | --- |
| *value* | Объект. |

##### *Возвращает*

Возвращает массив байтов, представляющий сериализованные данные.

### Свойства

* JsonSerializerSettings **Settings** [get]

*Настройки.*

### Подробное описание

Класс Json-сериализатора.

Реализует **ISerializer**.

См. определение в файле **JsonSerializer.cs** строка **14**

### Методы

#### T? ViewModel.Implementations.AppStates.JsonSerializer.Deserialize< T > (byte[] data)

Десериализует объект.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает десериализованный объект.

Замещает **ViewModel.Interfaces.AppStates.ISerializer** (*стр.276*).

См. определение в файле **JsonSerializer.cs** строка **27**

#### byte[] ViewModel.Implementations.AppStates.JsonSerializer.Serialize (object value)

Сериализует объект.

##### Аргументы

|  |  |
| --- | --- |
| *value* | Объект. |

##### Возвращает

Возвращает массив байтов, представляющий сериализованные данные.

Замещает **ViewModel.Interfaces.AppStates.ISerializer** (*стр.276*).

См. определение в файле **JsonSerializer.cs** строка **34**

### Полный список свойств

#### JsonSerializerSettings ViewModel.Implementations.AppStates.JsonSerializer.Settings[get]

**Инициализатор**

= new()

{

ObjectCreationHandling = ObjectCreationHandling.Replace,

TypeNameHandling = TypeNameHandling.All,

Formatting = Formatting.Indented

}

Настройки.

См. определение в файле **JsonSerializer.cs** строка **19**

## Класс MachineLearning.LearningModels.KMeanLearningModel

Класс модель обучения кластеризации с алгоритмом K-средних.

Граф наследования:MachineLearning.LearningModels.KMeanLearningModel:

### Открытые члены

* Task **Train** (IEnumerable< IEnumerable< double > > data)
* int **Predict** (IEnumerable< double > data)

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IUnsupervisedLearningModel< T, R >

* Task **Train** (IEnumerable< T > data)

*Обучает модель на основе данных.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ILearningModel< T, R >

* R **Predict** (T data)

*Предсказывает значение на основе данных.*

### Свойства

* int **NumberOfClusters** [get, set]

*Возвращает и задаёт количество кластеров.*

* **IPointDistanceMetric** **DistanceMetric** [get, set]

*Возвращает и задаёт метрику расстояний.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс модель обучения кластеризации с алгоритмом K-средних.

Наследует **TrackableObject**. Реализует **IClusteringModel**.

См. определение в файле **KMeanLearningModel.cs** строка **16**

### Методы

#### Task MachineLearning.LearningModels.KMeanLearningModel.Train (IEnumerable< IEnumerable< double > > data)

См. определение в файле **KMeanLearningModel.cs** строка **57**

### Полный список свойств

#### IPointDistanceMetric MachineLearning.LearningModels.KMeanLearningModel.DistanceMetric[get], [set]

Возвращает и задаёт метрику расстояний.

См. определение в файле **KMeanLearningModel.cs** строка **50**

#### int MachineLearning.LearningModels.KMeanLearningModel.NumberOfClusters[get], [set]

Возвращает и задаёт количество кластеров.

См. определение в файле **KMeanLearningModel.cs** строка **41**

## Класс MachineLearning.LearningModels.KNearestNeighborsModel

Класс модель обучения классификации с алгоритмом K-ближащих соседей.

Граф наследования:MachineLearning.LearningModels.KNearestNeighborsModel:

### Открытые члены

* Task **Train** (IEnumerable< IEnumerable< double > > data, IEnumerable< int > targets)
* int **Predict** (IEnumerable< double > data)

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedLearningModel< T, R >

* Task **Train** (IEnumerable< T > data, IEnumerable< R > targets)

*Обучает модель на основе данных и целевых значений.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ILearningModel< T, R >

* R **Predict** (T data)

*Предсказывает значение на основе данных.*

### Свойства

* int **NumberOfNeighbors** [get, set]

*Возвращает и задаёт количество соседей.*

* **IPointDistanceMetric** **DistanceMetric** [get, set]

*Возвращает и задаёт метрику расстояний.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс модель обучения классификации с алгоритмом K-ближащих соседей.

Наследует **TrackableObject**. Реализует **IClassificationModel**.

См. определение в файле **KNearestNeighborsModel.cs** строка **16**

### Методы

#### Task MachineLearning.LearningModels.KNearestNeighborsModel.Train (IEnumerable< IEnumerable< double > > data, IEnumerable< int > targets)

См. определение в файле **KNearestNeighborsModel.cs** строка **57**

### Полный список свойств

#### IPointDistanceMetric MachineLearning.LearningModels.KNearestNeighborsModel.DistanceMetric[get], [set]

Возвращает и задаёт метрику расстояний.

См. определение в файле **KNearestNeighborsModel.cs** строка **50**

#### int MachineLearning.LearningModels.KNearestNeighborsModel.NumberOfNeighbors[get], [set]

Возвращает и задаёт количество соседей.

См. определение в файле **KNearestNeighborsModel.cs** строка **41**

## Шаблон структуры MachineLearning.LearningModelData< T, R >

Структура хранения данных для модели обучения.

### Открытые члены

* **LearningModelData** (IEnumerable< T > data, IEnumerable< R > targets)

*Создаёт экземпляр класса* ***LearningModelData<T, R>****.*

### Свойства

* IEnumerable< T > **Data** [get]

*Возвращает и задаёт данные.*

* IEnumerable< R > **Targets** [get]

*Возвращает и задаёт целевые значения.*

### Подробное описание

Структура хранения данных для модели обучения.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип входных данных для предсказания. |
| *R* | Тип выходных данных для предсказания. |

См. определение в файле **LearningModelData.cs** строка **8**

### Методы

#### MachineLearning.LearningModelData< T, R >.LearningModelData (IEnumerable< T > data, IEnumerable< R > targets)

Создаёт экземпляр класса **LearningModelData<T, R>**.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |
| *targets* | целевые значения. |

См. определение в файле **LearningModelData.cs** строка **25**

### Полный список свойств

#### IEnumerable<T> MachineLearning.LearningModelData< T, R >.Data[get]

Возвращает и задаёт данные.

См. определение в файле **LearningModelData.cs** строка **13**

#### IEnumerable<R> MachineLearning.LearningModelData< T, R >.Targets[get]

Возвращает и задаёт целевые значения.

См. определение в файле **LearningModelData.cs** строка **18**

## Класс View.Desktop.Linux.LinuxNotificationManager

Класс менеджера уведомлений **Linux**.

Граф наследования:View.Desktop.Linux.LinuxNotificationManager:

### Открытые члены

* void **SendNotification** (string description, string title)

*Отправляет уведомление.*

##### *Аргументы*

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

### Подробное описание

Класс менеджера уведомлений **Linux**.

Реализует **INotificationManager**.

См. определение в файле **LinuxNotificationManager.cs** строка **15**

### Методы

#### void View.Desktop.Linux.LinuxNotificationManager.SendNotification (string description, string title)

Отправляет уведомление.

##### Аргументы

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

Замещает **ViewModel.Interfaces.AppStates.INotificationManager** (*стр.261*).

См. определение в файле **LinuxNotificationManager.cs** строка **28**

## Класс MachineLearning.DataProcessors.LowVariationColumnProcessor

Класс обработчика столбцов для устранения низкой вариативности.

Граф наследования:MachineLearning.DataProcessors.LowVariationColumnProcessor:

### Открытые члены

* DataProcessorResult< IEnumerable< double > > **Process** (IEnumerable< IEnumerable< double > > data)

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IDataProcessor< T, R >

* DataProcessorResult< R > **Process** (IEnumerable< T > data)

*Преобразует данные.*

### Подробное описание

Класс обработчика столбцов для устранения низкой вариативности.

Реализует **IPointDataProcessor**.

См. определение в файле **LowVariationColumnProcessor.cs** строка **14**

### Методы

#### DataProcessorResult< IEnumerable< double > > MachineLearning.DataProcessors.LowVariationColumnProcessor.Process (IEnumerable< IEnumerable< double > > data)

См. определение в файле **LowVariationColumnProcessor.cs** строка **28**

## Класс View.Desktop.macOS.MacOsNotificationManager

Класс менеджера уведомлений **macOS**.

Граф наследования:View.Desktop.macOS.MacOsNotificationManager:

### Открытые члены

* void **SendNotification** (string description, string title)

*Отправляет уведомление.*

##### *Аргументы*

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

### Подробное описание

Класс менеджера уведомлений **macOS**.

Реализует **INotificationManager**.

См. определение в файле **MacOsNotificationManager.cs** строка **13**

### Методы

#### void View.Desktop.macOS.MacOsNotificationManager.SendNotification (string description, string title)

Отправляет уведомление.

##### Аргументы

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

Замещает **ViewModel.Interfaces.AppStates.INotificationManager** (*стр.261*).

См. определение в файле **MacOsNotificationManager.cs** строка **18**

## Класс View.Android.MainActivity

Класс основной активности.

Граф наследования:View.Android.MainActivity:

### Открытые члены

* **MainActivity** ()

*Создаёт экземпляр класса* ***MainActivity*** *по умолчанию.*

### Защищенные члены

* override AppBuilder **CustomizeAppBuilder** (AppBuilder builder)
* override AppBuilder **CreateAppBuilder** ()

### События

* EventHandler **AppDeactivated**

#### События унаследованные от ViewModel.Interfaces.AppStates.IAppLifeState

* EventHandler **AppDeactivated**

*Событие, которое возникает, когда приложение деактивируется.*

### Подробное описание

Класс основной активности.

Наследует AvaloniaMainActivity<App>. Реализует **IAppLifeState**.

См. определение в файле **MainActivity.cs** строка **27**

### Конструктор(ы)

#### View.Android.MainActivity.MainActivity ()

Создаёт экземпляр класса **MainActivity** по умолчанию.

См. определение в файле **MainActivity.cs** строка **35**

### Методы

#### override AppBuilder View.Android.MainActivity.CreateAppBuilder ()[protected]

См. определение в файле **MainActivity.cs** строка **49**

#### override AppBuilder View.Android.MainActivity.CustomizeAppBuilder (AppBuilder builder)[protected]

См. определение в файле **MainActivity.cs** строка **41**

### Cобытия

#### EventHandler View.Android.MainActivity.AppDeactivated

См. определение в файле **MainActivity.cs** строка **30**

## Класс View.Views.MainView

Класс основного пользовательского элемента приложения.

Граф наследования:View.Views.MainView:

### Открытые члены

* **MainView** ()

*Создаёт экземпляр класса* ***MainView*** *по умолчанию.*

### Статические открытые данные

* static readonly StyledProperty< bool > **IsPaneOpenProperty**

*Свойство стиля* ***IsPaneOpen****.*

* static readonly StyledProperty< double > **InputPaneHeightProperty**

*Свойство стиля* ***InputPaneHeight****.*

* static readonly StyledProperty< bool > **IsInputPaneVisibleProperty**

*Свойство стиля* ***IsInputPaneVisible****.*

### Защищенные члены

* override void **OnLoaded** (RoutedEventArgs e)

### Свойства

* bool **IsPaneOpen** [get, set]

*Возвращает и задаёт логическкое значение, указывающее, открыта ли панель меню.*

* double **InputPaneHeight** [get, set]

*Возвращает и задаёт высоту панели ввода.*

* bool **IsInputPaneVisible** [get, set]

*Возвращает и задаёт логическкое значение, указывающее, видна ли панель ввода.*

### Подробное описание

Класс основного пользовательского элемента приложения.

Наследует ReactiveUserControl<MainViewModel>.

См. определение в файле **MainView.axaml.cs** строка **18**

### Конструктор(ы)

#### View.Views.MainView.MainView ()

Создаёт экземпляр класса **MainView** по умолчанию.

См. определение в файле **MainView.axaml.cs** строка **68**

### Методы

#### override void View.Views.MainView.OnLoaded (RoutedEventArgs e)[protected]

См. определение в файле **MainView.axaml.cs** строка **76**

### Данные класса

#### readonly StyledProperty<double> View.Views.MainView.InputPaneHeightProperty[static]

**Инициализатор**

=

AvaloniaProperty.Register<MainView, double>(nameof(InputPaneHeight), defaultValue: 320)

Свойство стиля **InputPaneHeight**.

См. определение в файле **MainView.axaml.cs** строка **29**

#### readonly StyledProperty<bool> View.Views.MainView.IsInputPaneVisibleProperty[static]

**Инициализатор**

=

AvaloniaProperty.Register<MainView, bool>(nameof(IsInputPaneVisible), defaultValue: false)

Свойство стиля **IsInputPaneVisible**.

См. определение в файле **MainView.axaml.cs** строка **35**

#### readonly StyledProperty<bool> View.Views.MainView.IsPaneOpenProperty[static]

**Инициализатор**

=

AvaloniaProperty.Register<MainView, bool>(nameof(IsPaneOpen), defaultValue: false)

Свойство стиля **IsPaneOpen**.

См. определение в файле **MainView.axaml.cs** строка **23**

### Полный список свойств

#### double View.Views.MainView.InputPaneHeight[get], [set]

Возвращает и задаёт высоту панели ввода.

См. определение в файле **MainView.axaml.cs** строка **50**

#### bool View.Views.MainView.IsInputPaneVisible[get], [set]

Возвращает и задаёт логическкое значение, указывающее, видна ли панель ввода.

См. определение в файле **MainView.axaml.cs** строка **59**

#### bool View.Views.MainView.IsPaneOpen[get], [set]

Возвращает и задаёт логическкое значение, указывающее, открыта ли панель меню.

См. определение в файле **MainView.axaml.cs** строка **41**

## Класс ViewModel.ViewModels.MainViewModel

Класс главного контроллера.

Граф наследования:ViewModel.ViewModels.MainViewModel:

### Открытые члены

* **MainViewModel** (IEnumerable< **BasePageViewModel** > pages, **ISettings** settings, **ISession** session, **IAppLifeState** appLifeState, **ILogger** logger, IEnumerable< IModelTeacher< **ITaskElement** > > modelTeachers)

*Создаёт экземпляр класса* ***MainViewModel****.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс главного контроллера.

Наследует **BaseViewModel**.

См. определение в файле **MainViewModel.cs** строка **21**

### Конструктор(ы)

#### ViewModel.ViewModels.MainViewModel.MainViewModel (IEnumerable< BasePageViewModel > pages, ISettings settings, ISession session, IAppLifeState appLifeState, ILogger logger, IEnumerable< IModelTeacher< ITaskElement > > modelTeachers)

Создаёт экземпляр класса **MainViewModel**.

##### Аргументы

|  |  |
| --- | --- |
| *pages* | Страницы. |
| *settings* | Настройки. |
| *session* | Сессия. |
| *appLifeState* | Управление жизненным циклом приложения. |
| *logger* | Логгирование. |
| *modelTeachers* | Учителя моделей обучения на элементарных задачах. |

См. определение в файле **MainViewModel.cs** строка **69**

## Класс View.Views.MainWindow

Класс основного окна приложения.

Граф наследования:View.Views.MainWindow:

### Открытые члены

* **MainWindow** ()

*Создаёт экземпляр класса* ***MainWindow*** *по умолчанию.*

### Подробное описание

Класс основного окна приложения.

Наследует Window.

См. определение в файле **MainWindow.axaml.cs** строка **11**

### Конструктор(ы)

#### View.Views.MainWindow.MainWindow ()

Создаёт экземпляр класса **MainWindow** по умолчанию.

См. определение в файле **MainWindow.axaml.cs** строка **16**

## Класс ViewModel.Implementations.ModelLearning.Converters.MetadataCategoriesTransformer

Класс преобразования категории метаданных в данные для предсказания.

Граф наследования:ViewModel.Implementations.ModelLearning.Converters.MetadataCategoriesTransformer:

### Открытые члены

* IEnumerable< int?> **FitTransform** (IEnumerable< **TaskMetadata** > data)
* int? **Transform** (**TaskMetadata** data)

#### Открытые члены унаследованные от ViewModel.Interfaces.ModelLearning.IDataTransformer< T, R >

* IEnumerable< R > **FitTransform** (IEnumerable< T > data)

*Адаптирует преобразование под данные.*

* R **Transform** (T data)

*Преобразует данные.*

### Подробное описание

Класс преобразования категории метаданных в данные для предсказания.

Реализует **IDataTransformer<Metadata, int?>**.

См. определение в файле **MetadataCategoriesTransformer.cs** строка **12**

### Методы

#### IEnumerable< int?> ViewModel.Implementations.ModelLearning.Converters.MetadataCategoriesTransformer.FitTransform (IEnumerable< TaskMetadata > data)

См. определение в файле **MetadataCategoriesTransformer.cs** строка **20**

#### int? ViewModel.Implementations.ModelLearning.Converters.MetadataCategoriesTransformer.Transform (TaskMetadata data)

См. определение в файле **MetadataCategoriesTransformer.cs** строка **38**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Entities.MetadataEntity

Класс сущности метаданных.

### Свойства

* int **TaskId** [get, set]

*Возвращает и задаёт индентификатор задачи.*

* string **Title** [get, set]

*Возвращает и задаёт название.*

* string? **Description** [get, set]

*Возвращает и задаёт описание.*

* string? **Category** [get, set]

*Возвращает и задаёт категории.*

* virtual **TaskEntity** **Task** [get, set]

*Возвращает и задаёт задачу.*

* virtual ICollection< **TagEntity** > **Tags** [get, set]

*Возвращает и задаёт теги.*

### Подробное описание

Класс сущности метаданных.

См. определение в файле **MetadataEntity.cs** строка **13**

### Полный список свойств

#### string? ViewModel.Implementations.AppStates.Sessions.Database.Entities.MetadataEntity.Category[get], [set]

Возвращает и задаёт категории.

См. определение в файле **MetadataEntity.cs** строка **33**

#### string? ViewModel.Implementations.AppStates.Sessions.Database.Entities.MetadataEntity.Description[get], [set]

Возвращает и задаёт описание.

См. определение в файле **MetadataEntity.cs** строка **28**

#### virtual ICollection<TagEntity> ViewModel.Implementations.AppStates.Sessions.Database.Entities.MetadataEntity.Tags[get], [set]

Возвращает и задаёт теги.

См. определение в файле **MetadataEntity.cs** строка **44**

#### virtual TaskEntity ViewModel.Implementations.AppStates.Sessions.Database.Entities.MetadataEntity.Task[get], [set]

Возвращает и задаёт задачу.

См. определение в файле **MetadataEntity.cs** строка **39**

#### int ViewModel.Implementations.AppStates.Sessions.Database.Entities.MetadataEntity.TaskId[get], [set]

Возвращает и задаёт индентификатор задачи.

См. определение в файле **MetadataEntity.cs** строка **18**

#### string ViewModel.Implementations.AppStates.Sessions.Database.Entities.MetadataEntity.Title[get], [set]

Возвращает и задаёт название.

См. определение в файле **MetadataEntity.cs** строка **23**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Mappers.MetadataMapper

Класс перобразования значений метаданных между двумя предметными областями.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.Mappers.MetadataMapper:

### Открытые члены

* object **Map** (**MetadataEntity** value)
* **MetadataEntity** **MapBack** (object value)

#### Открытые члены унаследованные от ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 >

* T2 **Map** (T1 value)

*Преобразует исходные данные в целевые данные.*

* T1 **MapBack** (T2 value)

*Преобразует целевые данные в исходные данные.*

### Подробное описание

Класс перобразования значений метаданных между двумя предметными областями.

Реализует **IMapper<MetadataEntity, object>**.

См. определение в файле **MetadataMapper.cs** строка **13**

### Методы

#### object ViewModel.Implementations.AppStates.Sessions.Database.Mappers.MetadataMapper.Map (MetadataEntity value)

См. определение в файле **MetadataMapper.cs** строка **16**

#### MetadataEntity ViewModel.Implementations.AppStates.Sessions.Database.Mappers.MetadataMapper.MapBack (object value)

См. определение в файле **MetadataMapper.cs** строка **28**

## Класс ViewModel.Implementations.ModelLearning.Converters.MetadataTagsTransformer

Класс преобразования тегов метаданных в данные для предсказания.

Граф наследования:ViewModel.Implementations.ModelLearning.Converters.MetadataTagsTransformer:

### Открытые члены

* IEnumerable< IEnumerable< int > > **FitTransform** (IEnumerable< **TaskMetadata** > data)
* IEnumerable< int > **Transform** (**TaskMetadata** data)

#### Открытые члены унаследованные от ViewModel.Interfaces.ModelLearning.IDataTransformer< T, R >

* IEnumerable< R > **FitTransform** (IEnumerable< T > data)

*Адаптирует преобразование под данные.*

* R **Transform** (T data)

*Преобразует данные.*

### Подробное описание

Класс преобразования тегов метаданных в данные для предсказания.

Реализует **IDataTransformer<Metadata, IEnumerable<int>>**.

См. определение в файле **MetadataTagsTransformer.cs** строка **12**

### Методы

#### IEnumerable< IEnumerable< int > > ViewModel.Implementations.ModelLearning.Converters.MetadataTagsTransformer.FitTransform (IEnumerable< TaskMetadata > data)

См. определение в файле **MetadataTagsTransformer.cs** строка **20**

#### IEnumerable< int > ViewModel.Implementations.ModelLearning.Converters.MetadataTagsTransformer.Transform (TaskMetadata data)

См. определение в файле **MetadataTagsTransformer.cs** строка **39**

## Класс MachineLearning.LearningModels.MetricAdapter

Адаптер для использования метрики расстояния **IPointDistanceMetric** с интерфейсом IMetric<double[]>.

Граф наследования:MachineLearning.LearningModels.MetricAdapter:

### Открытые члены

* double **Distance** (double[] x, double[] y)

### Свойства

* **IPointDistanceMetric** **DistanceMetric** [get, set]

*Возвращает и задаёт метрику дистанцию.*

### Подробное описание

Адаптер для использования метрики расстояния **IPointDistanceMetric** с интерфейсом IMetric<double[]>.

Реализует IMetric<double[]>.

См. определение в файле **MetricAdapter.cs** строка **14**

### Полный список свойств

#### IPointDistanceMetric MachineLearning.LearningModels.MetricAdapter.DistanceMetric[get], [set]

Возвращает и задаёт метрику дистанцию.

См. определение в файле **MetricAdapter.cs** строка **19**

## Класс MachineLearning.Scalers.MinMaxScaler

Класс масштабрирования данных с помощью минимума и максимума.

