

# Namespace Application.Dto Classes

[ProfileDto](#)

[ProfileDto.ProfileCreateDto](#)

[ProfileDto.ProfileDetailsDto](#)

[ProfileDto.ProfileSearchDto](#)

DTO для поиска профилей по различным критериям.

[ProfileDto.ProfileShortDto](#)

[ProfileDto.ProfileUpdateDto](#)

[TodoTaskDto](#)

[TodoTaskDto.TodoTaskCreateDto](#)

[TodoTaskDto.TodoTaskDetailsDto](#)

[TodoTaskDto.TodoTaskSearchDto](#)

DTO для поиска задач по различным критериям.

[TodoTaskDto.TodoTaskShortDto](#)

[TodoTaskDto.TodoTaskUpdateDto](#)

# Class ProfileDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

```
public static class ProfileDto
```

## Inheritance

[object](#) ← ProfileDto

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### FromCreateDto(ProfileCreateDto, IPasswordHasher)

```
public static Profile FromCreateDto(this ProfileDto.ProfileCreateDto  
profileCreateDto, IPasswordHasher passwordHashed)
```

## Parameters

profileCreateDto [ProfileDto.ProfileCreateDto](#)

passwordHashed [IPasswordHasher](#)

## Returns

[Profile](#)

### FromUpdateDto(ProfileUpdateDto)

```
public static Profile FromUpdateDto(this ProfileDto.ProfileUpdateDto  
profileUpdateDto)
```

## Parameters

profileUpdateDto [ProfileDto.ProfileUpdateDto](#)

## Returns

[Profile](#)

## ToCreateDto(Profile)

```
public static ProfileDto.ProfileCreateDto ToCreateDto(this Profile profile)
```

## Parameters

profile [Profile](#)

## Returns

[ProfileDto.ProfileCreateDto](#)

## ToDetailsDto(Profile)

```
public static ProfileDto.ProfileDetailsDto ToDetailsDto(this Profile profile)
```

## Parameters

profile [Profile](#)

## Returns

[ProfileDto.ProfileDetailsDto](#)

## ToProfileCriteria(ProfileSearchDto)

Преобразует ProfileSearchDto в ProfileCriteria.

```
public static ProfileCriteria ToProfileCriteria(this ProfileDto.ProfileSearchDto searchDto)
```

## Parameters

searchDto [ProfileDto.ProfileSearchDto](#)

## Returns

[ProfileCriteria](#)

## ToShortDto(Profile)

```
public static ProfileDto.ProfileShortDto ToShortDto(this Profile profile)
```

## Parameters

profile [Profile](#)

## Returns

[ProfileDto.ProfileShortDto](#)

## ToUpdateDto(Profile)

```
public static ProfileDto.ProfileUpdateDto ToUpdateDto(this Profile profile)
```

## Parameters

profile [Profile](#)

## Returns

[ProfileDto.ProfileUpdateDto](#)

# Class ProfileDto.ProfileCreateDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

```
public record ProfileDto.ProfileCreateDto : IEquatable<ProfileDto.ProfileCreateDto>
```

## Inheritance

[object](#) ← ProfileDto.ProfileCreateDto

## Implements

[IEquatable](#)<[ProfileDto.ProfileCreateDto](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[ProfileDto.FromCreateDto\(ProfileDto.ProfileCreateDto, IPasswordHasher\)](#)

## Constructors

### ProfileCreateDto(string, string, DateTime, string)

```
public ProfileCreateDto(string FirstName, string LastName, DateTime DateOfBirth,  
string Password)
```

## Parameters

FirstName [string](#)

LastName [string](#)

DateOfBirth [DateTime](#)

Password [string](#)

# Properties

## DateOfBirth

```
public DateTime DateOfBirth { get; init; }
```

## Property Value

[DateTime](#) ↗

## FirstName

```
public string FirstName { get; init; }
```

## Property Value

[string](#) ↗

## LastName

```
public string LastName { get; init; }
```

## Property Value

[string](#) ↗

## Password

```
public string Password { get; init; }
```

## Property Value

[string](#) ↗

# Class ProfileDto.ProfileDetailsDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

```
public record ProfileDto.ProfileDetailsDto :  
IEquatable<ProfileDto.ProfileDetailsDto>
```

## Inheritance

[object](#) ← ProfileDto.ProfileDetailsDto

## Implements

[IEquatable](#)<[ProfileDto.ProfileDetailsDto](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### ProfileDetailsDto(Guid, string, string, DateTime, DateTime)

```
public ProfileDetailsDto(Guid ProfileId, string FirstName, string LastName, DateTime  
DateOfBirth, DateTime CreatedAt)
```

## Parameters

ProfileId [Guid](#)