Граф наследования:MachineLearning.Scalers.MinMaxScaler:

### Открытые члены

* IEnumerable< double > **FitTransform** (IEnumerable< double > data)

*Адаптирует преобразования под заданные данные.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает коллекцию нормализованных данных.

* double **Transform** (double value)

*Преобразует значение.*

##### *Аргументы*

|  |  |
| --- | --- |
| *value* | Значение. |

##### *Возвращает*

Возвращает нормализованное значение.

### Свойства

* double **Min** [get]

*Возвращает минимальное значение.*

* double **Max** [get]

*Возвращает максимальное значение.*

### Подробное описание

Класс масштабрирования данных с помощью минимума и максимума.

Реализует **IScaler**.

См. определение в файле **MinMaxScaler.cs** строка **11**

### Методы

#### IEnumerable< double > MachineLearning.Scalers.MinMaxScaler.FitTransform (IEnumerable< double > data)

Адаптирует преобразования под заданные данные.

##### Аргументы

|  |  |
| --- | --- |
| *data* | Данные. |

##### Возвращает

Возвращает коллекцию нормализованных данных.

Замещает **MachineLearning.Interfaces.IScaler** (*стр.273*).

См. определение в файле **MinMaxScaler.cs** строка **24**

#### double MachineLearning.Scalers.MinMaxScaler.Transform (double value)

Преобразует значение.

##### Аргументы

|  |  |
| --- | --- |
| *value* | Значение. |

##### Возвращает

Возвращает нормализованное значение.

Замещает **MachineLearning.Interfaces.IScaler** (*стр.273*).

См. определение в файле **MinMaxScaler.cs** строка **39**

### Полный список свойств

#### double MachineLearning.Scalers.MinMaxScaler.Max[get]

Возвращает максимальное значение.

См. определение в файле **MinMaxScaler.cs** строка **21**

#### double MachineLearning.Scalers.MinMaxScaler.Min[get]

Возвращает минимальное значение.

См. определение в файле **MinMaxScaler.cs** строка **16**

## Класс ViewModel.Implementations.Mocks.MockAppLifeState

Класс-заглушка управления состоянием жизненного цикла приложения.

Граф наследования:ViewModel.Implementations.Mocks.MockAppLifeState:

### Открытые члены

* void **DeactivateApp** ()

*Деактивирует приложение.*

### События

* EventHandler **AppDeactivated**

#### События унаследованные от ViewModel.Interfaces.AppStates.IAppLifeState

* EventHandler **AppDeactivated**

*Событие, которое возникает, когда приложение деактивируется.*

### Подробное описание

Класс-заглушка управления состоянием жизненного цикла приложения.

Реализует **IAppLifeState**.

См. определение в файле **MockAppLifeState.cs** строка **11**

### Cобытия

#### EventHandler ViewModel.Implementations.Mocks.MockAppLifeState.AppDeactivated

См. определение в файле **MockAppLifeState.cs** строка **14**

## Класс ViewModel.Implementations.Mocks.MockLocalizationManager

Класс-заглушка менеджера локализаций.

Граф наследования:ViewModel.Implementations.Mocks.MockLocalizationManager:

### Свойства

* IEnumerable< CultureInfo > **Localizations** [get, set]

*Возвращает локализации.*

* CultureInfo **ActualLocalization** [get, set]

*Возвращает и задаёт актуальную локализацию.*

### Подробное описание

Класс-заглушка менеджера локализаций.

Реализует **ILocalizationManager**.

См. определение в файле **MockLocalizationManager.cs** строка **13**

### Полный список свойств

#### CultureInfo ViewModel.Implementations.Mocks.MockLocalizationManager.ActualLocalization[get], [set]

Возвращает и задаёт актуальную локализацию.

Замещает **ViewModel.Interfaces.AppStates.Settings.ILocalizationManager** (*стр.256*).

См. определение в файле **MockLocalizationManager.cs** строка **32**

#### IEnumerable<CultureInfo> ViewModel.Implementations.Mocks.MockLocalizationManager.Localizations[get], [set]

Возвращает локализации.

Замещает **ViewModel.Interfaces.AppStates.Settings.ILocalizationManager** (*стр.256*).

См. определение в файле **MockLocalizationManager.cs** строка **21**

## Класс ViewModel.Implementations.Mocks.MockNotificationManager

Класс-заглушка менеджера уведомлений.

Граф наследования:ViewModel.Implementations.Mocks.MockNotificationManager:

### Открытые члены

* void **SendNotification** (string description, string title)

*Отправляет уведомление.*

##### *Аргументы*

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

### События

* EventHandler< **NotificationEventArgs** > **NotificationSended**

*Событие, которое возникает при отправке уведомления.*

### Подробное описание

Класс-заглушка менеджера уведомлений.

Реализует **INotificationManager**.

См. определение в файле **MockNotificationManager.cs** строка **11**

### Методы

#### void ViewModel.Implementations.Mocks.MockNotificationManager.SendNotification (string description, string title)

Отправляет уведомление.

##### Аргументы

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

Замещает **ViewModel.Interfaces.AppStates.INotificationManager** (*стр.261*).

### Cобытия

#### EventHandler<NotificationEventArgs> ViewModel.Implementations.Mocks.MockNotificationManager.NotificationSended

Событие, которое возникает при отправке уведомления.

См. определение в файле **MockNotificationManager.cs** строка **16**

## Класс ViewModel.Implementations.Mocks.MockResourceService

Класс-заглушка сервиса ресурсов.

Граф наследования:ViewModel.Implementations.Mocks.MockResourceService:

### Открытые члены

* object? **GetResource** (object key)

*Возвращает ресурс по ключу.*

##### *Аргументы*

|  |  |
| --- | --- |
| *key* | Ключ. |

##### *Возвращает*

Возвращает ресурс.

### Свойства

* Dictionary< object, object > **Resources** = new() [get, set]

*Возвращает и задаёт ресурсы.*

### Подробное описание

Класс-заглушка сервиса ресурсов.

Реализует **IResourceService**.

См. определение в файле **MockResourceService.cs** строка **11**

### Методы

#### object? ViewModel.Implementations.Mocks.MockResourceService.GetResource (object key)

Возвращает ресурс по ключу.

##### Аргументы

|  |  |
| --- | --- |
| *key* | Ключ. |

##### Возвращает

Возвращает ресурс.

Замещает **ViewModel.Interfaces.AppStates.IResourceService** (*стр.272*).

См. определение в файле **MockResourceService.cs** строка **19**

### Полный список свойств

#### Dictionary<object, object> ViewModel.Implementations.Mocks.MockResourceService.Resources = new()[get], [set]

Возвращает и задаёт ресурсы.

См. определение в файле **MockResourceService.cs** строка **16**

## Класс ViewModel.Implementations.Mocks.MockThemeManager

Класс-заглушка менеджера тем.

Граф наследования:ViewModel.Implementations.Mocks.MockThemeManager:

### Свойства

* IEnumerable< object > **Themes** [get, set]

*Возвращает темы.*

* object **ActualTheme** [get, set]

*Возвращает и задаёт актуальную тему.*

### Подробное описание

Класс-заглушка менеджера тем.

Реализует **IThemeManager**.

См. определение в файле **MockThemeManager.cs** строка **11**

### Полный список свойств

#### object ViewModel.Implementations.Mocks.MockThemeManager.ActualTheme[get], [set]

Возвращает и задаёт актуальную тему.

Замещает **ViewModel.Interfaces.AppStates.Settings.IThemeManager** (*стр.317*).

См. определение в файле **MockThemeManager.cs** строка **30**

#### IEnumerable<object> ViewModel.Implementations.Mocks.MockThemeManager.Themes[get], [set]

Возвращает темы.

Замещает **ViewModel.Interfaces.AppStates.Settings.IThemeManager** (*стр.317*).

См. определение в файле **MockThemeManager.cs** строка **19**

## Класс View.Views.Modals.MoveTasksView

Класс пользовательского элемента диалога перемещения задач.

Граф наследования:View.Views.Modals.MoveTasksView:

### Открытые члены

* **MoveTasksView** ()

*Создаёт экземпляр класса* ***MoveTasksView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента диалога перемещения задач.

Наследует ReactiveUserControl<MoveTasksViewModel>.

См. определение в файле **MoveTasksView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Modals.MoveTasksView.MoveTasksView ()

Создаёт экземпляр класса **MoveTasksView** по умолчанию.

См. определение в файле **MoveTasksView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Modals.MoveTasksViewModel

Класс диалога перемещения задач.

Граф наследования:ViewModel.ViewModels.Modals.MoveTasksViewModel:

### Открытые члены

* **MoveTasksViewModel** ()

*Создаёт экземпляр класса* ***MoveTasksViewModel*** *по умолчанию.*

#### Открытые члены унаследованные от ViewModel.ViewModels.Modals.TasksViewModel< A, R >

* **TasksViewModel** ()

*Создаёт экземпляр класса* ***TasksViewModel<A, R>*** *по умолчанию.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* async Task< R > **Invoke** (**BaseViewModel** parent, A args)

*Вызывает диалог.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Защищенные члены

* override void **GetArgs** (**ItemsTasksViewModelArgs** args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.Modals.TasksViewModel< A, R >

* override void **GetArgs** (A args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* void **GetArgs** (A args)

*Получает аргументы, переданные в диалог.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.Modals.TasksViewModel< A, R >

* IObservable< bool > **\_canExecuteGoToPrevious**

*Наблюдатель, который отслеживает возможность выполнения GoToPrevious.*

* IObservable< bool > **\_canExecuteGo**

*Наблюдатель, который отслеживает возможность выполнения Go.*

* IEnumerable< **ITask** > **\_mainList**

*Основной список.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* TaskCompletionSource< R >? **\_taskSource**

*Источник для завершения задачи, который используется для возврата результата диалога.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* **BaseViewModel**? **Parent** [get]

*Возвращает родителя.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс диалога перемещения задач.

Наследует **BaseDialogViewModel<ItemsTasksViewModelArgs, IEnumerable<ITask>?>**.

См. определение в файле **MoveTasksViewModel.cs** строка **15**

### Конструктор(ы)

#### ViewModel.ViewModels.Modals.MoveTasksViewModel.MoveTasksViewModel ()

Создаёт экземпляр класса **MoveTasksViewModel** по умолчанию.

См. определение в файле **MoveTasksViewModel.cs** строка **21**

### Методы

#### override void ViewModel.ViewModels.Modals.MoveTasksViewModel.GetArgs (ItemsTasksViewModelArgs args)[protected]

См. определение в файле **MoveTasksViewModel.cs** строка **29**

## Класс MachineLearning.LearningModels.MultipleLinearRegressionModel

Класс модель обучения регрессии с алгоритмом мультилинейной регрессии.

Граф наследования:MachineLearning.LearningModels.MultipleLinearRegressionModel:

### Открытые члены

* Task **Train** (IEnumerable< IEnumerable< double > > data, IEnumerable< double > targets)
* double **Predict** (IEnumerable< double > data)

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedLearningModel< T, R >

* Task **Train** (IEnumerable< T > data, IEnumerable< R > targets)

*Обучает модель на основе данных и целевых значений.*

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ILearningModel< T, R >

* R **Predict** (T data)

*Предсказывает значение на основе данных.*

### Подробное описание

Класс модель обучения регрессии с алгоритмом мультилинейной регрессии.

Реализует **IRegressionModel**.

См. определение в файле **MultipleLinearRegressionModel.cs** строка **13**

### Методы

#### Task MachineLearning.LearningModels.MultipleLinearRegressionModel.Train (IEnumerable< IEnumerable< double > > data, IEnumerable< double > targets)

См. определение в файле **MultipleLinearRegressionModel.cs** строка **21**

## Класс ViewModel.Implementations.Mocks.NotificationEventArgs

Класс аргументов события уведомления.

Граф наследования:ViewModel.Implementations.Mocks.NotificationEventArgs:

### Открытые члены

* **NotificationEventArgs** (string description, string title)

*Создаёт экземпяр класса* ***NotificationEventArgs****.*

### Свойства

* string **Description** [get]

*Возвращает описание.*

* string **Title** [get]

*Возвращает заголовок.*

### Подробное описание

Класс аргументов события уведомления.

Наследует EventArgs.

См. определение в файле **NotificationEventArgs.cs** строка **9**

### Конструктор(ы)

#### ViewModel.Implementations.Mocks.NotificationEventArgs.NotificationEventArgs (string description, string title)

Создаёт экземпяр класса **NotificationEventArgs**.

##### Аргументы

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

См. определение в файле **NotificationEventArgs.cs** строка **26**

### Полный список свойств

#### string ViewModel.Implementations.Mocks.NotificationEventArgs.Description[get]

Возвращает описание.

См. определение в файле **NotificationEventArgs.cs** строка **14**

#### string ViewModel.Implementations.Mocks.NotificationEventArgs.Title[get]

Возвращает заголовок.

См. определение в файле **NotificationEventArgs.cs** строка **19**

## Класс MachineLearning.DataProcessors.OutlierRowProcessor

Класс обработчика строк для устранения выбросов.

Граф наследования:MachineLearning.DataProcessors.OutlierRowProcessor:

### Открытые члены

* DataProcessorResult< IEnumerable< double > > **Process** (IEnumerable< IEnumerable< double > > data)

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IDataProcessor< T, R >

* DataProcessorResult< R > **Process** (IEnumerable< T > data)

*Преобразует данные.*

### Подробное описание

Класс обработчика строк для устранения выбросов.

Реализует **IPointDataProcessor**.

См. определение в файле **OutlierRowProcessor.cs** строка **14**

### Методы

#### DataProcessorResult< IEnumerable< double > > MachineLearning.DataProcessors.OutlierRowProcessor.Process (IEnumerable< IEnumerable< double > > data)

См. определение в файле **OutlierRowProcessor.cs** строка **22**

## Класс View.DataTemplates.PageDataTemplate

Класс шаблона данных для страниц.

Граф наследования:View.DataTemplates.PageDataTemplate:

### Открытые члены

* bool **Match** (object? data)
* Control? **Build** (object? data)
* Control? **Build** (object? data, Control? existing)

### Свойства

* object? **Content** [get, set]

*Возвращает и задаёт контент.*

### Подробное описание

Класс шаблона данных для страниц.

Наследует IRecyclingDataTemplate.

См. определение в файле **PageDataTemplate.cs** строка **18**

### Методы

#### Control? View.DataTemplates.PageDataTemplate.Build (object? data, Control? existing)

См. определение в файле **PageDataTemplate.cs** строка **34**

### Полный список свойств

#### object? View.DataTemplates.PageDataTemplate.Content[get], [set]

Возвращает и задаёт контент.

См. определение в файле **PageDataTemplate.cs** строка **25**

## Класс View.Technilcals.PageMetadata

Класс метаданных страницы.

### Свойства

* string? **Title** [get, set]

*Возвращает и задаёт название.*

* object? **Icon** [get, set]

*Возвращает и задаёт иконку.*

### Подробное описание

Класс метаданных страницы.

См. определение в файле **PageMetadata.cs** строка **6**

### Полный список свойств

#### object? View.Technilcals.PageMetadata.Icon[get], [set]

Возвращает и задаёт иконку.

См. определение в файле **PageMetadata.cs** строка **16**

#### string? View.Technilcals.PageMetadata.Title[get], [set]

Возвращает и задаёт название.

См. определение в файле **PageMetadata.cs** строка **11**

## Класс ViewModel.Implementations.ModelLearning.PlannedRealTaskElementEvaluatorLearningController

Класс контроллера обучения модели обучения с учителем для предсказания запланированого реального показателя элементарной задачи.

Граф наследования:ViewModel.Implementations.ModelLearning.PlannedRealTaskElementEvaluatorLearningController:

### Открытые члены

* **PlannedRealTaskElementEvaluatorLearningController** (**PlannedRealTaskElementLearningConverter** converter, **IRegressionModel** model, **IRegressionEvaluator** evaluator)

*Создаёт экземпляр классса* ***PlannedRealTaskElementEvaluatorLearningController****.*

#### Открытые члены унаследованные от ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >

* **BaseSupervisedEvaluatorLearningController** (ISupervisedLearningConverter< T, R, D, DT, DR > converter, ISupervisedLearningModel< T, R > model, ISupervisedLearningEvaluator< T, R > evaluator)

*Создаёт экземпляр классса* ***BaseSupervisedEvaluatorLearningController<T, R, D, DT, DR>****.*

* async Task< bool > **Train** (IEnumerable< D > data)

*Выполняет обучение модели на основе данных.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает задачу процесса обучения с результатом true , если обучение прошло успешно, иначе false .

* DR **Predict** (DT data)

*Выполняет предсказание на основе данных.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает предсказанное значение.

### Дополнительные унаследованные члены

#### Свойства унаследованные от ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >

* **ScoreMetricCategory** **MinScoreCategory** [get, set]

*Возвращает и задаёт минимальную категорию метрики оценки.*

* bool **IsValidModel** [get, protected set]

*Возвращает логическое значение, указывающее на достоверность модели обучения.*

### Подробное описание

Класс контроллера обучения модели обучения с учителем для предсказания запланированого реального показателя элементарной задачи.

Наследует **BaseSupervisedEvaluatorLearningController<IEnumerable<double>, double, ITaskElement, ITaskElement, double>**.

См. определение в файле **PlannedRealTaskElementEvaluatorLearningController.cs** строка **17**

### Конструктор(ы)

#### ViewModel.Implementations.ModelLearning.PlannedRealTaskElementEvaluatorLearningController.PlannedRealTaskElementEvaluatorLearningController (PlannedRealTaskElementLearningConverter converter, IRegressionModel model, IRegressionEvaluator evaluator)

Создаёт экземпляр классса **PlannedRealTaskElementEvaluatorLearningController**.

##### Аргументы

|  |  |
| --- | --- |
| *converter* | Конвертор данных в данные для предсказания с учителем и наоборот. |
| *model* | Модель обучения c учителем. |
| *evaluator* | Оценка модели обучения с учителем. |

См. определение в файле **PlannedRealTaskElementEvaluatorLearningController.cs** строка **30**

## Класс ViewModel.Implementations.ModelLearning.Converters.PlannedRealTaskElementLearningConverter

Класс конвертора элементраных задач в данные для предсказания запланированного реального показателя с учителем и наоборот.

Граф наследования:ViewModel.Implementations.ModelLearning.Converters.PlannedRealTaskElementLearningConverter:

### Открытые члены

* **PlannedRealTaskElementLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors, IFactory< **IScaler** > scalerFactory, IDataTransformer< **TaskMetadata**, int?> metadataCategoriesTransformer, IDataTransformer< **TaskMetadata**, IEnumerable< int > > metadataTagsITransformer)

*Создаёт экземпляр класса* ***PlannedRealTaskElementLearningConverter****.*

* override double **ConvertPredicted** (double predicted)

#### Открытые члены унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* LearningModelData< IEnumerable< double >, R > **FitConvertData** (IEnumerable< D > data)

*Адаптирует конвертацию данных под данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные данные для предсказания.

#### Открытые члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

##### *Аргументы*

|  |  |
| --- | --- |
| *predicted* | Предсказанное значение. |

##### *Возвращает*

Возвращает конвертированное выходное значение.

* IEnumerable< double > **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные входные данные для предсказания.

### Защищенные члены

* override List< double?> **ExtractPrimaryFeatures** (**ITaskElement** dataItem)
* override double **ProcessTarget** (**ITaskElement** item)

#### Защищенные члены унаследованные от ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >

* **BaseTaskElementSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors, IFactory< **IScaler** > scalerFactory, IDataTransformer< **TaskMetadata**, int?> metadataCategoriesTransformer, IDataTransformer< **TaskMetadata**, IEnumerable< int > > metadataTagsITransformer)

*Создаёт экземпляр класса* ***BaseTaskElementSupervisedLearningConverter<R, DR>****.*

* override IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< **ITaskElement** > data)
* override IEnumerable< double?> **ExtractFeatures** (**ITaskElement** dataItem)
* override **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)
* List< double?> **ExtractPrimaryFeatures** (**ITaskElement** dataItem)

*Извлекает первичные признаки у элемента данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* **BaseSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors)

*Создаёт экземпляр класса* ***BaseSupervisedLearningConverter<R, D, DT, DR>****.*

* R **ProcessTarget** (D item)

*Обрабатывает целевые значения элемента.*

* IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)

*Создаёт масштабирование данных.*

* DataProcessorResult< IEnumerable< double > > **ProcessFeaturesData** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* void **CreateScalers** (double[][] featuresArray)

*Создаёт коллекцию масштабирования данных.*

* IEnumerable< int > **NormalizeRemovedIndices** (IEnumerable< IEnumerable< int > > removedIndicesGroups)

*Нормализует индексы удалённых индексов.*

* IEnumerable< R > **ProcessTargets** (IEnumerable< D > data, IEnumerable< int > removedRowsIndices)

*Обрабатывает целевые значения данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* **BaseLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor)

*Создаёт экземпляр класса* ***BaseLearningConverter<R, D, DT, DR>****.*

* IEnumerable< double?> **ExtractFeatures** (DT dataItem)

*Извлекает признаки у элемента данных.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >

* readonly IFactory< **IScaler** > **\_scalerFactory**

*Фабрика, создающая масштабирования данных.*

* readonly IDataTransformer< **TaskMetadata**, int?> **\_metadataCategoriesTransformer**

*Преобразование категории метаданных в данные для предсказания.*

* readonly IDataTransformer< **TaskMetadata**, IEnumerable< int > > **\_metadataTagsTransformer**

*Преобразование теги метаданных в данные для предсказания.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* readonly IEnumerable< **IPointDataProcessor** > **\_pointDataProcessors**

*Обработчики точечных данных.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* readonly **IPrimaryPointDataProcessor** **\_primaryPointDataProcessor**

*Первичный обработчик точечных данных.*

* IEnumerable< int >? **\_removedColumnsIndices**

*Индексы удалённых столбцов.*

* IEnumerable< **IScaler** >? **\_scalers**

*Коллекция масштабирования данных.*

### Подробное описание

Класс конвертора элементраных задач в данные для предсказания запланированного реального показателя с учителем и наоборот.

Наследует **BaseTaskElementSupervisedLearningConverter<double, double>**.

См. определение в файле **PlannedRealTaskElementLearningConverter.cs** строка **18**

### Конструктор(ы)

#### ViewModel.Implementations.ModelLearning.Converters.PlannedRealTaskElementLearningConverter.PlannedRealTaskElementLearningConverter (IPrimaryPointDataProcessor primaryPointDataProcessor, IEnumerable< IPointDataProcessor > pointDataProcessors, IFactory< IScaler > scalerFactory, IDataTransformer< TaskMetadata, int?> metadataCategoriesTransformer, IDataTransformer< TaskMetadata, IEnumerable< int > > metadataTagsITransformer)

Создаёт экземпляр класса **PlannedRealTaskElementLearningConverter**.

##### Аргументы

|  |  |
| --- | --- |
| *primaryPointDataProcessor* | Первичный обработчик точечных данных. |
| *pointDataProcessors* | Обработчики точечных данных. |
| *scalerFactory* | Фабрика, создающая масштабирования данных. |
| *metadataCategoriesTransformer* | Преобразование категории метаданных в данные для предсказания. |
| *metadataTagsITransformer* | Преобразование теги метаданных в данные для предсказания. |

См. определение в файле **PlannedRealTaskElementLearningConverter.cs** строка **33**

## Класс ViewModel.Implementations.ModelLearning.PlannedTimeTaskElementEvaluatorLearningController

Класс контроллера обучения модели обучения с учителем для предсказания запланированого времени элементарной задачи.

Граф наследования:ViewModel.Implementations.ModelLearning.PlannedTimeTaskElementEvaluatorLearningController:

### Открытые члены

* **PlannedTimeTaskElementEvaluatorLearningController** (**PlannedTimeTaskElementLearningConverter** converter, **IRegressionModel** model, **IRegressionEvaluator** evaluator)

*Создаёт экземпляр классса* ***PlannedTimeTaskElementEvaluatorLearningController****.*

#### Открытые члены унаследованные от ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >

* **BaseSupervisedEvaluatorLearningController** (ISupervisedLearningConverter< T, R, D, DT, DR > converter, ISupervisedLearningModel< T, R > model, ISupervisedLearningEvaluator< T, R > evaluator)

*Создаёт экземпляр классса* ***BaseSupervisedEvaluatorLearningController<T, R, D, DT, DR>****.*

* async Task< bool > **Train** (IEnumerable< D > data)

*Выполняет обучение модели на основе данных.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает задачу процесса обучения с результатом true , если обучение прошло успешно, иначе false .

* DR **Predict** (DT data)

*Выполняет предсказание на основе данных.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает предсказанное значение.

### Дополнительные унаследованные члены

#### Свойства унаследованные от ViewModel.Implementations.ModelLearning.BaseSupervisedEvaluatorLearningController< T, R, D, DT, DR >

* **ScoreMetricCategory** **MinScoreCategory** [get, set]

*Возвращает и задаёт минимальную категорию метрики оценки.*

* bool **IsValidModel** [get, protected set]

*Возвращает логическое значение, указывающее на достоверность модели обучения.*

### Подробное описание

Класс контроллера обучения модели обучения с учителем для предсказания запланированого времени элементарной задачи.

Наследует **BaseSupervisedEvaluatorLearningController<IEnumerable<double>, double, ITaskElement, ITaskElement, TimeSpan>**.

См. определение в файле **PlannedTimeTaskElementEvaluatorLearningController.cs** строка **17**

### Конструктор(ы)

#### ViewModel.Implementations.ModelLearning.PlannedTimeTaskElementEvaluatorLearningController.PlannedTimeTaskElementEvaluatorLearningController (PlannedTimeTaskElementLearningConverter converter, IRegressionModel model, IRegressionEvaluator evaluator)

Создаёт экземпляр классса **PlannedTimeTaskElementEvaluatorLearningController**.

##### Аргументы

|  |  |
| --- | --- |
| *converter* | Конвертор данных в данные для предсказания с учителем и наоборот. |
| *model* | Модель обучения c учителем. |
| *evaluator* | Оценка модели обучения с учителем. |

См. определение в файле **PlannedTimeTaskElementEvaluatorLearningController.cs** строка **30**

## Класс ViewModel.Implementations.ModelLearning.Converters.PlannedTimeTaskElementLearningConverter

Класс конвертора элементраных задач в данные для предсказания запланированного времени с учителем и наоборот.

Граф наследования:ViewModel.Implementations.ModelLearning.Converters.PlannedTimeTaskElementLearningConverter:

### Открытые члены

* **PlannedTimeTaskElementLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors, IFactory< **IScaler** > scalerFactory, IDataTransformer< **TaskMetadata**, int?> metadataCategoriesTransformer, IDataTransformer< **TaskMetadata**, IEnumerable< int > > metadataTagsITransformer)

*Создаёт экземпляр класса* ***PlannedTimeTaskElementLearningConverter****.*

* override TimeSpan **ConvertPredicted** (double predicted)

#### Открытые члены унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* LearningModelData< IEnumerable< double >, R > **FitConvertData** (IEnumerable< D > data)

*Адаптирует конвертацию данных под данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные данные для предсказания.

#### Открытые члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* DR **ConvertPredicted** (R predicted)

*Конвертирует предсказанное значение в выходное значение.*

##### *Аргументы*

|  |  |
| --- | --- |
| *predicted* | Предсказанное значение. |

##### *Возвращает*

Возвращает конвертированное выходное значение.

* IEnumerable< double > **ConvertData** (DT data)

*Конвертирует входные данные под входные данные для предсказания.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |

##### *Возвращает*

Возвращает конвертированные входные данные для предсказания.

### Защищенные члены

* override List< double?> **ExtractPrimaryFeatures** (**ITaskElement** dataItem)
* override double **ProcessTarget** (**ITaskElement** item)

#### Защищенные члены унаследованные от ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >

* **BaseTaskElementSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors, IFactory< **IScaler** > scalerFactory, IDataTransformer< **TaskMetadata**, int?> metadataCategoriesTransformer, IDataTransformer< **TaskMetadata**, IEnumerable< int > > metadataTagsITransformer)

*Создаёт экземпляр класса* ***BaseTaskElementSupervisedLearningConverter<R, DR>****.*

* override IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< **ITaskElement** > data)
* override IEnumerable< double?> **ExtractFeatures** (**ITaskElement** dataItem)
* override **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)
* List< double?> **ExtractPrimaryFeatures** (**ITaskElement** dataItem)

*Извлекает первичные признаки у элемента данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* **BaseSupervisedLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor, IEnumerable< **IPointDataProcessor** > pointDataProcessors)

*Создаёт экземпляр класса* ***BaseSupervisedLearningConverter<R, D, DT, DR>****.*

* R **ProcessTarget** (D item)

*Обрабатывает целевые значения элемента.*

* IEnumerable< IEnumerable< double?> > **ProcessFeatures** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* **IScaler** **CreateScaler** (int index, IEnumerable< int > removedColumnsIndices)

*Создаёт масштабирование данных.*

* DataProcessorResult< IEnumerable< double > > **ProcessFeaturesData** (IEnumerable< D > data)

*Обрабатывает признаков данных.*

* void **CreateScalers** (double[][] featuresArray)

*Создаёт коллекцию масштабирования данных.*

* IEnumerable< int > **NormalizeRemovedIndices** (IEnumerable< IEnumerable< int > > removedIndicesGroups)

*Нормализует индексы удалённых индексов.*

* IEnumerable< R > **ProcessTargets** (IEnumerable< D > data, IEnumerable< int > removedRowsIndices)

*Обрабатывает целевые значения данных.*

#### Защищенные члены унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* **BaseLearningConverter** (**IPrimaryPointDataProcessor** primaryPointDataProcessor)

*Создаёт экземпляр класса* ***BaseLearningConverter<R, D, DT, DR>****.*

* IEnumerable< double?> **ExtractFeatures** (DT dataItem)

*Извлекает признаки у элемента данных.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.Implementations.ModelLearning.Converters.BaseTaskElementSupervisedLearningConverter< R, DR >

* readonly IFactory< **IScaler** > **\_scalerFactory**

*Фабрика, создающая масштабирования данных.*

* readonly IDataTransformer< **TaskMetadata**, int?> **\_metadataCategoriesTransformer**

*Преобразование категории метаданных в данные для предсказания.*

* readonly IDataTransformer< **TaskMetadata**, IEnumerable< int > > **\_metadataTagsTransformer**

*Преобразование теги метаданных в данные для предсказания.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseSupervisedLearningConverter< R, D, DT, DR >

* readonly IEnumerable< **IPointDataProcessor** > **\_pointDataProcessors**

*Обработчики точечных данных.*

#### Защищенные данные унаследованные от MachineLearning.Converters.BaseLearningConverter< R, D, DT, DR >

* readonly **IPrimaryPointDataProcessor** **\_primaryPointDataProcessor**

*Первичный обработчик точечных данных.*

* IEnumerable< int >? **\_removedColumnsIndices**

*Индексы удалённых столбцов.*

* IEnumerable< **IScaler** >? **\_scalers**

*Коллекция масштабирования данных.*

### Подробное описание

Класс конвертора элементраных задач в данные для предсказания запланированного времени с учителем и наоборот.

Наследует **BaseTaskElementSupervisedLearningConverter<double, TimeSpan>**.

См. определение в файле **PlannedTimeTaskElementLearningConverter.cs** строка **18**

### Конструктор(ы)

#### ViewModel.Implementations.ModelLearning.Converters.PlannedTimeTaskElementLearningConverter.PlannedTimeTaskElementLearningConverter (IPrimaryPointDataProcessor primaryPointDataProcessor, IEnumerable< IPointDataProcessor > pointDataProcessors, IFactory< IScaler > scalerFactory, IDataTransformer< TaskMetadata, int?> metadataCategoriesTransformer, IDataTransformer< TaskMetadata, IEnumerable< int > > metadataTagsITransformer)

Создаёт экземпляр класса **PlannedTimeTaskElementLearningConverter**.

##### Аргументы

|  |  |
| --- | --- |
| *primaryPointDataProcessor* | Первичный обработчик точечных данных. |
| *pointDataProcessors* | Обработчики точечных данных. |
| *scalerFactory* | Фабрика, создающая масштабирования данных. |
| *metadataCategoriesTransformer* | Преобразование категории метаданных в данные для предсказания. |
| *metadataTagsITransformer* | Преобразование теги метаданных в данные для предсказания. |

См. определение в файле **PlannedTimeTaskElementLearningConverter.cs** строка **33**

## Класс View.Desktop.Linux.Program

Класс программы.

### Открытые статические члены

* static void **Main** (string[] args)

*Запускает основной поток приложения.*

* static AppBuilder **BuildAvaloniaApp** ()

*Настраивает приложение.*

### Подробное описание

Класс программы.

См. определение в файле **Program.cs** строка **13**

### Методы

#### static AppBuilder View.Desktop.Linux.Program.BuildAvaloniaApp ()[static]

Настраивает приложение.

##### Возвращает

Возвращает настройки приложения.

#### static void View.Desktop.Linux.Program.Main (string[] args)[static]

Запускает основной поток приложения.

##### Аргументы

|  |  |
| --- | --- |
| *args* | Аргументы. |

## Класс View.Desktop.macOS.Program

Класс программы.

### Открытые статические члены

* static void **Main** (string[] args)

*Запускает основной поток приложения.*

* static AppBuilder **BuildAvaloniaApp** ()

*Настраивает приложение.*

### Подробное описание

Класс программы.

См. определение в файле **Program.cs** строка **13**

### Методы

#### static AppBuilder View.Desktop.macOS.Program.BuildAvaloniaApp ()[static]

Настраивает приложение.

##### Возвращает

Возвращает настройки приложения.

#### static void View.Desktop.macOS.Program.Main (string[] args)[static]

Запускает основной поток приложения.

##### Аргументы

|  |  |
| --- | --- |
| *args* | Аргументы. |

## Класс View.Desktop.Windows.Program

Класс программы.

### Открытые статические члены

* static void **Main** (string[] args)

*Запускает основной поток приложения.*

* static AppBuilder **BuildAvaloniaApp** ()

*Настраивает приложение.*

### Подробное описание

Класс программы.

См. определение в файле **Program.cs** строка **13**

### Методы

#### static AppBuilder View.Desktop.Windows.Program.BuildAvaloniaApp ()[static]

Настраивает приложение.

##### Возвращает

Возвращает настройки приложения.

#### static void View.Desktop.Windows.Program.Main (string[] args)[static]

Запускает основной поток приложения.

##### Аргументы

|  |  |
| --- | --- |
| *args* | Аргументы. |

## Класс MachineLearning.LearningEvaluators.RegressionCrossValidationEvaluator

Класс оценки модели обучения регрессии методом кросс-валидации.

Граф наследования:MachineLearning.LearningEvaluators.RegressionCrossValidationEvaluator:

### Открытые члены

* **RegressionCrossValidationEvaluator** ()

*Создаёт экземпляр класса* ***RegressionCrossValidationEvaluator*** *по умолчанию.*

#### Открытые члены унаследованные от MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >

* async Task< **ScoreMetricCategory** > **Evaluate** (IEnumerable< T > data, IEnumerable< R > targets)

*Оценивает модель обучения.*

##### *Аргументы*

|  |  |
| --- | --- |
| *data* | Данные. |
| *targets* | Целевые значения. |

##### *Возвращает*

Возвращает категорию модели обучения.

#### Открытые члены унаследованные от MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

* **BaseCrossValidationLearningEvaluator** ()

*Создаёт экземпляр класса* ***BaseCrossValidationLearningEvaluator*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* override IEnumerable< **ValidationFold** > **GetValidationFolds** (IEnumerable< IEnumerable< double > > data, IEnumerable< double > targets)

#### Защищенные члены унаследованные от MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >

* IEnumerable< **ValidationFold** > **GetValidationFolds** (IEnumerable< T > data, IEnumerable< R > targets)

*Возвращает сегменты валидации.*

* virtual **ScoreMetricCategory** **GetScoresCategory** (IEnumerable< double > scores)

*Определяет категорию оценок.*

* override void **OnPropertyChanged< T >** (T oldValue, T newValue)

*Вызывается при изменении свойства.*

##### *Параметры шаблона*

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### *Аргументы*

|  |  |
| --- | --- |
| *oldValue* | Старое значение. |
| *newValue* | Новое значение. |

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Дополнительные унаследованные члены

#### Свойства унаследованные от MachineLearning.LearningEvaluators.BaseSupervisedCrossValidationEvaluator< T, R >

* ISupervisedLearningModel< T, R > **Model** [get, set]

*Возвращает и задаёт модель обучения с учителем.*

* ISupervisedScoreMetric< R > **ScoreMetric** [get, set]

*Возвращает и задаёт модель обучения с учителем.*

#### Свойства унаследованные от MachineLearning.LearningEvaluators.BaseCrossValidationLearningEvaluator

* int **NumberOfFolds** [get, set]

*Возвращает и задаёт количество сегментов.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс оценки модели обучения регрессии методом кросс-валидации.

Наследует **BaseSupervisedCrossValidationEvaluator<IEnumerable<double>, double>**. Реализует **IRegressionEvaluator**.

См. определение в файле **RegressionCrossValidationEvaluator.cs** строка **13**

### Конструктор(ы)

#### MachineLearning.LearningEvaluators.RegressionCrossValidationEvaluator.RegressionCrossValidationEvaluator ()

Создаёт экземпляр класса **RegressionCrossValidationEvaluator** по умолчанию.

См. определение в файле **RegressionCrossValidationEvaluator.cs** строка **19**

### Методы

#### override IEnumerable< ValidationFold > MachineLearning.LearningEvaluators.RegressionCrossValidationEvaluator.GetValidationFolds (IEnumerable< IEnumerable< double > > data, IEnumerable< double > targets)[protected]

См. определение в файле **RegressionCrossValidationEvaluator.cs** строка **25**

## Класс View.Views.Modals.RemoveTasksView

Класс пользовательского элемента диалога удаления задач.

Граф наследования:View.Views.Modals.RemoveTasksView:

### Открытые члены

* **RemoveTasksView** ()

*Создаёт экземпляр класса* ***RemoveTasksView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента диалога удаления задач.

Наследует ReactiveUserControl<RemoveTasksViewModel>.

См. определение в файле **RemoveTasksView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Modals.RemoveTasksView.RemoveTasksView ()

Создаёт экземпляр класса **RemoveTasksView** по умолчанию.

См. определение в файле **RemoveTasksView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Modals.RemoveTasksViewModel

Класс диалога удаления задач.

Граф наследования:ViewModel.ViewModels.Modals.RemoveTasksViewModel:

### Защищенные члены

* override void **GetArgs** (IList< **ITask** > args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* void **GetArgs** (A args)

*Получает аргументы, переданные в диалог.*

### Дополнительные унаследованные члены

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* async Task< R > **Invoke** (**BaseViewModel** parent, A args)

*Вызывает диалог.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* TaskCompletionSource< R >? **\_taskSource**

*Источник для завершения задачи, который используется для возврата результата диалога.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* **BaseViewModel**? **Parent** [get]

*Возвращает родителя.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс диалога удаления задач.

Наследует **BaseDialogViewModel<IList<ITask>, bool>**.

См. определение в файле **RemoveTasksViewModel.cs** строка **13**

## Класс MachineLearning.ScoreMetrics.RSquaredScoreMetric

РљР»Р°СЃСЃ РјРµС‚СЂРёРєРё RВІ РѕС†РµРЅРєРё РґР»СЏ РјРѕРґРµР»Рё РѕР±СѓС‡РµРЅРёСЏ СЂРµРіСЂРµСЃСЃРёРё.

Граф наследования:MachineLearning.ScoreMetrics.RSquaredScoreMetric:

### Открытые члены

* double **CalculateScore** (IEnumerable< double > actual, IEnumerable< double > predicted)
* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

##### *Аргументы*

|  |  |
| --- | --- |
| *score* | Оценка. |

##### *Возвращает*

Возвращает категорию оценки.

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T >

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< T > predicted)

*Вычисляет оценку на основе фактических и предсказанных данных.*

### Подробное описание

РљР»Р°СЃСЃ РјРµС‚СЂРёРєРё RВІ РѕС†РµРЅРєРё РґР»СЏ РјРѕРґРµР»Рё РѕР±СѓС‡РµРЅРёСЏ СЂРµРіСЂРµСЃСЃРёРё.

Реализует **IRegressionScoreMetric**.

См. определение в файле **RSquaredScoreMetric.cs** строка **13**

### Методы

#### double MachineLearning.ScoreMetrics.RSquaredScoreMetric.CalculateScore (IEnumerable< double > actual, IEnumerable< double > predicted)

См. определение в файле **RSquaredScoreMetric.cs** строка **16**

#### ScoreMetricCategory MachineLearning.ScoreMetrics.RSquaredScoreMetric.GetScoreCategory (double score)

Определяет категорию оценки.

##### Аргументы

|  |  |
| --- | --- |
| *score* | Оценка. |

##### Возвращает

Возвращает категорию оценки.

Замещает **MachineLearning.Interfaces.Generals.IScoreMetric** (*стр.275*).

## Класс ViewModel.Implementations.ModelLearning.Converters.ScalerFactory

Класс фабрики, создающая масштабирования данных.

Граф наследования:ViewModel.Implementations.ModelLearning.Converters.ScalerFactory:

### Открытые члены

* **IScaler** **Create** ()

*Создает новый экземпляр объекта.*

##### *Возвращает*

Возвращает новый экземпляр объекта.

### Подробное описание

Класс фабрики, создающая масштабирования данных.

Реализует **IFactory<IScaler>**.

См. определение в файле **ScalerFactory.cs** строка **14**

### Методы

#### IScaler ViewModel.Implementations.ModelLearning.Converters.ScalerFactory.Create ()

Создает новый экземпляр объекта.

##### Возвращает

Возвращает новый экземпляр объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >** (*стр.245*).

## Класс View.Views.Pages.SettingsView

Класс пользовательского элемента страницы настроек.

Граф наследования:View.Views.Pages.SettingsView:

### Открытые члены

* **SettingsView** ()

*Создаёт экземпляр класса* ***SettingsView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента страницы настроек.

Наследует ReactiveUserControl<SettingsViewModel>.

См. определение в файле **SettingsView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Pages.SettingsView.SettingsView ()

Создаёт экземпляр класса **SettingsView** по умолчанию.

См. определение в файле **SettingsView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Pages.SettingsViewModel

Класс контроллера страницы настроек.

Граф наследования:ViewModel.ViewModels.Pages.SettingsViewModel:

### Открытые члены

* **SettingsViewModel** (**ISettings** settings, **IResourceService** resourceService)

*Создаёт экземпляр класса* ***SettingsViewModel****.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Открытые атрибуты

* object **\_configuration**

*Конфигурация.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BasePageViewModel

* object **Metadata** [get, protected set]

*Возращает метаданные.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс контроллера страницы настроек.

Наследует **BaseViewModel**.

См. определение в файле **SettingsViewModel.cs** строка **15**

### Конструктор(ы)

#### ViewModel.ViewModels.Pages.SettingsViewModel.SettingsViewModel (ISettings settings, IResourceService resourceService)

Создаёт экземпляр класса **SettingsViewModel**.

##### Аргументы

|  |  |
| --- | --- |
| *settings* | Настройки. |
| *resourceService* | Сервис ресурсов. |

См. определение в файле **SettingsViewModel.cs** строка **33**

### Данные класса

#### object ViewModel.ViewModels.Pages.SettingsViewModel.\_configuration

Конфигурация.

См. определение в файле **SettingsViewModel.cs** строка **26**

## Класс MachineLearning.ScoreMetrics.SilhouetteScoreMetric

Класс метрики оценки Силуэта для модели обучения кластеризации.

Граф наследования:MachineLearning.ScoreMetrics.SilhouetteScoreMetric:

### Открытые члены

* double **CalculateScore** (IEnumerable< int > actual, IEnumerable< IEnumerable< double > > data)
* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

##### *Аргументы*

|  |  |
| --- | --- |
| *score* | Оценка. |

##### *Возвращает*

Возвращает категорию оценки.

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.IUnsupervisedScoreMetric< T, D >

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< D > data)

*Вычисляет оценку на основе фактических данных и данных для предсказания.*

### Свойства

* **IPointDistanceMetric** **PointDistanceMetric** [get, set]

*Возвращает и задаёт метрику дистанцию.*

### Подробное описание

Класс метрики оценки Силуэта для модели обучения кластеризации.

Реализует **IClusteringScoreMetric**.

См. определение в файле **SilhouetteScoreMetric.cs** строка **12**

### Методы

#### double MachineLearning.ScoreMetrics.SilhouetteScoreMetric.CalculateScore (IEnumerable< int > actual, IEnumerable< IEnumerable< double > > data)

См. определение в файле **SilhouetteScoreMetric.cs** строка **21**

#### ScoreMetricCategory MachineLearning.ScoreMetrics.SilhouetteScoreMetric.GetScoreCategory (double score)

Определяет категорию оценки.

##### Аргументы

|  |  |
| --- | --- |
| *score* | Оценка. |

##### Возвращает

Возвращает категорию оценки.