FirstName [string](#)

LastName [string](#)

DateOfBirth [DateTime](#)

CreatedAt [DateTime](#)

# Properties

## CreatedAt

```
public DateTime CreatedAt { get; init; }
```

## Property Value

[DateTime ↗](#)

## DateOfBirth

```
public DateTime DateOfBirth { get; init; }
```

## Property Value

[DateTime ↗](#)

## FirstName

```
public string FirstName { get; init; }
```

## Property Value

[string ↗](#)

## LastName

```
public string LastName { get; init; }
```

## Property Value

[string ↗](#)

## ProfileId

```
public Guid ProfileId { get; init; }
```

## Property Value

[Guid](#)

# Class ProfileDto.ProfileSearchDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

DTO для поиска профилей по различным критериям.

```
public record ProfileDto.ProfileSearchDto : IEquatable<ProfileDto.ProfileSearchDto>
```

## Inheritance

[object](#) ← ProfileDto.ProfileSearchDto

## Implements

[IEquatable](#)<[ProfileDto.ProfileSearchDto](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[ProfileDto.ToProfileCriteria\(ProfileDto.ProfileSearchDto\)](#)

## Constructors

ProfileSearchDto(Guid?, string?, string?, DateTime?,  
DateTime?, DateTime?, DateTime?, SearchTypes?)

DTO для поиска профилей по различным критериям.

```
public ProfileSearchDto(Guid? ProfileId = null, string? FirstName = null, string?  
LastName = null, DateTime? CreatedAtFrom = null, DateTime? CreatedAtTo = null,  
DateTime? DateOfBirthFrom = null, DateTime? DateOfBirthTo = null, SearchTypes?  
SearchType = SearchTypes.Contains)
```

## Parameters

ProfileId [Guid](#)?

FirstName [string](#)?

LastName [string](#)?

CreatedAtFrom [DateTime](#)?

CreatedAtTo [DateTime](#)?

DateOfBirthFrom [DateTime](#)?

DateOfBirthTo [DateTime](#)?

SearchType [SearchTypes](#)?

## Properties

### CreatedAtFrom

```
public DateTime? CreatedAtFrom { get; init; }
```

### Property Value

[DateTime](#)?

### CreatedAtTo

```
public DateTime? CreatedAtTo { get; init; }
```

### Property Value

[DateTime](#)?

### DateOfBirthFrom

```
public DateTime? DateOfBirthFrom { get; init; }
```

## PropertyValue

[DateTime](#)?

## DateOfBirthTo

```
public DateTime? DateOfBirthTo { get; init; }
```

## PropertyValue

[DateTime](#)?

## FirstName

```
public string? FirstName { get; init; }
```

## PropertyValue

[string](#)

## LastName

```
public string? LastName { get; init; }
```

## PropertyValue

[string](#)

## ProfileId

```
public Guid? ProfileId { get; init; }
```

## PropertyValue

[Guid](#)?

## SearchType

```
public SearchTypes? SearchType { get; init; }
```

### Property Value

[SearchTypes?](#)

# Class ProfileDto.ProfileShortDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

```
public record ProfileDto.ProfileShortDto : IEquatable<ProfileDto.ProfileShortDto>
```

## Inheritance

[object](#) ← ProfileDto.ProfileShortDto

## Implements

[IEquatable](#)<[ProfileDto.ProfileShortDto](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### ProfileShortDto(Guid, string, string)

```
public ProfileShortDto(Guid ProfileId, string FirstName, string LastName)
```

## Parameters

ProfileId [Guid](#)

FirstName [string](#)

LastName [string](#)

## Properties

### FirstName

```
public string FirstName { get; init; }
```

## PropertyValue

[string](#) ↗

## LastName

```
public string LastName { get; init; }
```

## PropertyValue

[string](#) ↗

## ProfileId

```
public Guid ProfileId { get; init; }
```

## PropertyValue

[Guid](#) ↗

# Class ProfileDto.ProfileUpdateDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

```
public record ProfileDto.ProfileUpdateDto : IEquatable<ProfileDto.ProfileUpdateDto>
```

## Inheritance

[object](#) ← ProfileDto.ProfileUpdateDto

## Implements

[IEquatable](#)<[ProfileDto.ProfileUpdateDto](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[ProfileDto.FromUpdateDto\(ProfileDto.ProfileUpdateDto\)](#)

## Constructors

### ProfileUpdateDto(Guid, string, string, DateTime)

```
public ProfileUpdateDto(Guid ProfileId, string FirstName, string LastName,  
DateTime DateOfBirth)
```

## Parameters

ProfileId [Guid](#)

FirstName [string](#)

LastName [string](#)

DateOfBirth [DateTime](#)