Замещает **MachineLearning.Interfaces.Generals.IScoreMetric** (*стр.275*).

### Полный список свойств

#### IPointDistanceMetric MachineLearning.ScoreMetrics.SilhouetteScoreMetric.PointDistanceMetric[get], [set]

**Инициализатор**

=

new EuclideanDistanceMetric()

Возвращает и задаёт метрику дистанцию.

См. определение в файле **SilhouetteScoreMetric.cs** строка **17**

## Класс MachineLearning.ScoreMetrics.SmapeScoreMetric

Класс метрики оценки SMAPE для модели обучения регрессии.

Граф наследования:MachineLearning.ScoreMetrics.SmapeScoreMetric:

### Открытые члены

* double **CalculateScore** (IEnumerable< double > actual, IEnumerable< double > predicted)
* **ScoreMetricCategory** **GetScoreCategory** (double score)

*Определяет категорию оценки.*

##### *Аргументы*

|  |  |
| --- | --- |
| *score* | Оценка. |

##### *Возвращает*

Возвращает категорию оценки.

#### Открытые члены унаследованные от MachineLearning.Interfaces.Generals.ISupervisedScoreMetric< T >

* double **CalculateScore** (IEnumerable< T > actual, IEnumerable< T > predicted)

*Вычисляет оценку на основе фактических и предсказанных данных.*

### Подробное описание

Класс метрики оценки SMAPE для модели обучения регрессии.

Реализует **IRegressionScoreMetric**.

См. определение в файле **SmapeScoreMetric.cs** строка **11**

### Методы

#### double MachineLearning.ScoreMetrics.SmapeScoreMetric.CalculateScore (IEnumerable< double > actual, IEnumerable< double > predicted)

См. определение в файле **SmapeScoreMetric.cs** строка **14**

#### ScoreMetricCategory MachineLearning.ScoreMetrics.SmapeScoreMetric.GetScoreCategory (double score)

Определяет категорию оценки.

##### Аргументы

|  |  |
| --- | --- |
| *score* | Оценка. |

##### Возвращает

Возвращает категорию оценки.

Замещает **MachineLearning.Interfaces.Generals.IScoreMetric** (*стр.275*).

## Класс ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.SqliteDbContext

Класс контекста базы данных SQLite.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.SqliteDbContext:

### Открытые члены

* **SqliteDbContext** (string connectionString)

*Создаёт экземпляр класса* ***SqliteDbContext****.*

#### Открытые члены унаследованные от ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext

* **BaseDbContext** (string connectionString)

*Создаёт экземпляр класса* ***BaseDbContext****.*

### Защищенные члены

* override void **OnConfiguring** (DbContextOptionsBuilder optionsBuilder)

#### Защищенные члены унаследованные от ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext

* override void **OnModelCreating** (ModelBuilder modelBuilder)
* override void **OnConfiguring** (DbContextOptionsBuilder optionsBuilder)

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext

* string **\_connectionString**

*Строка подключения.*

#### Свойства унаследованные от ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.BaseDbContext

* DbSet< **TaskEntity** > **Tasks** [get, set]

*Возвращает и задаёт сущности задачи.*

* DbSet< **MetadataEntity** > **Metadata** [get, set]

*Возвращает и задаёт сущности метаданных.*

* DbSet< **TagEntity** > **Tags** [get, set]

*Возвращает и задаёт сущности тега.*

* DbSet< **TaskCompositeEntity** > **TaskComposites** [get, set]

*Возвращает и задаёт сущности составной задачи.*

* DbSet< **TaskElementEntity** > **TaskElements** [get, set]

*Возвращает и задаёт сущности элементарной задачи.*

* DbSet< **TimeIntervalEntity** > **TimeIntervals** [get, set]

*Возвращает и задаёт сущности временного интерала.*

### Подробное описание

Класс контекста базы данных SQLite.

Наследует **BaseDbContext**.

См. определение в файле **SqliteDbContext.cs** строка **11**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.SqliteDbContext.SqliteDbContext (string connectionString)

Создаёт экземпляр класса **SqliteDbContext**.

##### Аргументы

|  |  |
| --- | --- |
| *connectionString* | Строка подключения. |

См. определение в файле **SqliteDbContext.cs** строка **17**

### Методы

#### override void ViewModel.Implementations.AppStates.Sessions.Database.DbContexts.SqliteDbContext.OnConfiguring (DbContextOptionsBuilder optionsBuilder)[protected]

См. определение в файле **SqliteDbContext.cs** строка **20**

## Класс ViewModel.Technicals.StatisticElement

Класс элемента статистики.

### Открытые члены

* **StatisticElement** (double value, string name)

*Создаёт экземпляр класса* ***StatisticElement****.*

### Свойства

* double **Value** [get]

*Возвращает значение.*

* string **Name** [get]

*Возвращает название.*

### Подробное описание

Класс элемента статистики.

См. определение в файле **StatisticElement.cs** строка **6**

### Конструктор(ы)

#### ViewModel.Technicals.StatisticElement.StatisticElement (double value, string name)

Создаёт экземпляр класса **StatisticElement**.

##### Аргументы

|  |  |
| --- | --- |
| *value* | Значение. |
| *name* | Название. |

См. определение в файле **StatisticElement.cs** строка **23**

### Полный список свойств

#### string ViewModel.Technicals.StatisticElement.Name[get]

Возвращает название.

См. определение в файле **StatisticElement.cs** строка **16**

#### double ViewModel.Technicals.StatisticElement.Value[get]

Возвращает значение.

См. определение в файле **StatisticElement.cs** строка **11**

## Класс View.Converters.StatisticElementsToColumnSeriesConverter

Класс конвертора элементов статистики в столбцы столбчатой диаграммы.

Граф наследования:View.Converters.StatisticElementsToColumnSeriesConverter:

### Открытые члены

* object **Convert** (object? value, Type targetType, object? parameter, CultureInfo culture)
* object **ConvertBack** (object? value, Type targetType, object? parameter, CultureInfo culture)

### Подробное описание

Класс конвертора элементов статистики в столбцы столбчатой диаграммы.

Реализует IValueConverter.

См. определение в файле **StatisticElementsToColumnSeriesConverter.cs** строка **19**

### Методы

#### object View.Converters.StatisticElementsToColumnSeriesConverter.Convert (object? value, Type targetType, object? parameter, CultureInfo culture)

См. определение в файле **StatisticElementsToColumnSeriesConverter.cs** строка **22**

## Класс View.Converters.StatisticElementsToPieSeriesConverter

Класс конвертора элементов статистики в сектора круговой диаграммы.

Граф наследования:View.Converters.StatisticElementsToPieSeriesConverter:

### Открытые члены

* object **Convert** (object? value, Type targetType, object? parameter, CultureInfo culture)
* object **ConvertBack** (object? value, Type targetType, object? parameter, CultureInfo culture)

### Подробное описание

Класс конвертора элементов статистики в сектора круговой диаграммы.

Реализует IValueConverter.

См. определение в файле **StatisticElementsToPieSeriesConverter.cs** строка **19**

### Методы

#### object View.Converters.StatisticElementsToPieSeriesConverter.Convert (object? value, Type targetType, object? parameter, CultureInfo culture)

См. определение в файле **StatisticElementsToPieSeriesConverter.cs** строка **22**

## Класс View.Views.Pages.StatisticView

Класс пользовательского элемента страницы статистики.

Граф наследования:View.Views.Pages.StatisticView:

### Открытые члены

* **StatisticView** ()

*Создаёт экземпляр класса* ***StatisticView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента страницы статистики.

Наследует ReactiveUserControl<StatisticViewModel>.

См. определение в файле **StatisticView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Pages.StatisticView.StatisticView ()

Создаёт экземпляр класса **StatisticView** по умолчанию.

См. определение в файле **StatisticView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Pages.StatisticViewModel

Класс контроллера страницы статистики.

Граф наследования:ViewModel.ViewModels.Pages.StatisticViewModel:

### Открытые члены

* **StatisticViewModel** (**ISession** session, **IResourceService** resourceService)

*Создаёт экземпляр класса* ***StatisticViewModel****.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BasePageViewModel

* object **Metadata** [get, protected set]

*Возращает метаданные.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс контроллера страницы статистики.

Наследует **BasePageViewModel**.

См. определение в файле **StatisticViewModel.cs** строка **18**

### Конструктор(ы)

#### ViewModel.ViewModels.Pages.StatisticViewModel.StatisticViewModel (ISession session, IResourceService resourceService)

Создаёт экземпляр класса **StatisticViewModel**.

##### Аргументы

|  |  |
| --- | --- |
| *session* | Сессия. |
| *resourceService* | Сервис ресурсов. |

См. определение в файле **StatisticViewModel.cs** строка **89**

## Класс ViewModel.Implementations.Sessions.Database.Entities.TagEntity

Класс сущности тега.

### Свойства

* int **MetadataId** [get, set]

*Возвращает и задаёт индентифиактор метаданных.*

* string **Tag** [get, set]

*Возвращает и задаёт тег.*

* virtual **MetadataEntity** **Metadata** [get, set]

*Возвращает и задаёт метаданные.*

### Подробное описание

Класс сущности тега.

См. определение в файле **TagEntity.cs** строка **13**

### Полный список свойств

#### virtual MetadataEntity ViewModel.Implementations.Sessions.Database.Entities.TagEntity.Metadata[get], [set]

Возвращает и задаёт метаданные.

См. определение в файле **TagEntity.cs** строка **29**

#### int ViewModel.Implementations.Sessions.Database.Entities.TagEntity.MetadataId[get], [set]

Возвращает и задаёт индентифиактор метаданных.

См. определение в файле **TagEntity.cs** строка **18**

#### string ViewModel.Implementations.Sessions.Database.Entities.TagEntity.Tag[get], [set]

Возвращает и задаёт тег.

См. определение в файле **TagEntity.cs** строка **23**

## Класс View.Converters.TagsToStringConverter

Класс конвертора тегов в строку и наоборот.

Граф наследования:View.Converters.TagsToStringConverter:

### Открытые члены

* object **Convert** (object? value, Type targetType, object? parameter, CultureInfo culture)
* object **ConvertBack** (object? value, Type targetType, object? parameter, CultureInfo culture)

### Подробное описание

Класс конвертора тегов в строку и наоборот.

Реализует IValueConverter.

См. определение в файле **TagsToStringConverter.cs** строка **16**

### Методы

#### object View.Converters.TagsToStringConverter.Convert (object? value, Type targetType, object? parameter, CultureInfo culture)

См. определение в файле **TagsToStringConverter.cs** строка **21**

#### object View.Converters.TagsToStringConverter.ConvertBack (object? value, Type targetType, object? parameter, CultureInfo culture)

См. определение в файле **TagsToStringConverter.cs** строка **29**

## Класс Model.Tasks.TaskComposite

Класс составной задачи.

Граф наследования:Model.Tasks.TaskComposite:

### Открытые члены

* **TaskComposite** (IEnumerable< **ITask** >? subtasks=null)

*Создаёт экземпляр класса* ***TaskComposite****.*

* **TaskComposite** ()

*Создаёт экземпляр класса* ***TaskComposite*** *по умолчанию.*

* virtual object **Clone** ()

#### Открытые члены унаследованные от TrackableFeatures.TrackableCollection< T >

* **TrackableCollection** (IEnumerable< T >? items=null)
* bool **Contains** (T item)
* bool **Contains** (object? value)
* int **IndexOf** (T item)
* int **IndexOf** (object? value)
* void **CopyTo** (T[] array, int arrayIndex)
* void **CopyTo** (Array array, int index)
* void **Add** (T item)
* int **Add** (object? value)
* void **Insert** (int index, T item)
* void **Insert** (int index, object? value)
* bool **Remove** (T item)
* void **Remove** (object? value)
* void **RemoveAt** (int index)
* void **Replace** (int index, T item)

*Заменяет элемент по указанному индексу на новый.*

* void **Move** (int oldIndex, int newIndex)

*Перемещает элемент из одного индекса в другой.*

* void **Clear** ()
* IEnumerator< T > **GetEnumerator** ()

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* override void **OnAddedItem** (**ITask** task, bool arePropertiesUpdate=true)
* override void **OnRemovedItem** (**ITask** task)

#### Защищенные члены унаследованные от TrackableFeatures.TrackableCollection< T >

* int **AddItem** (T item)

*Добавляет элемент в коллекцию.*

* void **InsertItem** (int index, T item)

*Вставляет элемент в указанную позицию коллекции.*

* bool **RemoveItem** (T item)

*Удаляет элемент из коллекции.*

* virtual void **OnAddedItem** (T item, bool arePropertiesUpdate=true)

*Вызывается при добавлении элемента.*

* virtual void **OnRemovedItem** (T item)

*Вызывается при удалении элемента.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? oldItem, object? newItem, int index)

*Вызывает событие* ***CollectionChanged*** *для индекса, в котором изменился элемент.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? item, int index)

*Вызывает событие* ***CollectionChanged*** *для элемента, над которым совершили действие.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? item, int newIndex, int oldIndex)

*Вызывает событие* ***CollectionChanged*** *для элемента, который изменил индекс.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action)

*Вызывает событие* ***CollectionChanged*** *для действия.*

* void **OnCollectionChanged** (NotifyCollectionChangedEventArgs args)

*Вызывает событие* ***CollectionChanged*** *с аргументами.*

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Свойства

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

* int **Difficult** [get]

*Возращает сложность.*

* int **Priority** [get]

*Возращает приоритет.*

* **TaskStatus** **Status** [get]

*Возращает статус.*

* DateTime? **Deadline** [get]

*Возращает срок.*

* double **Progress** [get]

*Возращает прогресс.*

* TimeSpan **PlannedTime** [get]

*Возращает запланированное время.*

* TimeSpan **SpentTime** [get]

*Возращает потраченное время.*

#### Свойства унаследованные от TrackableFeatures.TrackableCollection< T >

* T **this[int index]** [get, set]
* int **Count** [get]
* bool **IsReadOnly** [get]
* bool **IsFixedSize** [get]
* bool **IsSynchronized** [get]
* object **SyncRoot** [get]

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### События унаследованные от TrackableFeatures.TrackableCollection< T >

* NotifyCollectionChangedEventHandler? **CollectionChanged**

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс составной задачи.

Наследует **TrackableCollection<ITask>**. Реализует **ITaskComposite** и ICloneable.

См. определение в файле **TaskComposite.cs** строка **16**

### Конструктор(ы)

#### Model.Tasks.TaskComposite.TaskComposite (IEnumerable< ITask >? subtasks = null)

Создаёт экземпляр класса **TaskComposite**.

##### Аргументы

|  |  |
| --- | --- |
| *subtasks* | Подзадачи. |

См. определение в файле **TaskComposite.cs** строка **76**

#### Model.Tasks.TaskComposite.TaskComposite ()

Создаёт экземпляр класса **TaskComposite** по умолчанию.

См. определение в файле **TaskComposite.cs** строка **81**

### Методы

#### virtual object Model.Tasks.TaskComposite.Clone ()[virtual]

Переопределяется в **ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain** (*стр.438*).

См. определение в файле **TaskComposite.cs** строка **84**

#### override void Model.Tasks.TaskComposite.OnAddedItem (ITask task, bool arePropertiesUpdate = true)[protected]

См. определение в файле **TaskComposite.cs** строка **103**

#### override void Model.Tasks.TaskComposite.OnRemovedItem (ITask task)[protected]

См. определение в файле **TaskComposite.cs** строка **120**

### Полный список свойств

#### DateTime? Model.Tasks.TaskComposite.Deadline[get]

Возращает срок.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TaskComposite.cs** строка **59**

#### int Model.Tasks.TaskComposite.Difficult[get]

Возращает сложность.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TaskComposite.cs** строка **50**

#### object? Model.Tasks.TaskComposite.Metadata[get], [set]

Возращает и задаёт метаданные.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TaskComposite.cs** строка **43**

#### ITaskComposite? Model.Tasks.TaskComposite.ParentTask[get], [set]

Возращает и задаёт родительскую задачу.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TaskComposite.cs** строка **36**

#### TimeSpan Model.Tasks.TaskComposite.PlannedTime[get]

Возращает запланированное время.

Замещает **Model.Interfaces.ITask** (*стр.297*).

См. определение в файле **TaskComposite.cs** строка **65**

#### int Model.Tasks.TaskComposite.Priority[get]

Возращает приоритет.

Замещает **Model.Interfaces.ITask** (*стр.297*).

См. определение в файле **TaskComposite.cs** строка **53**

#### double Model.Tasks.TaskComposite.Progress[get]

Возращает прогресс.

Замещает **Model.Interfaces.ITask** (*стр.297*).

См. определение в файле **TaskComposite.cs** строка **62**

#### TimeSpan Model.Tasks.TaskComposite.SpentTime[get]

Возращает потраченное время.

Замещает **Model.Interfaces.ITask** (*стр.297*).

См. определение в файле **TaskComposite.cs** строка **69**

#### TaskStatus Model.Tasks.TaskComposite.Status[get]

Возращает статус.

Замещает **Model.Interfaces.ITask** (*стр.297*).

См. определение в файле **TaskComposite.cs** строка **56**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain

Класс домменной модели составной задачи.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain:

### Открытые члены

* **TaskCompositeDomain** (IEnumerable< **ITask** >? subtasks=null)

*Создаёт экземпляр класса* ***TaskCompositeDomain****.*

* **TaskCompositeDomain** ()

*Создаёт экземпляр класса* ***TaskCompositeDomain*** *по умолчанию.*

* override object **Clone** ()

#### Открытые члены унаследованные от Model.Tasks.TaskComposite

* **TaskComposite** (IEnumerable< **ITask** >? subtasks=null)

*Создаёт экземпляр класса* ***TaskComposite****.*

* **TaskComposite** ()

*Создаёт экземпляр класса* ***TaskComposite*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableCollection< T >

* **TrackableCollection** (IEnumerable< T >? items=null)
* bool **Contains** (T item)
* bool **Contains** (object? value)
* int **IndexOf** (T item)
* int **IndexOf** (object? value)
* void **CopyTo** (T[] array, int arrayIndex)
* void **CopyTo** (Array array, int index)
* void **Add** (T item)
* int **Add** (object? value)
* void **Insert** (int index, T item)
* void **Insert** (int index, object? value)
* bool **Remove** (T item)
* void **Remove** (object? value)
* void **RemoveAt** (int index)
* void **Replace** (int index, T item)

*Заменяет элемент по указанному индексу на новый.*

* void **Move** (int oldIndex, int newIndex)

*Перемещает элемент из одного индекса в другой.*

* void **Clear** ()
* IEnumerator< T > **GetEnumerator** ()

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Свойства

* **TaskCompositeEntity** **Entity** [get, set]

*Возвращает и задаёт связанную сущность.*

* object **EntityId** [get]

*Возвращает индетификатор связанной сущности.*

#### Свойства унаследованные от Model.Tasks.TaskComposite

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

* int **Difficult** [get]

*Возращает сложность.*

* int **Priority** [get]

*Возращает приоритет.*

* **TaskStatus** **Status** [get]

*Возращает статус.*

* DateTime? **Deadline** [get]

*Возращает срок.*

* double **Progress** [get]

*Возращает прогресс.*

* TimeSpan **PlannedTime** [get]

*Возращает запланированное время.*

* TimeSpan **SpentTime** [get]

*Возращает потраченное время.*

#### Свойства унаследованные от TrackableFeatures.TrackableCollection< T >

* T **this[int index]** [get, set]
* int **Count** [get]
* bool **IsReadOnly** [get]
* bool **IsFixedSize** [get]
* bool **IsSynchronized** [get]
* object **SyncRoot** [get]

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от Model.Tasks.TaskComposite

* override void **OnAddedItem** (**ITask** task, bool arePropertiesUpdate=true)
* override void **OnRemovedItem** (**ITask** task)

#### Защищенные члены унаследованные от TrackableFeatures.TrackableCollection< T >

* int **AddItem** (T item)

*Добавляет элемент в коллекцию.*

* void **InsertItem** (int index, T item)

*Вставляет элемент в указанную позицию коллекции.*

* bool **RemoveItem** (T item)

*Удаляет элемент из коллекции.*

* virtual void **OnAddedItem** (T item, bool arePropertiesUpdate=true)

*Вызывается при добавлении элемента.*

* virtual void **OnRemovedItem** (T item)

*Вызывается при удалении элемента.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? oldItem, object? newItem, int index)

*Вызывает событие* ***CollectionChanged*** *для индекса, в котором изменился элемент.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? item, int index)

*Вызывает событие* ***CollectionChanged*** *для элемента, над которым совершили действие.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? item, int newIndex, int oldIndex)

*Вызывает событие* ***CollectionChanged*** *для элемента, который изменил индекс.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action)

*Вызывает событие* ***CollectionChanged*** *для действия.*

* void **OnCollectionChanged** (NotifyCollectionChangedEventArgs args)

*Вызывает событие* ***CollectionChanged*** *с аргументами.*

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableCollection< T >

* NotifyCollectionChangedEventHandler? **CollectionChanged**

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс домменной модели составной задачи.

Наследует **TaskComposite**. Реализует **IDomain**.

См. определение в файле **TaskCompositeDomain.cs** строка **15**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain.TaskCompositeDomain (IEnumerable< ITask >? subtasks = null)

Создаёт экземпляр класса **TaskCompositeDomain**.

##### Аргументы

|  |  |
| --- | --- |
| *subtasks* | Подзадачи. |

См. определение в файле **TaskCompositeDomain.cs** строка **29**

#### ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain.TaskCompositeDomain ()

Создаёт экземпляр класса **TaskCompositeDomain** по умолчанию.

См. определение в файле **TaskCompositeDomain.cs** строка **34**

### Методы

#### override object ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain.Clone ()[virtual]

Переопределяет метод предка **Model.Tasks.TaskComposite** (*стр.431*).

См. определение в файле **TaskCompositeDomain.cs** строка **37**

### Полный список свойств

#### TaskCompositeEntity ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain.Entity[get], [set]

Возвращает и задаёт связанную сущность.

См. определение в файле **TaskCompositeDomain.cs** строка **20**

#### object ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskCompositeDomain.EntityId[get]

Возвращает индетификатор связанной сущности.

Замещает **ViewModel.Implementations.AppStates.Sessions.Database.Domains.IDomain** (*стр.241*).

См. определение в файле **TaskCompositeDomain.cs** строка **23**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskCompositeEntity

Класс сущности составной задачи.

### Свойства

* int **Id** [get, set]

*Возвращает и задаёт индентификатор.*

* virtual **TaskEntity** **Task** [get, set]

*Возвращает и задаёт задачи.*

* virtual ICollection< **TaskEntity** > **Subtasks** = new List<**TaskEntity**>() [get, set]

*Возвращает и задаёт подзадачи.*

### Подробное описание

Класс сущности составной задачи.

См. определение в файле **TaskCompositeEntity.cs** строка **11**

### Полный список свойств

#### int ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskCompositeEntity.Id[get], [set]

Возвращает и задаёт индентификатор.

См. определение в файле **TaskCompositeEntity.cs** строка **16**

#### virtual ICollection<TaskEntity> ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskCompositeEntity.Subtasks = new List<TaskEntity>()[get], [set]

Возвращает и задаёт подзадачи.

См. определение в файле **TaskCompositeEntity.cs** строка **27**

#### virtual TaskEntity ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskCompositeEntity.Task[get], [set]

Возвращает и задаёт задачи.

См. определение в файле **TaskCompositeEntity.cs** строка **22**

## Класс ViewModel.Implementations.DataManagers.Factories.TaskCompositeFactory

Класс фабрики, создающая составные задачи.

Граф наследования:ViewModel.Implementations.DataManagers.Factories.TaskCompositeFactory:

### Открытые члены

* **TaskCompositeFactory** (IFactory< object > metadataFactory)

*Создаёт экземпляр класса* ***TaskCompositeFactory****.*

* **ITaskComposite** **Create** ()

*Создает новый экземпляр объекта.*

##### *Возвращает*

Возвращает новый экземпляр объекта.

### Подробное описание

Класс фабрики, создающая составные задачи.

Реализует **IFactory<ITaskComposite>**.

См. определение в файле **TaskCompositeFactory.cs** строка **14**

### Конструктор(ы)

#### ViewModel.Implementations.DataManagers.Factories.TaskCompositeFactory.TaskCompositeFactory (IFactory< object > metadataFactory)

Создаёт экземпляр класса **TaskCompositeFactory**.

##### Аргументы

|  |  |
| --- | --- |
| *metadataFactory* | Фабрика, создающая метаданные. |

См. определение в файле **TaskCompositeFactory.cs** строка **25**

### Методы

#### ITaskComposite ViewModel.Implementations.DataManagers.Factories.TaskCompositeFactory.Create ()

Создает новый экземпляр объекта.

##### Возвращает

Возвращает новый экземпляр объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >** (*стр.245*).

См. определение в файле **TaskCompositeFactory.cs** строка **32**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskCompositeMapper

Класс перобразования значений составных задач между двумя предметными областями.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskCompositeMapper:

### Открытые члены

* **TaskCompositeMapper** (IMapper< **MetadataEntity**, object > metadataMapper, IMapper< **TaskElementEntity**, **ITaskElement** > taskElementMapper)

*Создаёт экземпляр класса* ***TaskCompositeMapper****.*

* **ITaskComposite** **Map** (**TaskCompositeEntity** value)
* **TaskCompositeEntity** **MapBack** (**ITaskComposite** value)

#### Открытые члены унаследованные от ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 >

* T2 **Map** (T1 value)

*Преобразует исходные данные в целевые данные.*

* T1 **MapBack** (T2 value)

*Преобразует целевые данные в исходные данные.*

### Подробное описание

Класс перобразования значений составных задач между двумя предметными областями.

Реализует **IMapper<TimeIntervalEntity, ITimeIntervalElement>**.

См. определение в файле **TaskCompositeMapper.cs** строка **14**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskCompositeMapper.TaskCompositeMapper (IMapper< MetadataEntity, object > metadataMapper, IMapper< TaskElementEntity, ITaskElement > taskElementMapper)

Создаёт экземпляр класса **TaskCompositeMapper**.

##### Аргументы

|  |  |
| --- | --- |
| *metadataMapper* | Преобразование значений между сущностью метаданных и метаданными. |
| *taskElementMapper* | Преобразование значений между сущностью элементарных задач и элементарными задачами. |

См. определение в файле **TaskCompositeMapper.cs** строка **35**

### Методы

#### ITaskComposite ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskCompositeMapper.Map (TaskCompositeEntity value)

См. определение в файле **TaskCompositeMapper.cs** строка **43**

#### TaskCompositeEntity ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskCompositeMapper.MapBack (ITaskComposite value)

См. определение в файле **TaskCompositeMapper.cs** строка **71**

## Класс View.DataTemplates.TaskDataTemplate

Класс шаблона данных для задач.

Граф наследования:View.DataTemplates.TaskDataTemplate:

### Открытые члены

* Control? **Build** (object? param)
* InstancedBinding? **ItemsSelector** (object item)
* bool **Match** (object? data)

### Свойства

* IDataTemplate? **ElementDataTemplate** [get, set]

*Возвращает и задаёт шаблона данных для элементарной задачи.*

* IDataTemplate? **CompositeDataTemplate** [get, set]

*Возвращает и задаёт шаблона данных для составной задачи.*

* IDataTemplate? **DataTemplate** [get, set]

*Возвращает и задаёт шаблона данных для задачи.*

### Подробное описание

Класс шаблона данных для задач.

Наследует ITreeDataTemplate.

См. определение в файле **TaskDataTemplate.cs** строка **15**

### Методы

#### Control? View.DataTemplates.TaskDataTemplate.Build (object? param)

См. определение в файле **TaskDataTemplate.cs** строка **33**

#### InstancedBinding? View.DataTemplates.TaskDataTemplate.ItemsSelector (object item)

См. определение в файле **TaskDataTemplate.cs** строка **51**

### Полный список свойств

#### IDataTemplate? View.DataTemplates.TaskDataTemplate.CompositeDataTemplate[get], [set]

Возвращает и задаёт шаблона данных для составной задачи.

См. определение в файле **TaskDataTemplate.cs** строка **25**

#### IDataTemplate? View.DataTemplates.TaskDataTemplate.DataTemplate[get], [set]

Возвращает и задаёт шаблона данных для задачи.

См. определение в файле **TaskDataTemplate.cs** строка **30**

#### IDataTemplate? View.DataTemplates.TaskDataTemplate.ElementDataTemplate[get], [set]

Возвращает и задаёт шаблона данных для элементарной задачи.

См. определение в файле **TaskDataTemplate.cs** строка **20**

## Класс Model.Tasks.TaskElement

Класс элементарной задачи.

Граф наследования:Model.Tasks.TaskElement:

### Открытые члены

* virtual object **Clone** ()

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Свойства

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

* int **Difficult** [get, set]

*Возвращает и задаёт сложность.*

* int **Priority** [get, set]

*Возвращает и задаёт приоритет.*

* DateTime? **Deadline** [get, set]

*Возвращает и задаёт срок.*

* **TaskStatus** **Status** [get, set]

*Возвращает и задаёт статус.*

* **ITimeIntervalList** **TimeIntervals** [get]

*Возвращает список временных интервалов.*

* double **Progress** [get, set]

*Возвращает и задаёт прогресс.*

* TimeSpan **PlannedTime** [get, set]

*Возвращает и задаёт запланированное время.*

* TimeSpan **SpentTime** [get, set]

*Возвращает и задаёт потраченное время.*

* double **PlannedReal** [get, set]

*Возвращает и задаёт запланированный реальный показатель.*

* double **ExecutedReal** [get, set]

*Возвращает и задаёт выполненный реальный показатель.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс элементарной задачи.

Наследует **TrackableObject**. Реализует **ITaskElement** и ICloneable.

См. определение в файле **TaskElement.cs** строка **15**

### Методы

#### virtual object Model.Tasks.TaskElement.Clone ()[virtual]

Переопределяется в **ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskElementDomain** (*стр.461*).

См. определение в файле **TaskElement.cs** строка **158**

### Полный список свойств

#### DateTime? Model.Tasks.TaskElement.Deadline[get], [set]

Возвращает и задаёт срок.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElement.cs** строка **106**

#### int Model.Tasks.TaskElement.Difficult[get], [set]

Возвращает и задаёт сложность.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElement.cs** строка **92**

#### double Model.Tasks.TaskElement.ExecutedReal[get], [set]

Возвращает и задаёт выполненный реальный показатель.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElement.cs** строка **151**

#### object? Model.Tasks.TaskElement.Metadata[get], [set]

Возращает и задаёт метаданные.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TaskElement.cs** строка **85**

#### ITaskComposite? Model.Tasks.TaskElement.ParentTask[get], [set]

Возращает и задаёт родительскую задачу.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TaskElement.cs** строка **78**

#### double Model.Tasks.TaskElement.PlannedReal[get], [set]

Возвращает и задаёт запланированный реальный показатель.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElement.cs** строка **144**

#### TimeSpan Model.Tasks.TaskElement.PlannedTime[get], [set]

Возвращает и задаёт запланированное время.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElement.cs** строка **130**

#### int Model.Tasks.TaskElement.Priority[get], [set]

Возвращает и задаёт приоритет.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElement.cs** строка **99**

#### double Model.Tasks.TaskElement.Progress[get], [set]

Возвращает и задаёт прогресс.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElement.cs** строка **123**

#### TimeSpan Model.Tasks.TaskElement.SpentTime[get], [set]

Возвращает и задаёт потраченное время.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElement.cs** строка **137**

#### TaskStatus Model.Tasks.TaskElement.Status[get], [set]

Возвращает и задаёт статус.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElement.cs** строка **113**

#### ITimeIntervalList Model.Tasks.TaskElement.TimeIntervals[get]

Возвращает список временных интервалов.

Замещает **Model.Interfaces.ITaskElement** (*стр.304*).

См. определение в файле **TaskElement.cs** строка **120**

## Класс ViewModel.Implementations.DataManagers.TaskElementCreatorProxy

Граф наследования:ViewModel.Implementations.DataManagers.TaskElementCreatorProxy:

### Открытые члены

* **TaskElementCreatorProxy** (**ITaskElement** taskElement, BaseSupervisedEvaluatorLearningController< IEnumerable< double >, double, **ITaskElement**, **ITaskElement**, double > plannedRealLearningController, BaseSupervisedEvaluatorLearningController< IEnumerable< double >, double, **ITaskElement**, **ITaskElement**, TimeSpan > plannedTimeLearningController, BaseSupervisedEvaluatorLearningController< IEnumerable< double >, double, **ITaskElement**, **ITaskElement**, DateTime?> deadlineLearningController)

*Создаёт экземпляр класса* ***TaskElementCreatorProxy****.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* void **UpdatePredictedValues< T >** (T oldValue, T newValue)

*Обновляет предсказанные значения.*

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Свойства

* int **Difficult** [get, set]

*Возвращает и задаёт сложность.*

* int **Priority** [get, set]

*Возвращает и задаёт приоритет.*

* TaskStatus **Status** [get, set]

*Возвращает и задаёт статус.*

* DateTime? **Deadline** [get, set]

*Возвращает и задаёт срок.*

* double **Progress** [get, set]

*Возвращает и задаёт прогресс.*

* TimeSpan **PlannedTime** [get, set]

*Возвращает и задаёт запланированное время.*

* TimeSpan **SpentTime** [get, set]

*Возвращает и задаёт потраченное время.*

* double **PlannedReal** [get, set]

*Возвращает и задаёт запланированный реальный показатель.*

* double **ExecutedReal** [get, set]

*Возвращает и задаёт выполненный реальный показатель.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

* **ITimeIntervalList** **TimeIntervals** [get]

*Возвращает список временных интервалов.*

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* **ITaskElement** **Target** [get]

*Возвращает заменяемого объекта.*

* DateTime? **PredictedDeadline** [get, protected set]

*Возвращает предсказанный срок.*

* TimeSpan **PredictedPlannedTime** [get, protected set]

*Возвращает предсказанное запланированное время.*

* double **PredictedPlannedReal** [get, protected set]

*Возвращает предсказанный реальный запланированный показатель.*

* bool **IsValidPredictedDeadline** [get]

*Возвращает логическое значение, указывающее на корректность предсказания срок.*

* bool **IsValidPredictedPlannedTime** [get]

*Возвращает логическое значение, указывающее на корректность предсказания запланированного времени.*

* bool **IsValidPredictedPlannedReal** [get]

*Возвращает логическое значение, указывающее на корректность предсказания реального запланированного показателя.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

См. определение в файле **TaskElementCreatorProxy.cs** строка **12**

### Конструктор(ы)

#### ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.TaskElementCreatorProxy (ITaskElement taskElement, BaseSupervisedEvaluatorLearningController< IEnumerable< double >, double, ITaskElement, ITaskElement, double > plannedRealLearningController, BaseSupervisedEvaluatorLearningController< IEnumerable< double >, double, ITaskElement, ITaskElement, TimeSpan > plannedTimeLearningController, BaseSupervisedEvaluatorLearningController< IEnumerable< double >, double, ITaskElement, ITaskElement, DateTime?> deadlineLearningController)

Создаёт экземпляр класса **TaskElementCreatorProxy**.

##### Аргументы

|  |  |
| --- | --- |
| *taskElement* | Элементарная задача. |
| *plannedRealLearningController* | Контроллер обучения модели обучения запланнированных реальных показателей. |
| *plannedTimeLearningController* | Контроллер обучения модели обучения запланнированного времени. |
| *deadlineLearningController* | Контроллер обучения модели обучения срока. |

См. определение в файле **TaskElementCreatorProxy.cs** строка **192**

### Методы

#### void ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.UpdatePredictedValues< T > (T oldValue, T newValue)[protected]

Обновляет предсказанные значения.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип данных. |

##### Аргументы

|  |  |
| --- | --- |
| *oldValue* | Старые значения. |
| *newValue* | Новые значения. |

См. определение в файле **TaskElementCreatorProxy.cs** строка **215**

### Полный список свойств

#### DateTime? ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.Deadline[get], [set]

Возвращает и задаёт срок.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **80**

#### int ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.Difficult[get], [set]

Возвращает и задаёт сложность.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **56**

#### double ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.ExecutedReal[get], [set]

Возвращает и задаёт выполненный реальный показатель.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **120**

#### bool ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.IsValidPredictedDeadline[get]

Возвращает логическое значение, указывающее на корректность предсказания срок.

Замещает **ViewModel.Interfaces.DataManagers.ITaskElementProxy** (*стр.307*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **171**

#### bool ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.IsValidPredictedPlannedReal[get]

Возвращает логическое значение, указывающее на корректность предсказания реального запланированного показателя.

Замещает **ViewModel.Interfaces.DataManagers.ITaskElementProxy** (*стр.307*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **177**

#### bool ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.IsValidPredictedPlannedTime[get]

Возвращает логическое значение, указывающее на корректность предсказания запланированного времени.

Замещает **ViewModel.Interfaces.DataManagers.ITaskElementProxy** (*стр.307*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **174**

#### object? ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.Metadata[get], [set]

Возращает и задаёт метаданные.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **128**

#### ITaskComposite? ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.ParentTask[get], [set]

Возращает и задаёт родительскую задачу.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **139**

#### double ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.PlannedReal[get], [set]

Возвращает и задаёт запланированный реальный показатель.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **112**

#### TimeSpan ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.PlannedTime[get], [set]

Возвращает и задаёт запланированное время.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **96**

#### DateTime? ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.PredictedDeadline[get], [protected set]

Возвращает предсказанный срок.

Замещает **ViewModel.Interfaces.DataManagers.ITaskElementProxy** (*стр.307*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **150**

#### double ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.PredictedPlannedReal[get], [protected set]

Возвращает предсказанный реальный запланированный показатель.

Замещает **ViewModel.Interfaces.DataManagers.ITaskElementProxy** (*стр.307*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **164**

#### TimeSpan ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.PredictedPlannedTime[get], [protected set]

Возвращает предсказанное запланированное время.

Замещает **ViewModel.Interfaces.DataManagers.ITaskElementProxy** (*стр.307*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **157**

#### int ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.Priority[get], [set]

Возвращает и задаёт приоритет.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **64**

#### double ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.Progress[get], [set]

Возвращает и задаёт прогресс.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **88**

#### TimeSpan ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.SpentTime[get], [set]

Возвращает и задаёт потраченное время.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **104**

#### TaskStatus ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.Status[get], [set]

Возвращает и задаёт статус.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **72**

#### ITaskElement ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.Target[get]

Возвращает заменяемого объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IProxy< out T >** (*стр.268*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **147**

#### ITimeIntervalList ViewModel.Implementations.DataManagers.TaskElementCreatorProxy.TimeIntervals[get]

Возвращает список временных интервалов.

Замещает **Model.Interfaces.ITaskElement** (*стр.304*).

См. определение в файле **TaskElementCreatorProxy.cs** строка **136**

## Класс ViewModel.Implementations.DataManagers.Factories.TaskElementCreatorProxyFactory

Класс фабрики, создающая заместителей элементарной задачи.

Граф наследования:ViewModel.Implementations.DataManagers.Factories.TaskElementCreatorProxyFactory:

### Открытые члены

* **TaskElementCreatorProxyFactory** (IFactory< **ITaskElement** > factory, **PlannedRealTaskElementEvaluatorLearningController** plannedRealLearningController, **PlannedTimeTaskElementEvaluatorLearningController** plannedTimeLearningController, **DeadlineTaskElementEvaluatorLearningController** deadlineLearningController)

*Создаёт экземпляр класса* ***TaskElementCreatorProxyFactory****.*

* **ITaskElementProxy** **Create** ()

*Создает новый экземпляр объекта.*

##### *Возвращает*

Возвращает новый экземпляр объекта.

### Подробное описание

Класс фабрики, создающая заместителей элементарной задачи.

Реализует **IFactory<ITaskElementProxy>**.

См. определение в файле **TaskElementCreatorProxyFactory.cs** строка **15**

### Конструктор(ы)

#### ViewModel.Implementations.DataManagers.Factories.TaskElementCreatorProxyFactory.TaskElementCreatorProxyFactory (IFactory< ITaskElement > factory, PlannedRealTaskElementEvaluatorLearningController plannedRealLearningController, PlannedTimeTaskElementEvaluatorLearningController plannedTimeLearningController, DeadlineTaskElementEvaluatorLearningController deadlineLearningController)

Создаёт экземпляр класса **TaskElementCreatorProxyFactory**.

##### Аргументы

|  |  |
| --- | --- |
| *factory* | Фабрика, создающая элементарные задачи. |
| *plannedRealLearningController* | Контроллер обучения модели обучения запланнированных реальных показателей. |
| *plannedTimeLearningController* | Контроллер обучения модели обучения запланнированного времени. |
| *deadlineLearningController* | Контроллер обучения модели обучения срока. |

См. определение в файле **TaskElementCreatorProxyFactory.cs** строка **50**

### Методы

#### ITaskElementProxy ViewModel.Implementations.DataManagers.Factories.TaskElementCreatorProxyFactory.Create ()

Создает новый экземпляр объекта.

##### Возвращает

Возвращает новый экземпляр объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >** (*стр.245*).

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskElementDomain

Класс домменной модели элементарной задачи.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskElementDomain:

### Открытые члены

* override object **Clone** ()

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Свойства

* **TaskElementEntity** **Entity** [get, set]

*Возвращает и задаёт связанную сущность.*

* object **EntityId** [get]

*Возвращает индетификатор связанной сущности.*

#### Свойства унаследованные от Model.Tasks.TaskElement

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

* int **Difficult** [get, set]

*Возвращает и задаёт сложность.*

* int **Priority** [get, set]

*Возвращает и задаёт приоритет.*

* DateTime? **Deadline** [get, set]

*Возвращает и задаёт срок.*

* **TaskStatus** **Status** [get, set]

*Возвращает и задаёт статус.*

* **ITimeIntervalList** **TimeIntervals** [get]

*Возвращает список временных интервалов.*

* double **Progress** [get, set]

*Возвращает и задаёт прогресс.*

* TimeSpan **PlannedTime** [get, set]

*Возвращает и задаёт запланированное время.*

* TimeSpan **SpentTime** [get, set]

*Возвращает и задаёт потраченное время.*

* double **PlannedReal** [get, set]

*Возвращает и задаёт запланированный реальный показатель.*

* double **ExecutedReal** [get, set]

*Возвращает и задаёт выполненный реальный показатель.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс домменной модели элементарной задачи.

Наследует **TaskElement**. Реализует **IDomain**.

См. определение в файле **TaskElementDomain.cs** строка **14**

### Методы

#### override object ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskElementDomain.Clone ()[virtual]

Переопределяет метод предка **Model.Tasks.TaskElement** (*стр.448*).

См. определение в файле **TaskElementDomain.cs** строка **25**

### Полный список свойств

#### TaskElementEntity ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskElementDomain.Entity[get], [set]

Возвращает и задаёт связанную сущность.

См. определение в файле **TaskElementDomain.cs** строка **19**

#### object ViewModel.Implementations.AppStates.Sessions.Database.Domains.TaskElementDomain.EntityId[get]

Возвращает индетификатор связанной сущности.

Замещает **ViewModel.Implementations.AppStates.Sessions.Database.Domains.IDomain** (*стр.241*).

См. определение в файле **TaskElementDomain.cs** строка **22**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity

Класс сущности элементарной задачи.

### Свойства

* int **Id** [get, set]

*Возвращает и задаёт индентификатор.*

* int **Difficult** [get, set]

*Возвращает и задаёт сложность.*

* int **Priority** [get, set]

*Возвращает и задаёт приоритет.*

* DateTime? **Deadline** [get, set]

*Возвращает и задаёт срока.*

* TaskStatus **Status** [get, set]

*Возвращает и задаёт статус.*

* double **Progress** [get, set]

*Возвращает и задаёт прогресс.*

* TimeSpan **PlannedTime** [get, set]

*Возвращает и задаёт запланированное времени.*

* TimeSpan **SpentTime** [get, set]

*Возвращает и задаёт проведённое времени.*

* double **PlannedReal** [get, set]

*Возвращает и задаёт запланированный реальный показатель.*

* double **ExecutedReal** [get, set]

*Возвращает и задаёт выполненный реальный показатель.*

* virtual **TaskEntity** **Task** [get, set]

*Возвращает и задаёт задачу.*

* virtual ICollection< **TimeIntervalEntity** > **TimeIntervals** [get, set]

*Возвращает и задаёт временные интервалы.*

### Подробное описание

Класс сущности элементарной задачи.

См. определение в файле **TaskElementEntity.cs** строка **13**

### Полный список свойств

#### DateTime? ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.Deadline[get], [set]

Возвращает и задаёт срока.

См. определение в файле **TaskElementEntity.cs** строка **33**

#### int ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.Difficult[get], [set]

Возвращает и задаёт сложность.