# Properties

## DateOfBirth

```
public DateTime DateOfBirth { get; init; }
```

## Property Value

[DateTime](#)

## FirstName

```
public string FirstName { get; init; }
```

## Property Value

[string](#)

## LastName

```
public string LastName { get; init; }
```

## Property Value

[string](#)

## ProfileId

```
public Guid ProfileId { get; init; }
```

## Property Value

[Guid](#)

# Class TodoTaskDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

```
public static class TodoTaskDto
```

## Inheritance

[object](#) ← TodoTaskDto

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### FromCreateDto(TodoTaskCreateDto, IUserContext)

```
public static TodoTask FromCreateDto(this TodoTaskDto.TodoTaskCreateDto  
todoTaskCreateDto, IUserContext userContext)
```

## Parameters

todoTaskCreateDto [TodoTaskDto.TodoTaskCreateDto](#)

userContext [IUserContext](#)

## Returns

[TodoTask](#)

### FromUpdateDto(TodoTaskUpdateDto)

```
public static TodoTask FromUpdateDto(this TodoTaskDto.TodoTaskUpdateDto  
todoTaskUpdateDto)
```

## Parameters

todoTaskUpdateDto [TodoTaskDto.TodoTaskUpdateDto](#)

## Returns

[TodoTask](#)

## ToCreateDto(TodoTask, IUserContext)

```
public static TodoTaskDto.TodoTaskCreateDto ToCreateDto(this TodoTask todoTask,  
IUserContext userContext)
```

## Parameters

todoTask [TodoTask](#)

userContext [IUserContext](#)

## Returns

[TodoTaskDto.TodoTaskCreateDto](#)

## ToDetailsDto(TodoTask)

```
public static TodoTaskDto.TodoTaskDetailsDto ToDetailsDto(this TodoTask todoTask)
```

## Parameters

todoTask [TodoTask](#)

## Returns

[TodoTaskDto.TodoTaskDetailsDto](#)

## ToShortDto(TodoTask)

```
public static TodoTaskDto.TodoTaskShortDto ToShortDto(this TodoTask todoTask)
```

## Parameters

todoTask [TodoTask](#)

## Returns

[TodoTaskDto](#).[TodoTaskShortDto](#)

## ToTaskCriteria(TodoTaskSearchDto)

Преобразует TodoTaskSearchDto в TaskCriteria.

```
public static TaskCriteria ToTaskCriteria(this TodoTaskDto.TodoTaskSearchDto searchDto)
```

## Parameters

searchDto [TodoTaskDto](#).[TodoTaskSearchDto](#)

## Returns

[TaskCriteria](#)

## ToUpdateDto(TodoTask)

```
public static TodoTaskDto.TodoTaskUpdateDto ToUpdateDto(this TodoTask todoTask)
```

## Parameters

todoTask [TodoTask](#)

## Returns

[TodoTaskDto](#).[TodoTaskUpdateDto](#)



# Class TodoTaskDto.TodoTaskCreateDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

```
public record TodoTaskDto.TodoTaskCreateDto :  
IEquatable<TodoTaskDto.TodoTaskCreateDto>
```

## Inheritance

[object](#) ← TodoTaskDto.TodoTaskCreateDto

## Implements

[IEquatable](#)<[TodoTaskDto.TodoTaskCreateDto](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[TodoTaskDto.FromCreateDto\(TodoTaskDto.TodoTaskCreateDto, IUserContext\)](#)

## Constructors

TodoTaskCreateDto(TaskState?, TaskPriority?, IUserContext, string, string?, DateTime?)

```
public TodoTaskCreateDto(TaskState? State, TaskPriority? Priority, IUserContext  
UserContext, string Name, string? Description, DateTime? Deadline)
```

## Parameters

State [TaskState](#)

Priority [TaskPriority](#)

UserContext [IUserContext](#)

Name [string](#)

Description [string](#)

Deadline [DateTime](#)?

## Properties

### Deadline

```
public DateTime? Deadline { get; init; }
```

### Property Value

[DateTime](#)?