См. определение в файле **TaskElementEntity.cs** строка **23**

#### double ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.ExecutedReal[get], [set]

Возвращает и задаёт выполненный реальный показатель.

См. определение в файле **TaskElementEntity.cs** строка **63**

#### int ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.Id[get], [set]

Возвращает и задаёт индентификатор.

См. определение в файле **TaskElementEntity.cs** строка **18**

#### double ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.PlannedReal[get], [set]

Возвращает и задаёт запланированный реальный показатель.

См. определение в файле **TaskElementEntity.cs** строка **58**

#### TimeSpan ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.PlannedTime[get], [set]

Возвращает и задаёт запланированное времени.

См. определение в файле **TaskElementEntity.cs** строка **48**

#### int ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.Priority[get], [set]

Возвращает и задаёт приоритет.

См. определение в файле **TaskElementEntity.cs** строка **28**

#### double ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.Progress[get], [set]

Возвращает и задаёт прогресс.

См. определение в файле **TaskElementEntity.cs** строка **43**

#### TimeSpan ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.SpentTime[get], [set]

Возвращает и задаёт проведённое времени.

См. определение в файле **TaskElementEntity.cs** строка **53**

#### TaskStatus ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.Status[get], [set]

Возвращает и задаёт статус.

См. определение в файле **TaskElementEntity.cs** строка **38**

#### virtual TaskEntity ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.Task[get], [set]

Возвращает и задаёт задачу.

См. определение в файле **TaskElementEntity.cs** строка **69**

#### virtual ICollection<TimeIntervalEntity> ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskElementEntity.TimeIntervals[get], [set]

Возвращает и задаёт временные интервалы.

См. определение в файле **TaskElementEntity.cs** строка **74**

## Класс ViewModel.Implementations.DataManagers.Factories.TaskElementFactory

Класс фабрики, создающая элементарные задачи.

Граф наследования:ViewModel.Implementations.DataManagers.Factories.TaskElementFactory:

### Открытые члены

* **TaskElementFactory** (IFactory< object > metadataFactory)

*Создаёт экземпляр класса* ***TaskElementFactory****.*

* **ITaskElement** **Create** ()

*Создает новый экземпляр объекта.*

##### *Возвращает*

Возвращает новый экземпляр объекта.

### Подробное описание

Класс фабрики, создающая элементарные задачи.

Реализует **IFactory<ITaskElement>**.

См. определение в файле **TaskElementFactory.cs** строка **14**

### Конструктор(ы)

#### ViewModel.Implementations.DataManagers.Factories.TaskElementFactory.TaskElementFactory (IFactory< object > metadataFactory)

Создаёт экземпляр класса **TaskElementFactory**.

##### Аргументы

|  |  |
| --- | --- |
| *metadataFactory* | Фабрика, создающая метаданные. |

См. определение в файле **TaskElementFactory.cs** строка **25**

### Методы

#### ITaskElement ViewModel.Implementations.DataManagers.Factories.TaskElementFactory.Create ()

Создает новый экземпляр объекта.

##### Возвращает

Возвращает новый экземпляр объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >** (*стр.245*).

См. определение в файле **TaskElementFactory.cs** строка **32**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskElementMapper

Класс перобразования значений элементарных задач между двумя предметными областями.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskElementMapper:

### Открытые члены

* **TaskElementMapper** (IMapper< **MetadataEntity**, object > metadataMapper, IMapper< **TimeIntervalEntity**, **ITimeIntervalElement** > timeIntervalMapper)

*Создаёт экземпляр класса* ***TaskElementMapper****.*

* **ITaskElement** **Map** (**TaskElementEntity** value)
* **TaskElementEntity** **MapBack** (**ITaskElement** value)

#### Открытые члены унаследованные от ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 >

* T2 **Map** (T1 value)

*Преобразует исходные данные в целевые данные.*

* T1 **MapBack** (T2 value)

*Преобразует целевые данные в исходные данные.*

### Подробное описание

Класс перобразования значений элементарных задач между двумя предметными областями.

Реализует **IMapper<TimeIntervalEntity, ITimeIntervalElement>**.

См. определение в файле **TaskElementMapper.cs** строка **14**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskElementMapper.TaskElementMapper (IMapper< MetadataEntity, object > metadataMapper, IMapper< TimeIntervalEntity, ITimeIntervalElement > timeIntervalMapper)

Создаёт экземпляр класса **TaskElementMapper**.

##### Аргументы

|  |  |
| --- | --- |
| *metadataMapper* | Преобразование значений между сущностью метаданных и метаданными. |
| *timeIntervalMapper* | Преобразование значений между сущностью временных интервалов и элементарными временными интервалами. |

См. определение в файле **TaskElementMapper.cs** строка **37**

### Методы

#### ITaskElement ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskElementMapper.Map (TaskElementEntity value)

См. определение в файле **TaskElementMapper.cs** строка **45**

#### TaskElementEntity ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskElementMapper.MapBack (ITaskElement value)

См. определение в файле **TaskElementMapper.cs** строка **70**

## Класс ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy

Класс заместитель элементарых задач для редактирования.

Граф наследования:ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy:

### Открытые члены

* void **ApplyChanges** ()

*Применения изменений.*

### Свойства

* **ITaskElement** **Target** [get, set]

*Возвращает и задаёт заменяемого объекта.*

* int **Difficult** [get, set]

*Возвращает и задаёт сложность.*

* int **Priority** [get, set]

*Возвращает и задаёт приоритет.*

* TaskStatus **Status** [get, set]

*Возвращает и задаёт статус.*

* DateTime? **Deadline** [get, set]

*Возвращает и задаёт срок.*

* double **Progress** [get, set]

*Возвращает и задаёт прогресс.*

* TimeSpan **PlannedTime** [get, set]

*Возвращает и задаёт запланированное время.*

* TimeSpan **SpentTime** [get, set]

*Возвращает и задаёт потраченное время.*

* double **PlannedReal** [get, set]

*Возвращает и задаёт запланированный реальный показатель.*

* double **ExecutedReal** [get, set]

*Возвращает и задаёт выполненный реальный показатель.*

* **ITimeIntervalList** **TimeIntervals** [get]

*Возвращает список временных интервалов.*

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

### Подробное описание

Класс заместитель элементарых задач для редактирования.

Реализует **ITaskElementsEditorProxy**.

См. определение в файле **TaskElementsEditorProxy.cs** строка **15**

### Методы

#### void ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.ApplyChanges ()

Применения изменений.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IEditorService** (*стр.244*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **71**

### Полный список свойств

#### DateTime? ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.Deadline[get], [set]

Возвращает и задаёт срок.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **43**

#### int ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.Difficult[get], [set]

Возвращает и задаёт сложность.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **34**

#### double ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.ExecutedReal[get], [set]

Возвращает и задаёт выполненный реальный показатель.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **58**

#### object? ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.Metadata[get], [set]

Возращает и задаёт метаданные.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **68**

#### ITaskComposite? ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.ParentTask[get], [set]

Возращает и задаёт родительскую задачу.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **65**

#### double ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.PlannedReal[get], [set]

Возвращает и задаёт запланированный реальный показатель.

Замещает **Model.Interfaces.ITaskElement** (*стр.302*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **55**

#### TimeSpan ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.PlannedTime[get], [set]

Возвращает и задаёт запланированное время.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **49**

#### int ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.Priority[get], [set]

Возвращает и задаёт приоритет.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **37**

#### double ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.Progress[get], [set]

Возвращает и задаёт прогресс.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **46**

#### TimeSpan ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.SpentTime[get], [set]

Возвращает и задаёт потраченное время.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **52**

#### TaskStatus ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.Status[get], [set]

Возвращает и задаёт статус.

Замещает **Model.Interfaces.ITaskElement** (*стр.303*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **40**

#### ITaskElement ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.Target[get], [set]

Возвращает и задаёт заменяемого объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T >** (*стр.242*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **23**

#### ITimeIntervalList ViewModel.Implementations.DataManagers.Editors.TaskElementsEditorProxy.TimeIntervals[get]

Возвращает список временных интервалов.

Замещает **Model.Interfaces.ITaskElement** (*стр.304*).

См. определение в файле **TaskElementsEditorProxy.cs** строка **61**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskEntity

Класс сущности задачи.

### Свойства

* int **Id** [get, set]

*Возвращает и задаёт индентификатор.*

* int? **ParentTaskId** [get, set]

*Возвращает и задаёт индентификатор родительской задачи.*

* virtual ? **TaskCompositeEntity** **ParentTask** [get, set]

*Возвращает и задаёт родительскую задачу.*

* virtual ? **TaskCompositeEntity** **TaskComposite** [get, set]

*Возвращает и задаёт составную задачу.*

* virtual ? **TaskElementEntity** **TaskElement** [get, set]

*Возвращает и задаёт элементарную задачу.*

* virtual **MetadataEntity** **Metadata** [get, set]

*Возвращает и задаёт метаданные.*

### Подробное описание

Класс сущности задачи.

См. определение в файле **TaskEntity.cs** строка **11**

### Полный список свойств

#### int ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskEntity.Id[get], [set]

Возвращает и задаёт индентификатор.

См. определение в файле **TaskEntity.cs** строка **16**

#### virtual MetadataEntity ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskEntity.Metadata[get], [set]

Возвращает и задаёт метаданные.

См. определение в файле **TaskEntity.cs** строка **42**

#### virtual ? TaskCompositeEntity ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskEntity.ParentTask[get], [set]

Возвращает и задаёт родительскую задачу.

См. определение в файле **TaskEntity.cs** строка **27**

#### int? ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskEntity.ParentTaskId[get], [set]

Возвращает и задаёт индентификатор родительской задачи.

См. определение в файле **TaskEntity.cs** строка **21**

#### virtual ? TaskCompositeEntity ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskEntity.TaskComposite[get], [set]

Возвращает и задаёт составную задачу.

См. определение в файле **TaskEntity.cs** строка **32**

#### virtual ? TaskElementEntity ViewModel.Implementations.AppStates.Sessions.Database.Entities.TaskEntity.TaskElement[get], [set]

Возвращает и задаёт элементарную задачу.

См. определение в файле **TaskEntity.cs** строка **37**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskMapper

Класс перобразования значений задач между двумя предметными областями.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskMapper:

### Открытые члены

* **TaskMapper** (IMapper< **TaskElementEntity**, **ITaskElement** > taskElementMapper, IMapper< **TaskCompositeEntity**, **ITaskComposite** > taskCompositeMapper)

*Создаёт экземпляр класса* ***TaskMapper****.*

* **ITask** **Map** (**TaskEntity** value)
* **TaskEntity** **MapBack** (**ITask** value)

#### Открытые члены унаследованные от ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 >

* T2 **Map** (T1 value)

*Преобразует исходные данные в целевые данные.*

* T1 **MapBack** (T2 value)

*Преобразует целевые данные в исходные данные.*

### Подробное описание

Класс перобразования значений задач между двумя предметными областями.

Реализует **IMapper<TimeIntervalEntity, ITimeIntervalElement>**.

См. определение в файле **TaskMapper.cs** строка **14**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskMapper.TaskMapper (IMapper< TaskElementEntity, ITaskElement > taskElementMapper, IMapper< TaskCompositeEntity, ITaskComposite > taskCompositeMapper)

Создаёт экземпляр класса **TaskMapper**.

##### Аргументы

|  |  |
| --- | --- |
| *taskElementMapper* | Преобразование значений между сущностью элементарных задач и элементарными задачами. |
| *taskCompositeMapper* | Преобразование значений между сущностью составных задач и составными задачами. |

См. определение в файле **TaskMapper.cs** строка **35**

### Методы

#### ITask ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskMapper.Map (TaskEntity value)

См. определение в файле **TaskMapper.cs** строка **43**

#### TaskEntity ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TaskMapper.MapBack (ITask value)

См. определение в файле **TaskMapper.cs** строка **57**

## Класс ViewModel.Technicals.TaskMetadata

Класс метаданных задачи.

Граф наследования:ViewModel.Technicals.TaskMetadata:

### Открытые члены

* object **Clone** ()
* override string **ToString** ()

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Свойства

* string **Title** [get, set]

*Возвращает и задаёт название.*

* string? **Description** [get, set]

*Возвращает и задаёт описание.*

* string? **Category** [get, set]

*Возвращает и задаёт категорию.*

* IEnumerable< string > **Tags** [get, set]

*Возвращает и задаёт теги.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс метаданных задачи.

Наследует **TrackableObject**. Реализует ICloneable.

См. определение в файле **TaskMetadata.cs** строка **12**

### Полный список свойств

#### string? ViewModel.Technicals.TaskMetadata.Category[get], [set]

Возвращает и задаёт категорию.

См. определение в файле **TaskMetadata.cs** строка **55**

#### string? ViewModel.Technicals.TaskMetadata.Description[get], [set]

Возвращает и задаёт описание.

См. определение в файле **TaskMetadata.cs** строка **46**

#### IEnumerable<string> ViewModel.Technicals.TaskMetadata.Tags[get], [set]

Возвращает и задаёт теги.

См. определение в файле **TaskMetadata.cs** строка **64**

#### string ViewModel.Technicals.TaskMetadata.Title[get], [set]

Возвращает и задаёт название.

См. определение в файле **TaskMetadata.cs** строка **37**

## Класс ViewModel.Implementations.DataManagers.Factories.TaskMetadataFactory

Класс фабрики, создающая метаданные задачи.

Граф наследования:ViewModel.Implementations.DataManagers.Factories.TaskMetadataFactory:

### Открытые члены

* object **Create** ()

*Создает новый экземпляр объекта.*

##### *Возвращает*

Возвращает новый экземпляр объекта.

### Подробное описание

Класс фабрики, создающая метаданные задачи.

Реализует **IFactory<object>**.

См. определение в файле **TaskMetadataFactory.cs** строка **12**

### Методы

#### object ViewModel.Implementations.DataManagers.Factories.TaskMetadataFactory.Create ()

Создает новый экземпляр объекта.

##### Возвращает

Возвращает новый экземпляр объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >** (*стр.245*).

## Класс ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy

Класс заместитель задач для редактирования.

Граф наследования:ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy:

### Открытые члены

* void **ApplyChanges** ()

*Применения изменений.*

### Свойства

* **ITask** **Target** [get, set]

*Возвращает и задаёт заменяемого объекта.*

* **ITaskComposite**? **ParentTask** [get, set]

*Возращает и задаёт родительскую задачу.*

* object? **Metadata** [get, set]

*Возращает и задаёт метаданные.*

* int **Difficult** [get]

*Возращает сложность.*

* int **Priority** [get]

*Возращает приоритет.*

* TaskStatus **Status** [get]

*Возращает статус.*

* DateTime? **Deadline** [get]

*Возращает срок.*

* double **Progress** [get]

*Возращает прогресс.*

* TimeSpan **PlannedTime** [get]

*Возращает запланированное время.*

* TimeSpan **SpentTime** [get]

*Возращает потраченное время.*

### Подробное описание

Класс заместитель задач для редактирования.

Реализует **ITasksEditorProxy**.

См. определение в файле **TasksEditorProxy.cs** строка **15**

### Методы

#### void ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.ApplyChanges ()

Применения изменений.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IEditorService** (*стр.244*).

См. определение в файле **TasksEditorProxy.cs** строка **61**

### Полный список свойств

#### DateTime? ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.Deadline[get]

Возращает срок.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TasksEditorProxy.cs** строка **49**

#### int ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.Difficult[get]

Возращает сложность.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TasksEditorProxy.cs** строка **40**

#### object? ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.Metadata[get], [set]

Возращает и задаёт метаданные.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TasksEditorProxy.cs** строка **37**

#### ITaskComposite? ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.ParentTask[get], [set]

Возращает и задаёт родительскую задачу.

Замещает **Model.Interfaces.ITask** (*стр.296*).

См. определение в файле **TasksEditorProxy.cs** строка **34**

#### TimeSpan ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.PlannedTime[get]

Возращает запланированное время.

Замещает **Model.Interfaces.ITask** (*стр.297*).

См. определение в файле **TasksEditorProxy.cs** строка **55**

#### int ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.Priority[get]

Возращает приоритет.

Замещает **Model.Interfaces.ITask** (*стр.297*).

См. определение в файле **TasksEditorProxy.cs** строка **43**

#### double ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.Progress[get]

Возращает прогресс.

Замещает **Model.Interfaces.ITask** (*стр.297*).

См. определение в файле **TasksEditorProxy.cs** строка **52**

#### TimeSpan ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.SpentTime[get]

Возращает потраченное время.

Замещает **Model.Interfaces.ITask** (*стр.297*).

См. определение в файле **TasksEditorProxy.cs** строка **58**

#### TaskStatus ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.Status[get]

Возращает статус.

Замещает **Model.Interfaces.ITask** (*стр.297*).

См. определение в файле **TasksEditorProxy.cs** строка **46**

#### ITask ViewModel.Implementations.DataManagers.Editors.TasksEditorProxy.Target[get], [set]

Возвращает и задаёт заменяемого объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T >** (*стр.242*).

См. определение в файле **TasksEditorProxy.cs** строка **23**

## Шаблон класса ViewModel.ViewModels.Modals.TasksViewModel< A, R >

Класс диалога задач.

Граф наследования:ViewModel.ViewModels.Modals.TasksViewModel< A, R >:

### Открытые члены

* **TasksViewModel** ()

*Создаёт экземпляр класса* ***TasksViewModel<A, R>*** *по умолчанию.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* async Task< R > **Invoke** (**BaseViewModel** parent, A args)

*Вызывает диалог.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Защищенные члены

* override void **GetArgs** (A args)

#### Защищенные члены унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* void **GetArgs** (A args)

*Получает аргументы, переданные в диалог.*

### Защищенные данные

* IObservable< bool > **\_canExecuteGoToPrevious**

*Наблюдатель, который отслеживает возможность выполнения GoToPrevious.*

* IObservable< bool > **\_canExecuteGo**

*Наблюдатель, который отслеживает возможность выполнения Go.*

* IEnumerable< **ITask** > **\_mainList**

*Основной список.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* TaskCompletionSource< R >? **\_taskSource**

*Источник для завершения задачи, который используется для возврата результата диалога.*

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

### Дополнительные унаследованные члены

#### Свойства унаследованные от ViewModel.ViewModels.BaseDialogViewModel< A, R >

* **BaseViewModel**? **Parent** [get]

*Возвращает родителя.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс диалога задач.

Наследует **BaseDialogViewModel<A, R>**.

**Согласование типов**

*A* : ***TasksViewModelArgs***

См. определение в файле **TasksViewModel.cs** строка **15**

### Методы

#### override void ViewModel.ViewModels.Modals.TasksViewModel< A, R >.GetArgs (A args)[protected]

См. определение в файле **TasksViewModel.cs** строка **63**

#### ViewModel.ViewModels.Modals.TasksViewModel< A, R >.TasksViewModel ()

Создаёт экземпляр класса **TasksViewModel<A, R>** по умолчанию.

См. определение в файле **TasksViewModel.cs** строка **54**

### Данные класса

#### IObservable<bool> ViewModel.ViewModels.Modals.TasksViewModel< A, R >.\_canExecuteGo[protected]

Наблюдатель, который отслеживает возможность выполнения Go.

См. определение в файле **TasksViewModel.cs** строка **26**

#### IObservable<bool> ViewModel.ViewModels.Modals.TasksViewModel< A, R >.\_canExecuteGoToPrevious[protected]

Наблюдатель, который отслеживает возможность выполнения GoToPrevious.

См. определение в файле **TasksViewModel.cs** строка **21**

#### IEnumerable<ITask> ViewModel.ViewModels.Modals.TasksViewModel< A, R >.\_mainList[protected]

Основной список.

См. определение в файле **TasksViewModel.cs** строка **31**

## Класс ViewModel.ViewModels.Modals.TasksViewModelArgs

Класс аргументов диалога списка задач.

Граф наследования:ViewModel.ViewModels.Modals.TasksViewModelArgs:

### Открытые члены

* **TasksViewModelArgs** (IEnumerable< **ITask** > list, IEnumerable< **ITask** > mainList)

*Создаёт экземпляр класса* ***TasksViewModelArgs****.*

### Свойства

* IEnumerable< **ITask** > **List** [get]

*Возвращает список.*

* IEnumerable< **ITask** > **MainList** [get]

*Возвращает основной список.*

### Подробное описание

Класс аргументов диалога списка задач.

См. определение в файле **TasksViewModelArgs.cs** строка **8**

### Конструктор(ы)

#### ViewModel.ViewModels.Modals.TasksViewModelArgs.TasksViewModelArgs (IEnumerable< ITask > list, IEnumerable< ITask > mainList)

Создаёт экземпляр класса **TasksViewModelArgs**.

##### Аргументы

|  |  |
| --- | --- |
| *list* | Список. |
| *mainList* | Основной список. |

См. определение в файле **TasksViewModelArgs.cs** строка **25**

### Полный список свойств

#### IEnumerable<ITask> ViewModel.ViewModels.Modals.TasksViewModelArgs.List[get]

Возвращает список.

См. определение в файле **TasksViewModelArgs.cs** строка **13**

#### IEnumerable<ITask> ViewModel.ViewModels.Modals.TasksViewModelArgs.MainList[get]

Возвращает основной список.

См. определение в файле **TasksViewModelArgs.cs** строка **18**

## Класс Model.Times.TimeIntervalElement

Класс элементарного временного интервала.

Граф наследования:Model.Times.TimeIntervalElement:

### Открытые члены

* **TimeIntervalElement** (DateTime? start=null, DateTime? end=null)

*Создаёт экземпляр класса* ***TimeIntervalElement****.*

* **TimeIntervalElement** ()

*Создаёт экземпляр класса* ***TimeIntervalElement*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Свойства

* DateTime **Start** [get, set]

*Возвращает и задаёт начало.*

* DateTime **End** [get, set]

*Возвращает и задаёт конец.*

* TimeSpan **Duration** [get]

*Возвращает длительность.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс элементарного временного интервала.

Наследует **TrackableObject**. Реализует **ITimeIntervalElement**.

См. определение в файле **TimeIntervalElement.cs** строка **15**

### Конструктор(ы)

#### Model.Times.TimeIntervalElement.TimeIntervalElement (DateTime? start = null, DateTime? end = null)

Создаёт экземпляр класса **TimeIntervalElement**.

##### Аргументы

|  |  |
| --- | --- |
| *start* | Начало. |
| *end* | Конец. |

См. определение в файле **TimeIntervalElement.cs** строка **49**

#### Model.Times.TimeIntervalElement.TimeIntervalElement ()

Создаёт экземпляр класса **TimeIntervalElement** по умолчанию.

См. определение в файле **TimeIntervalElement.cs** строка **58**

### Полный список свойств

#### TimeSpan Model.Times.TimeIntervalElement.Duration[get]

Возвращает длительность.

Замещает **Model.Interfaces.ITimeInterval** (*стр.318*).

См. определение в файле **TimeIntervalElement.cs** строка **42**

#### DateTime Model.Times.TimeIntervalElement.End[get], [set]

Возвращает и задаёт конец.

Замещает **Model.Interfaces.ITimeIntervalElement** (*стр.319*).

См. определение в файле **TimeIntervalElement.cs** строка **35**

#### DateTime Model.Times.TimeIntervalElement.Start[get], [set]

Возвращает и задаёт начало.

Замещает **Model.Interfaces.ITimeIntervalElement** (*стр.319*).

См. определение в файле **TimeIntervalElement.cs** строка **28**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Domains.TimeIntervalElementDomain

Класс домменной модели элементарного временного интервала.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.Domains.TimeIntervalElementDomain:

### Открытые члены

* **TimeIntervalElementDomain** (DateTime? start=null, DateTime? end=null)

*Создаёт экземпляр класса* ***TimeIntervalElementDomain****.*

* **TimeIntervalElementDomain** ()

*Создаёт экземпляр класса* ***TimeIntervalElementDomain*** *по умолчанию.*

#### Открытые члены унаследованные от Model.Times.TimeIntervalElement

* **TimeIntervalElement** (DateTime? start=null, DateTime? end=null)

*Создаёт экземпляр класса* ***TimeIntervalElement****.*

* **TimeIntervalElement** ()

*Создаёт экземпляр класса* ***TimeIntervalElement*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Свойства

* **TimeIntervalEntity** **Entity** [get, set]

*Возвращает и задаёт связанную сущность.*

* object **EntityId** [get]

*Возвращает индетификатор связанной сущности.*

#### Свойства унаследованные от Model.Times.TimeIntervalElement

* DateTime **Start** [get, set]

*Возвращает и задаёт начало.*

* DateTime **End** [get, set]

*Возвращает и задаёт конец.*

* TimeSpan **Duration** [get]

*Возвращает длительность.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс домменной модели элементарного временного интервала.

Наследует **TimeIntervalElement**. Реализует **IDomain**.

См. определение в файле **TimeIntervalElementDomain.cs** строка **14**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.Sessions.Database.Domains.TimeIntervalElementDomain.TimeIntervalElementDomain (DateTime? start = null, DateTime? end = null)

Создаёт экземпляр класса **TimeIntervalElementDomain**.

##### Аргументы

|  |  |
| --- | --- |
| *start* | Начало. |
| *end* | Конец. |

См. определение в файле **TimeIntervalElementDomain.cs** строка **29**

#### ViewModel.Implementations.AppStates.Sessions.Database.Domains.TimeIntervalElementDomain.TimeIntervalElementDomain ()

Создаёт экземпляр класса **TimeIntervalElementDomain** по умолчанию.

См. определение в файле **TimeIntervalElementDomain.cs** строка **36**

### Полный список свойств

#### TimeIntervalEntity ViewModel.Implementations.AppStates.Sessions.Database.Domains.TimeIntervalElementDomain.Entity[get], [set]

Возвращает и задаёт связанную сущность.

См. определение в файле **TimeIntervalElementDomain.cs** строка **19**

#### object ViewModel.Implementations.AppStates.Sessions.Database.Domains.TimeIntervalElementDomain.EntityId[get]

Возвращает индетификатор связанной сущности.

Замещает **ViewModel.Implementations.AppStates.Sessions.Database.Domains.IDomain** (*стр.241*).

См. определение в файле **TimeIntervalElementDomain.cs** строка **22**

## Класс ViewModel.Implementations.DataManagers.Factories.TimeIntervalElementFactory

Класс фабрики, создающая элементарные временные интервалы.

Граф наследования:ViewModel.Implementations.DataManagers.Factories.TimeIntervalElementFactory:

### Открытые члены

* **ITimeIntervalElement** **Create** ()

*Создает новый экземпляр объекта.*

##### *Возвращает*

Возвращает новый экземпляр объекта.

### Подробное описание

Класс фабрики, создающая элементарные временные интервалы.

Реализует **IFactory<ITimeIntervalElement>**.

См. определение в файле **TimeIntervalElementFactory.cs** строка **14**

### Методы

#### ITimeIntervalElement ViewModel.Implementations.DataManagers.Factories.TimeIntervalElementFactory.Create ()

Создает новый экземпляр объекта.

##### Возвращает

Возвращает новый экземпляр объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IFactory< out T >** (*стр.245*).

## Класс ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy

Класс заместитель элементарного временного интервала для редактирования.

Граф наследования:ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy:

### Открытые члены

* void **ApplyChanges** ()

*Применения изменений.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Открытые атрибуты

* **ITimeIntervalElement** **\_target**

*Заменяемый объект.*

### Свойства

* **ITimeIntervalElement** **Target** [get, set]

*Возвращает и задаёт заменяемого объекта.*

* DateTime **Start** [get, set]

*Возвращает и задаёт начало.*

* DateTime **End** [get, set]

*Возвращает и задаёт конец.*

* TimeSpan **Duration** [get]

*Возвращает длительность.*

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс заместитель элементарного временного интервала для редактирования.

Наследует **TrackableObject**. Реализует **ITimeIntervalElementsEditorProxy**.

См. определение в файле **TimeIntervalElementsEditorProxy.cs** строка **18**

### Методы

#### void ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy.ApplyChanges ()

Применения изменений.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IEditorService** (*стр.244*).

См. определение в файле **TimeIntervalElementsEditorProxy.cs** строка **65**

### Данные класса

#### ITimeIntervalElement ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy.\_target

Заменяемый объект.

См. определение в файле **TimeIntervalElementsEditorProxy.cs** строка **24**

### Полный список свойств

#### TimeSpan ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy.Duration[get]

Возвращает длительность.

Замещает **Model.Interfaces.ITimeInterval** (*стр.318*).

См. определение в файле **TimeIntervalElementsEditorProxy.cs** строка **62**

#### DateTime ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy.End[get], [set]

Возвращает и задаёт конец.

Замещает **Model.Interfaces.ITimeIntervalElement** (*стр.319*).

См. определение в файле **TimeIntervalElementsEditorProxy.cs** строка **55**

#### DateTime ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy.Start[get], [set]

Возвращает и задаёт начало.

Замещает **Model.Interfaces.ITimeIntervalElement** (*стр.319*).

См. определение в файле **TimeIntervalElementsEditorProxy.cs** строка **48**

#### ITimeIntervalElement ViewModel.Implementations.DataManagers.Editors.TimeIntervalElementsEditorProxy.Target[get], [set]

Возвращает и задаёт заменяемого объекта.

Замещает **ViewModel.Interfaces.DataManagers.Generals.IEditorProxy< T >** (*стр.242*).

См. определение в файле **TimeIntervalElementsEditorProxy.cs** строка **37**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Entities.TimeIntervalEntity

Класс сущности временных интервалов.

### Свойства

* int **Id** [get, set]

*Возвращает и задаёт индентификатор.*

* DateTime **Start** [get, set]

*Возвращает и задаёт начало.*

* DateTime **End** [get, set]

*Возвращает и задаёт конец.*

* int **TaskElementId** [get, set]

*Возвращает и задаёт индентификатор элементарной задачи.*

* virtual **TaskElementEntity** **TaskElement** [get, set]

*Возвращает и задаёт элементарную задачу.*

### Подробное описание

Класс сущности временных интервалов.

См. определение в файле **TimeIntervalEntity.cs** строка **11**

### Полный список свойств

#### DateTime ViewModel.Implementations.AppStates.Sessions.Database.Entities.TimeIntervalEntity.End[get], [set]

Возвращает и задаёт конец.

См. определение в файле **TimeIntervalEntity.cs** строка **26**

#### int ViewModel.Implementations.AppStates.Sessions.Database.Entities.TimeIntervalEntity.Id[get], [set]

Возвращает и задаёт индентификатор.

См. определение в файле **TimeIntervalEntity.cs** строка **16**

#### DateTime ViewModel.Implementations.AppStates.Sessions.Database.Entities.TimeIntervalEntity.Start[get], [set]

Возвращает и задаёт начало.

См. определение в файле **TimeIntervalEntity.cs** строка **21**

#### virtual TaskElementEntity ViewModel.Implementations.AppStates.Sessions.Database.Entities.TimeIntervalEntity.TaskElement[get], [set]

Возвращает и задаёт элементарную задачу.

См. определение в файле **TimeIntervalEntity.cs** строка **37**

#### int ViewModel.Implementations.AppStates.Sessions.Database.Entities.TimeIntervalEntity.TaskElementId[get], [set]

Возвращает и задаёт индентификатор элементарной задачи.

См. определение в файле **TimeIntervalEntity.cs** строка **31**

## Класс Model.Times.TimeIntervalList

Класс списка временных интервалов.

Граф наследования:Model.Times.TimeIntervalList:

### Открытые члены

* **TimeIntervalList** (IEnumerable< **ITimeIntervalElement** >? timeIntervals=null)

*Создаёт экземпляр класса* ***TimeIntervalList****.*

* **TimeIntervalList** ()

*Создаёт экземпляр класса* ***TimeIntervalList*** *по умолчанию.*

#### Открытые члены унаследованные от TrackableFeatures.TrackableCollection< T >

* **TrackableCollection** (IEnumerable< T >? items=null)
* bool **Contains** (T item)
* bool **Contains** (object? value)
* int **IndexOf** (T item)
* int **IndexOf** (object? value)
* void **CopyTo** (T[] array, int arrayIndex)
* void **CopyTo** (Array array, int index)
* void **Add** (T item)
* int **Add** (object? value)
* void **Insert** (int index, T item)
* void **Insert** (int index, object? value)
* bool **Remove** (T item)
* void **Remove** (object? value)
* void **RemoveAt** (int index)
* void **Replace** (int index, T item)

*Заменяет элемент по указанному индексу на новый.*

* void **Move** (int oldIndex, int newIndex)

*Перемещает элемент из одного индекса в другой.*

* void **Clear** ()
* IEnumerator< T > **GetEnumerator** ()

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* override void **OnAddedItem** (**ITimeIntervalElement** timeInterval, bool arePropertiesUpdate=true)
* override void **OnRemovedItem** (**ITimeIntervalElement** timeInterval)

#### Защищенные члены унаследованные от TrackableFeatures.TrackableCollection< T >

* int **AddItem** (T item)

*Добавляет элемент в коллекцию.*

* void **InsertItem** (int index, T item)

*Вставляет элемент в указанную позицию коллекции.*

* bool **RemoveItem** (T item)

*Удаляет элемент из коллекции.*

* virtual void **OnAddedItem** (T item, bool arePropertiesUpdate=true)

*Вызывается при добавлении элемента.*

* virtual void **OnRemovedItem** (T item)

*Вызывается при удалении элемента.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? oldItem, object? newItem, int index)

*Вызывает событие* ***CollectionChanged*** *для индекса, в котором изменился элемент.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? item, int index)

*Вызывает событие* ***CollectionChanged*** *для элемента, над которым совершили действие.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? item, int newIndex, int oldIndex)

*Вызывает событие* ***CollectionChanged*** *для элемента, который изменил индекс.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action)

*Вызывает событие* ***CollectionChanged*** *для действия.*

* void **OnCollectionChanged** (NotifyCollectionChangedEventArgs args)

*Вызывает событие* ***CollectionChanged*** *с аргументами.*

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Свойства

* TimeSpan **Duration** [get]

*Возвращает длительность.*

#### Свойства унаследованные от TrackableFeatures.TrackableCollection< T >

* T **this[int index]** [get, set]
* int **Count** [get]
* bool **IsReadOnly** [get]
* bool **IsFixedSize** [get]
* bool **IsSynchronized** [get]
* object **SyncRoot** [get]

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### Дополнительные унаследованные члены

#### События унаследованные от TrackableFeatures.TrackableCollection< T >

* NotifyCollectionChangedEventHandler? **CollectionChanged**

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Класс списка временных интервалов.

Наследует **TrackableCollection<ITimeIntervalElement>**. Реализует **ITimeIntervalList**.

См. определение в файле **TimeIntervalList.cs** строка **16**

### Конструктор(ы)

#### Model.Times.TimeIntervalList.TimeIntervalList (IEnumerable< ITimeIntervalElement >? timeIntervals = null)

Создаёт экземпляр класса **TimeIntervalList**.

##### Аргументы

|  |  |
| --- | --- |
| *timeIntervals* | Временные интервалы. |

См. определение в файле **TimeIntervalList.cs** строка **25**

#### Model.Times.TimeIntervalList.TimeIntervalList ()

Создаёт экземпляр класса **TimeIntervalList** по умолчанию.

См. определение в файле **TimeIntervalList.cs** строка **31**

### Методы

#### override void Model.Times.TimeIntervalList.OnAddedItem (ITimeIntervalElement timeInterval, bool arePropertiesUpdate = true)[protected]

См. определение в файле **TimeIntervalList.cs** строка **34**

#### override void Model.Times.TimeIntervalList.OnRemovedItem (ITimeIntervalElement timeInterval)[protected]

См. определение в файле **TimeIntervalList.cs** строка **48**

### Полный список свойств

#### TimeSpan Model.Times.TimeIntervalList.Duration[get]

Возвращает длительность.

Замещает **Model.Interfaces.ITimeInterval** (*стр.318*).

См. определение в файле **TimeIntervalList.cs** строка **19**

## Класс ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TimeIntervalMapper

Класс перобразования значений временных интервалов между двумя предметными областями.

Граф наследования:ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TimeIntervalMapper:

### Открытые члены

* **ITimeIntervalElement** **Map** (**TimeIntervalEntity** value)
* **TimeIntervalEntity** **MapBack** (**ITimeIntervalElement** value)

#### Открытые члены унаследованные от ViewModel.Implementations.AppStates.Sessions.Database.Mappers.IMapper< T1, T2 >

* T2 **Map** (T1 value)

*Преобразует исходные данные в целевые данные.*

* T1 **MapBack** (T2 value)

*Преобразует целевые данные в исходные данные.*

### Подробное описание

Класс перобразования значений временных интервалов между двумя предметными областями.

Реализует **IMapper<TimeIntervalEntity, ITimeIntervalElement>**.

См. определение в файле **TimeIntervalMapper.cs** строка **14**

### Методы

#### TimeIntervalEntity ViewModel.Implementations.AppStates.Sessions.Database.Mappers.TimeIntervalMapper.MapBack (ITimeIntervalElement value)

См. определение в файле **TimeIntervalMapper.cs** строка **24**

## Класс ViewModel.ViewModels.Modals.TimeIntervalViewModelArgs

Класс аргументов диалога **AddTimeIntervalViewModel**.

Граф наследования:ViewModel.ViewModels.Modals.TimeIntervalViewModelArgs:

### Открытые члены

* **TimeIntervalViewModelArgs** (IEnumerable< **ITask** > list, IEnumerable< **ITask** > mainList, **ITimeIntervalElement** timeIntervalElement)

*Создаёт экземпляр класса* ***TimeIntervalViewModelResult****.*

#### Открытые члены унаследованные от ViewModel.ViewModels.Modals.TasksViewModelArgs

* **TasksViewModelArgs** (IEnumerable< **ITask** > list, IEnumerable< **ITask** > mainList)

*Создаёт экземпляр класса* ***TasksViewModelArgs****.*

### Свойства

* **ITimeIntervalElement** **TimeIntervalElement** [get]

*Возвращает элементарный временной интервал.*

#### Свойства унаследованные от ViewModel.ViewModels.Modals.TasksViewModelArgs

* IEnumerable< **ITask** > **List** [get]

*Возвращает список.*

* IEnumerable< **ITask** > **MainList** [get]

*Возвращает основной список.*

### Подробное описание

Класс аргументов диалога **AddTimeIntervalViewModel**.

Наследует **TasksViewModelArgs**.

См. определение в файле **TimeIntervalViewModelArgs.cs** строка **11**

### Конструктор(ы)

#### ViewModel.ViewModels.Modals.TimeIntervalViewModelArgs.TimeIntervalViewModelArgs (IEnumerable< ITask > list, IEnumerable< ITask > mainList, ITimeIntervalElement timeIntervalElement)

Создаёт экземпляр класса **TimeIntervalViewModelResult**.

##### Аргументы

|  |  |
| --- | --- |
| *list* | Список. |
| *mainList* | Основной список. |
| *timeIntervalElement* | Элементарный временной интервал. |

См. определение в файле **TimeIntervalViewModelArgs.cs** строка **24**

### Полный список свойств

#### ITimeIntervalElement ViewModel.ViewModels.Modals.TimeIntervalViewModelArgs.TimeIntervalElement[get]

Возвращает элементарный временной интервал.

См. определение в файле **TimeIntervalViewModelArgs.cs** строка **16**

## Класс ViewModel.ViewModels.Modals.TimeIntervalViewModelResult

Класс результата диалога **AddTimeIntervalViewModel**.

### Открытые члены

* **TimeIntervalViewModelResult** (**ITaskElement** taskElement, **ITimeIntervalElement** timeIntervalElement)

*Создаёт экземпляр класса* ***TimeIntervalViewModelResult****.*

### Свойства

* **ITimeIntervalElement** **TimeIntervalElement** [get]

*Возвращает элементарный временной интервал.*

* **ITaskElement** **TaskElement** [get]

*Возвращает элементарную задачу.*

### Подробное описание

Класс результата диалога **AddTimeIntervalViewModel**.

См. определение в файле **TimeIntervalViewModelResult.cs** строка **8**

### Конструктор(ы)

#### ViewModel.ViewModels.Modals.TimeIntervalViewModelResult.TimeIntervalViewModelResult (ITaskElement taskElement, ITimeIntervalElement timeIntervalElement)

Создаёт экземпляр класса **TimeIntervalViewModelResult**.

##### Аргументы

|  |  |
| --- | --- |
| *taskElement* | Элементарная задача. |
| *timeIntervalElement* | Элементарный временной интервал. |

См. определение в файле **TimeIntervalViewModelResult.cs** строка **25**

### Полный список свойств

#### ITaskElement ViewModel.ViewModels.Modals.TimeIntervalViewModelResult.TaskElement[get]

Возвращает элементарную задачу.

См. определение в файле **TimeIntervalViewModelResult.cs** строка **18**

#### ITimeIntervalElement ViewModel.ViewModels.Modals.TimeIntervalViewModelResult.TimeIntervalElement[get]

Возвращает элементарный временной интервал.

См. определение в файле **TimeIntervalViewModelResult.cs** строка **13**

## Класс ViewModel.Implementations.AppStates.TimeScheduler

Класс планировщика времени, который отслеживает достижение временных меток.

Граф наследования:ViewModel.Implementations.AppStates.TimeScheduler:

### Открытые члены

* **TimeScheduler** ()

*Создаёт экземпляр класса* ***TimeScheduler*** *по умолчанию.*

### Свойства

* IList< DateTime > **Timepoints** [get]

*Возращает список временных меток.*

### События

* EventHandler< DateTime > **TimepointReached**

#### События унаследованные от ViewModel.Interfaces.ITimeScheduler

* EventHandler< DateTime > **TimepointReached**

*Событие, которое возникает при достижении временной метки.*

### Подробное описание

Класс планировщика времени, который отслеживает достижение временных меток.

Реализует **ITimeScheduler**.

См. определение в файле **TimeScheduler.cs** строка **18**

### Конструктор(ы)

#### ViewModel.Implementations.AppStates.TimeScheduler.TimeScheduler ()

Создаёт экземпляр класса **TimeScheduler** по умолчанию.

См. определение в файле **TimeScheduler.cs** строка **49**

### Полный список свойств

#### IList<DateTime> ViewModel.Implementations.AppStates.TimeScheduler.Timepoints[get]

Возращает список временных меток.

Замещает **ViewModel.Interfaces.ITimeScheduler** (*стр.324*).

См. определение в файле **TimeScheduler.cs** строка **41**

### Cобытия

#### EventHandler<DateTime> ViewModel.Implementations.AppStates.TimeScheduler.TimepointReached

См. определение в файле **TimeScheduler.cs** строка **44**

## Класс View.Views.Pages.TimeView

Класс пользовательского элемента страницы календаря.

Граф наследования:View.Views.Pages.TimeView:

### Открытые члены

* **TimeView** ()

*Создаёт экземпляр класса* ***TimeView*** *по умолчанию.*

### Защищенные члены

* override void **OnSizeChanged** (SizeChangedEventArgs e)

### Подробное описание

Класс пользовательского элемента страницы календаря.

Наследует ReactiveUserControl<TimeViewModel>.

См. определение в файле **TimeView.axaml.cs** строка **18**

### Конструктор(ы)

#### View.Views.Pages.TimeView.TimeView ()

Создаёт экземпляр класса **TimeView** по умолчанию.

См. определение в файле **TimeView.axaml.cs** строка **23**

### Методы

#### override void View.Views.Pages.TimeView.OnSizeChanged (SizeChangedEventArgs e)[protected]

См. определение в файле **TimeView.axaml.cs** строка **29**

## Класс ViewModel.ViewModels.Pages.TimeViewModel

Класс контроллера страницы календаря.

Граф наследования:ViewModel.ViewModels.Pages.TimeViewModel:

### Открытые члены

* **TimeViewModel** (**ISession** session, **IResourceService** resourceService, **ITimeScheduler** timeScheduler, **INotificationManager** notificationManager, BaseDialogViewModel< **TimeIntervalViewModelArgs**, **TimeIntervalViewModelResult** > addTimeIntervalDialog, BaseDialogViewModel< **ITimeIntervalElement**, bool > editTimeIntervalDialog, IFactory< **ITimeIntervalElement** > timeIntervalElementFactory, **ITimeIntervalElementsEditorProxy** timeIntervalElementsEditorProxy)

*Создаёт экземпляр класса* ***TimeViewModel****.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BasePageViewModel

* object **Metadata** [get, protected set]

*Возращает метаданные.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс контроллера страницы календаря.

Наследует **BasePageViewModel**.

См. определение в файле **TimeViewModel.cs** строка **26**

### Конструктор(ы)

#### ViewModel.ViewModels.Pages.TimeViewModel.TimeViewModel (ISession session, IResourceService resourceService, ITimeScheduler timeScheduler, INotificationManager notificationManager, BaseDialogViewModel< TimeIntervalViewModelArgs, TimeIntervalViewModelResult > addTimeIntervalDialog, BaseDialogViewModel< ITimeIntervalElement, bool > editTimeIntervalDialog, IFactory< ITimeIntervalElement > timeIntervalElementFactory, ITimeIntervalElementsEditorProxy timeIntervalElementsEditorProxy)

Создаёт экземпляр класса **TimeViewModel**.