### Description

```
public string? Description { get; init; }
```

### Property Value

[string](#)

### Name

```
public string Name { get; init; }
```

### Property Value

[string](#)

### Priority

```
public TaskPriority? Priority { get; init; }
```

## Property Value

[TaskPriority](#)

## State

```
public TaskState? State { get; init; }
```

## Property Value

[TaskState](#)

## UserContext

```
public IUserContext UserContext { get; init; }
```

## Property Value

[IUserContext](#)

# Class TodoTaskDto.TodoTaskDetailsDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

```
public record TodoTaskDto.TodoTaskDetailsDto :  
IEquatable<TodoTaskDto.TodoTaskDetailsDto>
```

## Inheritance

[object](#) ← TodoTaskDto.TodoTaskDetailsDto

## Implements

[IEquatable](#)<[TodoTaskDto.TodoTaskDetailsDto](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

TodoTaskDetailsDto(Guid, string, string, string, string, string?, DateTime, DateTime?)

```
public TodoTaskDetailsDto(Guid TaskId, string NameState, string DescriptionState,  
string NamePriority, string Name, string? Description, DateTime CreatedAt,  
DateTime? Deadline)
```

## Parameters

TaskId [Guid](#)

NameState [string](#)

DescriptionState [string](#)

NamePriority [string](#)

Name [string](#)

Description [string](#)

CreatedAt [DateTime](#)

Deadline [DateTime](#)?

## Properties

### CreatedAt

```
public DateTime CreatedAt { get; init; }
```

### Property Value

[DateTime](#)

### Deadline

```
public DateTime? Deadline { get; init; }
```

### Property Value

[DateTime](#)?

### Description

```
public string? Description { get; init; }
```

### Property Value

[string](#)

## DescriptionState

```
public string DescriptionState { get; init; }
```

### Property Value

[string ↗](#)

## Name

```
public string Name { get; init; }
```

### Property Value

[string ↗](#)

## NamePriority

```
public string NamePriority { get; init; }
```

### Property Value

[string ↗](#)

## NameState

```
public string NameState { get; init; }
```

### Property Value

[string ↗](#)

## TaskId

```
public Guid TaskId { get; init; }
```

## Property Value

[Guid](#) ↗

# Class TodoTaskDto.TodoTaskSearchDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

DTO для поиска задач по различным критериям.

```
public record TodoTaskDto.TodoTaskSearchDto :  
IEquatable<TodoTaskDto.TodoTaskSearchDto>
```

**Inheritance**

[object](#) ← TodoTaskDto.TodoTaskSearchDto

**Implements**

[IEquatable](#)<[TodoTaskDto.TodoTaskSearchDto](#)>

**Inherited Members**

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

**Extension Methods**

[TodoTaskDto.ToTaskCriteria\(TodoTaskDto.TodoTaskSearchDto\)](#)

## Constructors

TodoTaskSearchDto(IUserContext, Guid?, int?, int?, int?,  
string?, string?, DateTime?, DateTime?, DateTime?,  
DateTime?, SearchTypes?)

DTO для поиска задач по различным критериям.

```
public TodoTaskSearchDto(IUserContext UserContext, Guid? TaskId = null, int? StateId  
= null, int? PriorityLevelFrom = null, int? PriorityLevelTo = null, string? Name =  
null, string? Description = null, DateTime? CreatedAtFrom = null, DateTime?  
CreatedAtTo = null, DateTime? DeadlineFrom = null, DateTime? DeadlineTo = null,  
SearchTypes? SearchType = SearchTypes.Contains)
```

## Parameters

UserContext [IUserContext](#)

TaskId [Guid](#)?

StateId [int](#)?

PriorityLevelFrom [int](#)?

PriorityLevelTo [int](#)?

Name [string](#)

Description [string](#)

CreatedAtFrom [DateTime](#)?

CreatedAtTo [DateTime](#)?

DeadlineFrom [DateTime](#)?

DeadlineTo [DateTime](#)?

SearchType [SearchTypes](#)?

## Properties

### CreatedAtFrom

```
public DateTime? CreatedAtFrom { get; init; }
```

### Property Value

[DateTime](#)?

### CreatedAtTo

```
public DateTime? CreatedAtTo { get; init; }
```

## PropertyValue

[DateTime](#)?

## DeadlineFrom

```
public DateTime? DeadlineFrom { get; init; }
```

## PropertyValue

[DateTime](#)?

## DeadlineTo

```
public DateTime? DeadlineTo { get; init; }
```

## PropertyValue

[DateTime](#)?

## Description

```
public string? Description { get; init; }
```

## PropertyValue

[string](#)?

## Name

```
public string? Name { get; init; }
```

## PropertyValue

[string](#)?

## PriorityLevelFrom

`public int? PriorityLevelFrom { get; init; }`

### Property Value

[int](#)?

## PriorityLevelTo

`public int? PriorityLevelTo { get; init; }`

### Property Value

[int](#)?

## SearchType

`public SearchTypes? SearchType { get; init; }`

### Property Value

[SearchTypes](#)?

## StateId

`public int? StateId { get; init; }`

### Property Value

[int](#)?

## TaskId

```
public Guid? TaskId { get; init; }
```

## Property Value

[Guid](#)?