##### Аргументы

|  |  |
| --- | --- |
| *session* | Сессия. |
| *resourceService* | Сервис ресурсов. |
| *timeScheduler* | Планировщик времени. |
| *notificationManager* | Менеджер уведомлений. |
| *addTimeIntervalDialog* | Диалог добавления временного интервала. |
| *editTimeIntervalDialog* | Диалог изменения временного интервала. |
| *timeIntervalElementFactory* | Фабрика, создающая элементарный временной интервал. |
| *timeIntervalElementsEditorProxy* | Заместитель элементарный временной интервал для редактирования. |

См. определение в файле **TimeViewModel.cs** строка **118**

## Класс ViewModel.Technicals.ToDoListElement

Класс элемента списка задач для выполнения.

### Открытые члены

* **ToDoListElement** (**ITaskElement** taskElement, double? executionChance, bool isLagging, bool isExpired)

*Создаёт экземпляр класса* ***ToDoListElement****.*

### Свойства

* **ITaskElement** **TaskElement** [get]

*Возвращает элементарную задачу.*

* double? **ExecutionChance** [get]

*Возвращает шанс выполнения.*

* bool **IsLagging** [get]

*Возвращает логическое значение, указывающее отстаёт ли задача.*

* bool **IsExpired** [get]

*Возвращает логическое значение, указывающее просрочена ли задача.*

### Подробное описание

Класс элемента списка задач для выполнения.

См. определение в файле **ToDoListElement.cs** строка **8**

### Конструктор(ы)

#### ViewModel.Technicals.ToDoListElement.ToDoListElement (ITaskElement taskElement, double? executionChance, bool isLagging, bool isExpired)

Создаёт экземпляр класса **ToDoListElement**.

##### Аргументы

|  |  |
| --- | --- |
| *taskElement* | Элементарная задача. |
| *executionChance* | Шанс выполнения. |
| *isLagging* | Логическое значение, указывающее отстаёт ли задача. |
| *isExpired* | Логическое значение, указывающее просрочена ли задача. |

См. определение в файле **ToDoListElement.cs** строка **37**

### Полный список свойств

#### double? ViewModel.Technicals.ToDoListElement.ExecutionChance[get]

Возвращает шанс выполнения.

См. определение в файле **ToDoListElement.cs** строка **18**

#### bool ViewModel.Technicals.ToDoListElement.IsExpired[get]

Возвращает логическое значение, указывающее просрочена ли задача.

См. определение в файле **ToDoListElement.cs** строка **28**

#### bool ViewModel.Technicals.ToDoListElement.IsLagging[get]

Возвращает логическое значение, указывающее отстаёт ли задача.

См. определение в файле **ToDoListElement.cs** строка **23**

#### ITaskElement ViewModel.Technicals.ToDoListElement.TaskElement[get]

Возвращает элементарную задачу.

См. определение в файле **ToDoListElement.cs** строка **13**

## Класс View.Views.Pages.ToDoListView

Класс пользовательского элемента страницы списка задач для выполнения.

Граф наследования:View.Views.Pages.ToDoListView:

### Открытые члены

* **ToDoListView** ()

*Создаёт экземпляр класса* ***ToDoListView*** *по умолчанию.*

### Подробное описание

Класс пользовательского элемента страницы списка задач для выполнения.

Наследует ReactiveUserControl<ToDoListViewModel>.

См. определение в файле **ToDoListView.axaml.cs** строка **13**

### Конструктор(ы)

#### View.Views.Pages.ToDoListView.ToDoListView ()

Создаёт экземпляр класса **ToDoListView** по умолчанию.

См. определение в файле **ToDoListView.axaml.cs** строка **18**

## Класс ViewModel.ViewModels.Pages.ToDoListViewModel

Класс контроллера страницы списка задач для выполнения.

Граф наследования:ViewModel.ViewModels.Pages.ToDoListViewModel:

### Открытые члены

* **ToDoListViewModel** (**ISession** session, **IResourceService** resourceService, **ExecutionChanceTaskElementEvaluatorLearningController** progressLearningController)

*Создаёт экземпляр класса* ***ToDoListViewModel****.*

* void **Update** ()

*Обновляет список задач для выполнения.*

#### Открытые члены унаследованные от ViewModel.ViewModels.BaseViewModel

* **BaseViewModel** ()

*Создаёт экемпляр класса* ***BaseViewModel*** *по умолчанию.*

* async Task< R > **AddDialog< A, R >** (BaseDialogViewModel< A, R > dialog, A args)

*Добавляет и вызывает диалог.*

* async Task< R > **AddModal< A, R >** (BaseDialogViewModel< A, R > modal, A args)

*Добавляет и вызывает модальный диалог.*

### Дополнительные унаследованные члены

#### Защищенные данные унаследованные от ViewModel.ViewModels.BaseViewModel

* readonly IObservable< bool > **\_modalsObservable**

*Наблюдатель, который отслеживает модальные диалоги.*

#### Свойства унаследованные от ViewModel.ViewModels.BasePageViewModel

* object **Metadata** [get, protected set]

*Возращает метаданные.*

#### Свойства унаследованные от ViewModel.ViewModels.BaseViewModel

* ObservableCollection< **BaseViewModel** > **Dialogs** [get]

*Возвращает диалоги.*

* ObservableCollection< **BaseViewModel** > **Modals** [get]

*Возвращает модальные диалоги.*

* ViewModelActivator **Activator** = new ViewModelActivator() [get]

*Возвращает активатор.*

### Подробное описание

Класс контроллера страницы списка задач для выполнения.

Наследует **BasePageViewModel**.

См. определение в файле **ToDoListViewModel.cs** строка **20**

### Конструктор(ы)

#### ViewModel.ViewModels.Pages.ToDoListViewModel.ToDoListViewModel (ISession session, IResourceService resourceService, ExecutionChanceTaskElementEvaluatorLearningController progressLearningController)

Создаёт экземпляр класса **ToDoListViewModel**.

##### Аргументы

|  |  |
| --- | --- |
| *session* | Сессия. |
| *resourceService* | Сервис ресурсов. |
| *progressLearningController* | Контроллер обучения модели выполнения шанса. |

См. определение в файле **ToDoListViewModel.cs** строка **88**

### Методы

#### void ViewModel.ViewModels.Pages.ToDoListViewModel.Update ()

Обновляет список задач для выполнения.

См. определение в файле **ToDoListViewModel.cs** строка **110**

## Шаблон класса TrackableFeatures.TrackableCollection< T >

Базовый класс коллекции, предоставляющий поддержку отслеживания изменений в коллекции.

Граф наследования:TrackableFeatures.TrackableCollection< T >:

### Открытые члены

* **TrackableCollection** (IEnumerable< T >? items=null)
* bool **Contains** (T item)
* bool **Contains** (object? value)
* int **IndexOf** (T item)
* int **IndexOf** (object? value)
* void **CopyTo** (T[] array, int arrayIndex)
* void **CopyTo** (Array array, int index)
* void **Add** (T item)
* int **Add** (object? value)
* void **Insert** (int index, T item)
* void **Insert** (int index, object? value)
* bool **Remove** (T item)
* void **Remove** (object? value)
* void **RemoveAt** (int index)
* void **Replace** (int index, T item)

*Заменяет элемент по указанному индексу на новый.*

* void **Move** (int oldIndex, int newIndex)

*Перемещает элемент из одного индекса в другой.*

* void **Clear** ()
* IEnumerator< T > **GetEnumerator** ()

#### Открытые члены унаследованные от TrackableFeatures.TrackableObject

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* int **AddItem** (T item)

*Добавляет элемент в коллекцию.*

* void **InsertItem** (int index, T item)

*Вставляет элемент в указанную позицию коллекции.*

* bool **RemoveItem** (T item)

*Удаляет элемент из коллекции.*

* virtual void **OnAddedItem** (T item, bool arePropertiesUpdate=true)

*Вызывается при добавлении элемента.*

* virtual void **OnRemovedItem** (T item)

*Вызывается при удалении элемента.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? oldItem, object? newItem, int index)

*Вызывает событие* ***CollectionChanged*** *для индекса, в котором изменился элемент.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? item, int index)

*Вызывает событие* ***CollectionChanged*** *для элемента, над которым совершили действие.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action, object? item, int newIndex, int oldIndex)

*Вызывает событие* ***CollectionChanged*** *для элемента, который изменил индекс.*

* void **OnCollectionChanged** (NotifyCollectionChangedAction action)

*Вызывает событие* ***CollectionChanged*** *для действия.*

* void **OnCollectionChanged** (NotifyCollectionChangedEventArgs args)

*Вызывает событие* ***CollectionChanged*** *с аргументами.*

#### Защищенные члены унаследованные от TrackableFeatures.TrackableObject

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Свойства

* T **this[int index]** [get, set]
* int **Count** [get]
* bool **IsReadOnly** [get]
* bool **IsFixedSize** [get]
* bool **IsSynchronized** [get]
* object **SyncRoot** [get]

#### Свойства унаследованные от TrackableFeatures.TrackableObject

* bool **HasErrors** [get]

### События

* NotifyCollectionChangedEventHandler? **CollectionChanged**

#### События унаследованные от TrackableFeatures.TrackableObject

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Базовый класс коллекции, предоставляющий поддержку отслеживания изменений в коллекции.

Наследует **TrackableObject**. Реализует INotifyCollectionChanged, IList и IList<T>.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип элементов коллекции. |

См. определение в файле **TrackableCollection.cs** строка **14**

### Методы

#### int TrackableFeatures.TrackableCollection< T >.AddItem (T item)[protected]

Добавляет элемент в коллекцию.

##### Аргументы

|  |  |
| --- | --- |
| *item* | Элемент. |

##### Возвращает

Индекс добавленного элемента.

См. определение в файле **TrackableCollection.cs** строка **172**

#### void TrackableFeatures.TrackableCollection< T >.Clear ()

См. определение в файле **TrackableCollection.cs** строка **149**

#### void TrackableFeatures.TrackableCollection< T >.CopyTo (Array array, int index)

См. определение в файле **TrackableCollection.cs** строка **82**

#### void TrackableFeatures.TrackableCollection< T >.InsertItem (int index, T item)[protected]

Вставляет элемент в указанную позицию коллекции.

##### Аргументы

|  |  |
| --- | --- |
| *index* | Индекс. |
| *item* | Элемент. |

См. определение в файле **TrackableCollection.cs** строка **188**

#### void TrackableFeatures.TrackableCollection< T >.Move (int oldIndex, int newIndex)

Перемещает элемент из одного индекса в другой.

##### Аргументы

|  |  |
| --- | --- |
| *oldIndex* | Старый индекс. |
| *newIndex* | Новый индекс. |

См. определение в файле **TrackableCollection.cs** строка **139**

#### virtual void TrackableFeatures.TrackableCollection< T >.OnAddedItem (T item, bool arePropertiesUpdate = true)[protected], [virtual]

Вызывается при добавлении элемента.

##### Аргументы

|  |  |
| --- | --- |
| *item* | Элемент. |
| *arePropertiesUpdate* | Флаг обновления свойств. |

См. определение в файле **TrackableCollection.cs** строка **221**

#### void TrackableFeatures.TrackableCollection< T >.OnCollectionChanged (NotifyCollectionChangedAction action)[protected]

Вызывает событие **CollectionChanged** для действия.

##### Аргументы

|  |  |
| --- | --- |
| *action* | Действие. |

#### void TrackableFeatures.TrackableCollection< T >.OnCollectionChanged (NotifyCollectionChangedAction action, object? item, int index)[protected]

Вызывает событие **CollectionChanged** для элемента, над которым совершили действие.

##### Аргументы

|  |  |
| --- | --- |
| *action* | Действие. |
| *item* | Элемент. |
| *index* | Индекс. |

#### void TrackableFeatures.TrackableCollection< T >.OnCollectionChanged (NotifyCollectionChangedAction action, object? item, int newIndex, int oldIndex)[protected]

Вызывает событие **CollectionChanged** для элемента, который изменил индекс.

##### Аргументы

|  |  |
| --- | --- |
| *action* | Действие. |
| *item* | Элемент. |
| *newIndex* | Новый индекс. |
| *oldIndex* | Старый индекс. |

#### void TrackableFeatures.TrackableCollection< T >.OnCollectionChanged (NotifyCollectionChangedAction action, object? oldItem, object? newItem, int index)[protected]

Вызывает событие **CollectionChanged** для индекса, в котором изменился элемент.

##### Аргументы

|  |  |
| --- | --- |
| *action* | Действие. |
| *oldItem* | Старый элемент. |
| *newItem* | Новый элемент. |
| *index* | Индекс. |

#### void TrackableFeatures.TrackableCollection< T >.OnCollectionChanged (NotifyCollectionChangedEventArgs args)[protected]

Вызывает событие **CollectionChanged** с аргументами.

##### Аргументы

|  |  |
| --- | --- |
| *args* | Аргументы. |

#### virtual void TrackableFeatures.TrackableCollection< T >.OnRemovedItem (T item)[protected], [virtual]

Вызывается при удалении элемента.

##### Аргументы

|  |  |
| --- | --- |
| *item* | Удалённый элемент. |

См. определение в файле **TrackableCollection.cs** строка **227**

#### void TrackableFeatures.TrackableCollection< T >.RemoveAt (int index)

См. определение в файле **TrackableCollection.cs** строка **109**

#### bool TrackableFeatures.TrackableCollection< T >.RemoveItem (T item)[protected]

Удаляет элемент из коллекции.

##### Аргументы

|  |  |
| --- | --- |
| *item* | Элемент. |

##### Возвращает

Возвращает true , если элемент был удалён, иначе false .

См. определение в файле **TrackableCollection.cs** строка **202**

#### void TrackableFeatures.TrackableCollection< T >.Replace (int index, T item)

Заменяет элемент по указанному индексу на новый.

##### Аргументы

|  |  |
| --- | --- |
| *index* | Индекс. |
| *item* | Элемент. |

См. определение в файле **TrackableCollection.cs** строка **124**

#### TrackableFeatures.TrackableCollection< T >.TrackableCollection (IEnumerable< T >? items = null)

См. определение в файле **TrackableCollection.cs** строка **54**

### Полный список свойств

#### int TrackableFeatures.TrackableCollection< T >.Count[get]

См. определение в файле **TrackableCollection.cs** строка **40**

#### bool TrackableFeatures.TrackableCollection< T >.IsFixedSize[get]

См. определение в файле **TrackableCollection.cs** строка **46**

#### bool TrackableFeatures.TrackableCollection< T >.IsReadOnly[get]

См. определение в файле **TrackableCollection.cs** строка **43**

#### bool TrackableFeatures.TrackableCollection< T >.IsSynchronized[get]

См. определение в файле **TrackableCollection.cs** строка **49**

#### object TrackableFeatures.TrackableCollection< T >.SyncRoot[get]

См. определение в файле **TrackableCollection.cs** строка **52**

#### T TrackableFeatures.TrackableCollection< T >.this[int index][get], [set]

См. определение в файле **TrackableCollection.cs** строка **26**

### Cобытия

#### NotifyCollectionChangedEventHandler? TrackableFeatures.TrackableCollection< T >.CollectionChanged

См. определение в файле **TrackableCollection.cs** строка **23**

## Класс TrackableFeatures.TrackableObject

Базовый класс объекта, предоставляющий поддержку отслеживания изменений свойств и ошибок.

Граф наследования:TrackableFeatures.TrackableObject:

### Открытые члены

* IEnumerable **GetErrors** (string? propertyName)

### Защищенные члены

* void **UpdateProperty< T >** (ref T field, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **UpdateProperty< T >** (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action=null, [CallerMemberName] string propertyName="")

*Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.*

* void **AddError** (object error, [CallerMemberName] string propertyName="")

*Добавляет ошибку для свойства.*

* void **ClearErrors** ([CallerMemberName] string propertyName="")

*Очищает все ошибки для свойства.*

* void **ClearAllErrors** ()

*Очищает все ошибки для всех свойств.*

* void **OnPropertyChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***PropertyChanged*** *для свойства.*

* void **OnErrorsChanged** ([CallerMemberName] string propertyName="")

*Вызывает событие* ***ErrorsChanged*** *для свойства.*

### Свойства

* bool **HasErrors** [get]

### События

* PropertyChangedEventHandler? **PropertyChanged**
* EventHandler< DataErrorsChangedEventArgs >? **ErrorsChanged**

### Подробное описание

Базовый класс объекта, предоставляющий поддержку отслеживания изменений свойств и ошибок.

Реализует INotifyPropertyChanged и INotifyDataErrorInfo.

См. определение в файле **TrackableObject.cs** строка **13**

### Методы

#### void TrackableFeatures.TrackableObject.AddError (object error, [CallerMemberName] string propertyName = "")[protected]

Добавляет ошибку для свойства.

##### Аргументы

|  |  |
| --- | --- |
| *error* | Ошибка. |
| *propertyName* | Название свойства. |

См. определение в файле **TrackableObject.cs** строка **95**

#### void TrackableFeatures.TrackableObject.ClearAllErrors ()[protected]

Очищает все ошибки для всех свойств.

См. определение в файле **TrackableObject.cs** строка **132**

#### void TrackableFeatures.TrackableObject.ClearErrors ([CallerMemberName] string propertyName = "")[protected]

Очищает все ошибки для свойства.

##### Аргументы

|  |  |
| --- | --- |
| *propertyName* | Название свойства. |

См. определение в файле **TrackableObject.cs** строка **117**

#### IEnumerable TrackableFeatures.TrackableObject.GetErrors (string? propertyName)

См. определение в файле **TrackableObject.cs** строка **30**

#### void TrackableFeatures.TrackableObject.OnErrorsChanged ([CallerMemberName] string propertyName = "")[protected]

Вызывает событие **ErrorsChanged** для свойства.

##### Аргументы

|  |  |
| --- | --- |
| *propertyName* | Название свойства. |

См. определение в файле **TrackableObject.cs** строка **159**

#### void TrackableFeatures.TrackableObject.OnPropertyChanged ([CallerMemberName] string propertyName = "")[protected]

Вызывает событие **PropertyChanged** для свойства.

##### Аргументы

|  |  |
| --- | --- |
| *propertyName* | Название свойства. |

См. определение в файле **TrackableObject.cs** строка **150**

#### void TrackableFeatures.TrackableObject.UpdateProperty< T > (Func< T > getProperty, Action< T > setProperty, T newValue, Action< T, T >? action = null, [CallerMemberName] string propertyName = "")[protected]

Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип свойства. |

##### Аргументы

|  |  |
| --- | --- |
| *getProperty* | Функция для получения значения свойства. |
| *setProperty* | Действие для установки значения свойства. |
| *newValue* | Новое значение свойства. |
| *action* | Дополнительное действие, выполняемое при изменении свойства. |
| *propertyName* | Название свойства. |

См. определение в файле **TrackableObject.cs** строка **77**

#### void TrackableFeatures.TrackableObject.UpdateProperty< T > (ref T field, T newValue, Action< T, T >? action = null, [CallerMemberName] string propertyName = "")[protected]

Обновляет значение свойства, вызывает действие и уведомляет об изменении свойства.

##### Параметры шаблона

|  |  |
| --- | --- |
| *T* | Тип свойства. |

##### Аргументы

|  |  |
| --- | --- |
| *field* | Ссылка на поле свойства. |
| *newValue* | Новое значение свойства. |
| *action* | Дополнительное действие, выполняемое при изменении свойства. |
| *propertyName* | Название свойства. |

См. определение в файле **TrackableObject.cs** строка **53**

### Полный список свойств

#### bool TrackableFeatures.TrackableObject.HasErrors[get]

См. определение в файле **TrackableObject.cs** строка **21**

### Cобытия

#### EventHandler<DataErrorsChangedEventArgs>? TrackableFeatures.TrackableObject.ErrorsChanged

См. определение в файле **TrackableObject.cs** строка **27**

#### PropertyChangedEventHandler? TrackableFeatures.TrackableObject.PropertyChanged

См. определение в файле **TrackableObject.cs** строка **24**

## Структура MachineLearning.LearningEvaluators.ValidationFold

Структура хранения данных сегмента валидации.

### Открытые члены

* **ValidationFold** (IEnumerable< int > trainIndices, IEnumerable< int > testIndices)

*Создаёт экземпляр класса* ***ValidationFold****.*

### Свойства

* IEnumerable< int > **TrainIndices** [get]

*Возвращает и задаёт индексы данных обучения.*

* IEnumerable< int > **TestIndices** [get]

*Возвращает и задаёт индексы данных тестирования.*

### Подробное описание

Структура хранения данных сегмента валидации.

См. определение в файле **ValidationFold.cs** строка **6**

### Конструктор(ы)

#### MachineLearning.LearningEvaluators.ValidationFold.ValidationFold (IEnumerable< int > trainIndices, IEnumerable< int > testIndices)

Создаёт экземпляр класса **ValidationFold**.

##### Аргументы

|  |  |
| --- | --- |
| *trainIndices* | Индексы данных обучения. |
| *testIndices* | Индексы данных тестирования. |

См. определение в файле **ValidationFold.cs** строка **23**

### Полный список свойств

#### IEnumerable<int> MachineLearning.LearningEvaluators.ValidationFold.TestIndices[get]

Возвращает и задаёт индексы данных тестирования.

См. определение в файле **ValidationFold.cs** строка **16**

#### IEnumerable<int> MachineLearning.LearningEvaluators.ValidationFold.TrainIndices[get]

Возвращает и задаёт индексы данных обучения.

См. определение в файле **ValidationFold.cs** строка **11**

## Класс View.Desktop.Windows.WindowsNotificationManager

Класс менеджера уведомлений **Windows**.

Граф наследования:View.Desktop.Windows.WindowsNotificationManager:

### Открытые члены

* void **SendNotification** (string description, string title)

*Отправляет уведомление.*

##### *Аргументы*

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

### Подробное описание

Класс менеджера уведомлений **Windows**.

Реализует **INotificationManager**.

См. определение в файле **WindowsNotificationManager.cs** строка **13**

### Методы

#### void View.Desktop.Windows.WindowsNotificationManager.SendNotification (string description, string title)

Отправляет уведомление.

##### Аргументы

|  |  |
| --- | --- |
| *description* | Описание. |
| *title* | Заголовок. |

Замещает **ViewModel.Interfaces.AppStates.INotificationManager** (*стр.261*).

См. определение в файле **WindowsNotificationManager.cs** строка **16**

# Алфавитный указатель

INDEX