## UserContext

```
public IUserContext UserContext { get; init; }
```

## Property Value

[IUserContext](#)

# Class TodoTaskDto.TodoTaskShortDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

```
public record TodoTaskDto.TodoTaskShortDto :  
IEquatable<TodoTaskDto.TodoTaskShortDto>
```

## Inheritance

[object](#) ← TodoTaskDto.TodoTaskShortDto

## Implements

[IEquatable](#)<[TodoTaskDto.TodoTaskShortDto](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

TodoTaskShortDto(Guid, string, DateTime?)

```
public TodoTaskShortDto(Guid TaskId, string Name, DateTime? Deadline)
```

## Parameters

TaskId [Guid](#)

Name [string](#)

Deadline [DateTime](#)?

## Properties

Deadline

```
public DateTime? Deadline { get; init; }
```

## Property Value

[DateTime](#)?

## Name

```
public string Name { get; init; }
```

## Property Value

[string](#)

## TaskId

```
public Guid TaskId { get; init; }
```

## Property Value

[Guid](#)

# Class TodoTaskDto.TodoTaskUpdateDto

Namespace: [Application.Dto](#)

Assembly: Application.dll

```
public record TodoTaskDto.TodoTaskUpdateDto :  
IEquatable<TodoTaskDto.TodoTaskUpdateDto>
```

## Inheritance

[object](#) ← TodoTaskDto.TodoTaskUpdateDto

## Implements

[IEquatable](#)<[TodoTaskDto.TodoTaskUpdateDto](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[TodoTaskDto.FromUpdateDto\(TodoTaskDto.TodoTaskUpdateDto\)](#)

## Constructors

TodoTaskUpdateDto(Guid, TaskState, TaskPriority, string, string?, DateTime?)

```
public TodoTaskUpdateDto(Guid TaskId, TaskState State, TaskPriority Priority, string  
Name, string? Description, DateTime? Deadline)
```

## Parameters

TaskId [Guid](#)

State [TaskState](#)

Priority [TaskPriority](#)

Name [string](#)

Description [string](#)

Deadline [DateTime](#)?

## Properties

### Deadline

```
public DateTime? Deadline { get; init; }
```

### Property Value

[DateTime](#)?

### Description

```
public string? Description { get; init; }
```

### Property Value

[string](#)

### Name

```
public string Name { get; init; }
```

### Property Value

[string](#)

### Priority

```
public TaskPriority Priority { get; init; }
```

## Property Value

[TaskPriority](#)

## State

```
public TaskState State { get; init; }
```

## Property Value

[TaskState](#)

## TaskId

```
public Guid TaskId { get; init; }
```

## Property Value

[Guid](#)

# Namespace Application.Interfaces

## Interfaces

[ICommandManager](#)

[IPasswordHasher](#)

[IUserContext](#)

# Interface ICommandManager

Namespace: [Application.Interfaces](#)

Assembly: Application.dll

```
public interface ICommandManager
```

## Methods

### ExecuteCommandAsync(ICommandWithUndo)

```
Task<int> ExecuteCommandAsync(ICommandWithUndo command)
```

#### Parameters

command [ICommandWithUndo](#)

#### Returns

[Task](#)<int>

### Redo()

```
Task<int> Redo()
```

#### Returns

[Task](#)<int>

### Undo()

```
Task<int> Undo()
```

## Returns

[Task](#)<[int](#)>

# Interface IPasswordHasher

Namespace: [Application.Interfaces](#)

Assembly: Application.dll

```
public interface IPasswordHasher
```

## Methods

### HashedAsync(string)

```
Task<string> HashedAsync(string password)
```

#### Parameters

password [string](#)

#### Returns

[Task](#)<[string](#)>

### VerifyAsync(string, string)

```
Task<bool> VerifyAsync(string password, string hashedPassword)
```

#### Parameters

password [string](#)

hashedPassword [string](#)

#### Returns

[Task](#)<[bool](#)>

# Interface IUserContext

Namespace: [Application.Interfaces](#)

Assembly: Application.dll

```
public interface IUserContext
```

## Properties

### UserId

```
Guid? UserId { get; }
```

## Property Value

[Guid](#)?

## Methods

### Clear()

```
void Clear()
```

### Set(Guid?)

```
void Set(Guid? userId)
```

## Parameters

userId [Guid](#)?

# Namespace Application.Interfaces.

## Command

### Interfaces

#### [ICommandUseCase](#)

Интерфейс для command use case (добавление, обновление, удаление). Command операции изменяют состояние системы и возвращают количество затронутых записей.

#### [ICommandWithUndo](#)

Интерфейс для command use case с поддержкой отмены операции.

#### [IOperation<TResult>](#)

Главный интерфейс для всех операций (команд и запросов) в приложении. Представляет абсолютно все use cases - как query, так и command операции.

#### [IQueryUseCase<TResult>](#)

Интерфейс для query use case (поиск, получение данных). Query операции не изменяют состояние системы.

# Interface ICommandUseCase

Namespace: [Application.Interfaces.Command](#)

Assembly: Application.dll

Интерфейс для command use case (добавление, обновление, удаление). Command операции изменяют состояние системы и возвращают количество затронутых записей.

```
public interface ICommandUseCase : IOperation<int>
```

## Inherited Members

[IOperation<int>.Execute\(\)](#)

# Interface ICommandWithUndo

Namespace: [Application.Interfaces.Command](#)

Assembly: Application.dll

Интерфейс для command use case с поддержкой отмены операции.

```
public interface ICommandWithUndo : ICommandUseCase, IOperation<int>
```

## Inherited Members

[IOperation<int>.Execute\(\)](#)

## Methods

### Undo()

Отменяет выполненную операцию.

```
Task<int> Undo()
```

## Returns

[Task<int>](#)

Количество затронутых записей при отмене.

# Interface IOperation<TResult>

Namespace: [Application.Interfaces.Command](#)

Assembly: Application.dll

Главный интерфейс для всех операций (команд и запросов) в приложении. Представляет абсолютно все use cases - как query, так и command операции.

```
public interface IOperation<TResult>
```

## Type Parameters

### TResult

Тип возвращаемого результата операции.

## Methods

### Execute()

Выполняет операцию и возвращает результат.

```
Task<TResult> Execute()
```

## Returns

### [Task](#)<TResult>

Результат выполнения операции типа TResult.

# Interface IQueryUseCase<TResult>

Namespace: [Application.Interfaces.Command](#)

Assembly: Application.dll

Интерфейс для query use case (поиск, получение данных). Query операции не изменяют состояние системы.

```
public interface IQueryUseCase<TResult> : IOperation<TResult>
```

## Type Parameters

### TResult

Тип возвращаемого результата.

### Inherited Members

[IOperation<TResult>.Execute\(\)](#)

# Namespace Application.Interfaces. Repository

## Interfaces

[IBaseRepository<T>](#)

[IFilterByCriteria< TValue, TCriteria >](#)

[IProfileRepository](#)

[ITodoTaskRepository](#)

# Interface IBaseRepository<T>

Namespace: [Application.Interfaces.Repository](#)

Assembly: Application.dll

```
public interface IBaseRepository<T>
```

## Type Parameters

T

## Methods

### AddAsync(T)

```
Task<int> AddAsync(T obj)
```

#### Parameters

obj T

#### Returns

[Task](#)<[int](#)>

### DeleteAsync(Guid)

```
Task<int> DeleteAsync(Guid id)
```

#### Parameters

id [Guid](#)

#### Returns

[Task](#)<[int](#)>

## GetByIdAsync(Guid)

Task<T?> GetByIdAsync(Guid id)

### Parameters

**id** [Guid](#)

### Returns

[Task](#)<T>

## UpdateAsync(T)

Task<[int](#)> UpdateAsync(T obj)

### Parameters

**obj** T

### Returns

[Task](#)<[int](#)>

# Interface IFilterByCriteria< TValue, TCriteria >

Namespace: [Application.Interfaces.Repository](#)

Assembly: Application.dll

```
public interface IFilterByCriteria<TValue, TCriteria>
```

## Type Parameters

TValue

TCriteria

## Methods

### CountAsync(TCriteria)

```
Task<int> CountAsync(TCriteria profileCriteria)
```

#### Parameters

profileCriteria TCriteria

#### Returns

[Task](#)<[int](#)>

### ExistsAsync(TCriteria)

```
Task<bool> ExistsAsync(TCriteria profileCriteria)
```

#### Parameters

profileCriteria TCriteria

## Returns

[Task](#)<[bool](#)>

## FindAsync(TCriteria)

Task<IEnumerable<TValue>> FindAsync(TCriteria profileCriteria)

## Parameters

**profileCriteria** TCriteria

## Returns

[Task](#)<[IEnumerable](#)<TValue>>

## FindSingleAsync(TCriteria)

Task<TValue?> FindSingleAsync(TCriteria profileCriteria)

## Parameters

**profileCriteria** TCriteria

## Returns

[Task](#)<TValue>

# Interface IProfileRepository

Namespace: [Application.Interfaces.Repository](#)

Assembly: Application.dll

```
public interface IProfileRepository : IBaseRepository<Profile>,
IFilterByCriteria<Profile, ProfileCriteria>
```

## Inherited Members

[IBaseRepository<Profile>.AddAsync\(Profile\)](#) , [IBaseRepository<Profile>.UpdateAsync\(Profile\)](#) ,  
[IBaseRepository<Profile>.DeleteAsync\(Guid\)](#) , [IBaseRepository<Profile>.GetByIdAsync\(Guid\)](#) ,  
[IFilterByCriteria<Profile, ProfileCriteria>.FindAsync\(ProfileCriteria\)](#) ,  
[IFilterByCriteria<Profile, ProfileCriteria>.FindSingleAsync\(ProfileCriteria\)](#) ,  
[IFilterByCriteria<Profile, ProfileCriteria>.ExistsAsync\(ProfileCriteria\)](#) ,  
[IFilterByCriteria<Profile, ProfileCriteria>.CountAsync\(ProfileCriteria\)](#).

## Methods

### GetAllAsync()

```
Task<IEnumerable<Profile>> GetAllAsync()
```

## Returns

[Task<IEnumerable<Profile>>](#)

# Interface ITodoTaskRepository

Namespace: [Application.Interfaces.Repository](#)

Assembly: Application.dll

```
public interface ITodoTaskRepository : IBaseRepository<TodoTask>,
IFilterByCriteria<TodoTask, TaskCriteria>
```

## Inherited Members

[IBaseRepository<TodoTask>.AddAsync\(TodoTask\)](#) ,  
[IBaseRepository<TodoTask>.UpdateAsync\(TodoTask\)](#) ,  
[IBaseRepository<TodoTask>.DeleteAsync\(Guid\)](#) , [IBaseRepository<TodoTask>.GetByIdAsync\(Guid\)](#) ,  
[IFilterByCriteria<TodoTask, TaskCriteria>.FindAsync\(TaskCriteria\)](#) ,  
[IFilterByCriteria<TodoTask, TaskCriteria>.FindSingleAsync\(TaskCriteria\)](#) ,  
[IFilterByCriteria<TodoTask, TaskCriteria>.ExistsAsync\(TaskCriteria\)](#) ,  
[IFilterByCriteria<TodoTask, TaskCriteria>.CountAsync\(TaskCriteria\)](#)

## Methods

### GetAllAsync(IUserContext)

```
Task<IEnumerable<TodoTask>> GetAllAsync(IUserContext userContext)
```

## Parameters

userContext [IUserContext](#)

## Returns

[Task<IEnumerable<TodoTask>>](#)

# Namespace Application.Specifications

## Classes

### [ProfileCriteria](#)

Содержит критерии для фильтрации профилей пользователей. Поддерживает фильтрацию по ID, имени, фамилии и датам.

### [RangeObj<T>](#)

Представляет диапазон значений с от границей и до границей.

### [TaskCriteria](#)

Содержит критерии для фильтрации задач. Поддерживает фильтрацию по ID, статусу, приоритету, названию, описанию и датам.

# Class ProfileCriteria

Namespace: [Application.Specifications](#)

Assembly: Application.dll

Содержит критерии для фильтрации профилей пользователей. Поддерживает фильтрацию по ID, имени, фамилии и датам.

```
public class ProfileCriteria
```

## Inheritance

[object](#) ← ProfileCriteria

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Properties

## CreatedAt

Критерий фильтрации по дате создания профиля.

```
public CriteriaRangeObj<RangeObj<DateTime>, DateTime?> CreatedAt { get; init; }
```

## Property Value

[CriteriaRangeObj<RangeObj<DateTime>, DateTime>](#)

## DateOfBirth

Критерий фильтрации по дате рождения.

```
public CriteriaRangeObj<RangeObj<DateTime>, DateTime?> DateOfBirth { get; init; }
```

## Property Value

[CriteriaRangeObj<RangeObj<DateTime>, DateTime>](#)

## FirstName

Критерий фильтрации по имени пользователя.

```
public CriteriaObj<string>? FirstName { get; init; }
```

## Property Value

[CriteriaObj<string>](#)

## LastName

Критерий фильтрации по фамилии пользователя.

```
public CriteriaObj<string>? LastName { get; init; }
```

## Property Value

[CriteriaObj<string>](#)

## ProfileId

Критерий фильтрации по ID профиля.

```
public CriteriaId<Guid>? ProfileId { get; init; }
```

## Property Value

[CriteriaId<Guid>](#)

## Methods

## Add(ProfileCriteria)

Объединяет текущие критерии с другими критериями, приоритет имеют текущие критерии.

```
public ProfileCriteria Add(ProfileCriteria other)
```

### Parameters

**other** [ProfileCriteria](#)

Другие критерии для объединения.

### Returns

[ProfileCriteria](#)

Новый объект ProfileCriteria с объединенными критериями.

## ByCreatedAt(DateTime?, DateTime?)

Создает критерий фильтрации профилей по диапазону дат создания.

```
public static ProfileCriteria ByCreatedAt(DateTime? from, DateTime? to)
```

### Parameters

**from** [DateTime?](#)

Начальная дата. Null означает отсутствие нижней границы.

**to** [DateTime?](#)

Конечная дата. Null означает отсутствие верхней границы.

### Returns

[ProfileCriteria](#)

Новый объект ProfileCriteria с установленным критерием даты создания.

## ByDateOfBirth(DateTime?, DateTime?)

Создает критерий фильтрации профилей по диапазону дат рождения.

```
public static ProfileCriteria ByDateOfBirth(DateTime? from, DateTime? to)
```

### Parameters

**from** [DateTime](#)?

Начальная дата. Null означает отсутствие нижней границы.

**to** [DateTime](#)?

Конечная дата. Null означает отсутствие верхней границы.

### Returns

[ProfileCriteria](#)

Новый объект ProfileCriteria с установленным критерием даты рождения.

## ByFirstName(string)

Создает критерий фильтрации профилей по имени.

```
public static ProfileCriteria ByFirstName(string firstName)
```

### Parameters

**firstName** [string](#)

Имя для поиска.

### Returns

[ProfileCriteria](#)

Новый объект ProfileCriteria с установленным критерием имени.

## ByLastName(string)

Создает критерий фильтрации профилей по фамилии.

```
public static ProfileCriteria ByLastName(string lastName)
```

### Parameters

`lastName` [string](#)

Фамилия для поиска.

### Returns

[ProfileCriteria](#)

Новый объект ProfileCriteria с установленным критерием фамилии.

## ByProfileId(Guid)

Создает критерий фильтрации профиля по ID.

```
public static ProfileCriteria ByProfileId(Guid profileId)
```

### Parameters

`profileId` [Guid](#)

ID профиля для поиска.

### Returns

[ProfileCriteria](#)

Новый объект ProfileCriteria с установленным критерием ID.

## Contains()

Устанавливает тип сравнения на "Содержит" для текстовых критериев.

```
public ProfileCriteria Contains()
```

## Returns

[ProfileCriteria](#)

Новый объект ProfileCriteria с критериями поиска по вхождению подстроки.

## EndWith()

Устанавливает тип сравнения на "Заканчивается на" для текстовых критериев.

```
public ProfileCriteria EndWith()
```

## Returns

[ProfileCriteria](#)

Новый объект ProfileCriteria с критериями поиска по окончанию подстроки.

## Equals()

Устанавливает тип сравнения на "Точное совпадение" для текстовых критериев.

```
public ProfileCriteria Equals()
```

## Returns

[ProfileCriteria](#)

Новый объект ProfileCriteria с критериями поиска по точному совпадению.

## Not()

Инвертирует все критерии поиска.

```
public ProfileCriteria Not()
```

## Returns

### [ProfileCriteria](#)

Новый объект ProfileCriteria с инвертированными критериями.

## StartsWith()

Устанавливает тип сравнения на "Начинается с" для текстовых критериев.

```
public ProfileCriteria StartsWith()
```

## Returns

### [ProfileCriteria](#)

Новый объект ProfileCriteria с критериями поиска по началу подстроки.

## Operators

### operator +(ProfileCriteria, ProfileCriteria)

Оператор сложения для объединения двух критериев профилей.

```
public static ProfileCriteria operator +(ProfileCriteria left,  
ProfileCriteria right)
```

## Parameters

### **left** [ProfileCriteria](#)

Левый операнд (имеет приоритет).

### **right** [ProfileCriteria](#)

Правый операнд.

## Returns

### [ProfileCriteria](#)

Новый объект ProfileCriteria с объединенными критериями.

# Class RangeObj<T>

Namespace: [Application.Specifications](#)

Assembly: Application.dll

Представляет диапазон значений с от границей и до границей.

```
public class RangeObj<T> : IComparable<T> where T : struct, IComparable<T>
```

## Type Parameters

T

Тип значений диапазона. Должен быть структурой, реализующей IComparable<T>

### Inheritance

[object](#) ← RangeObj<T>

### Implements

[IComparable](#)<T>

### Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### RangeObj(T?, T?)

Инициализирует новый экземпляр класса RangeObj.

```
public RangeObj(T? from, T? to)
```

## Parameters

from T?

Начало диапазона.

**to** T?

Конец диапазона.

## Exceptions

[Exception ↗](#)

Выбрасывается, если from больше to.

## Properties

### From

Начало диапазона (минимальное значение). Null означает отсутствие нижней границы.

```
public T? From { get; }
```

### Property Value

T?

### To

Конец диапазона (максимальное значение). Null означает отсутствие верхней границы.

```
public T? To { get; }
```

### Property Value

T?

## Methods

### CompareTo(T)

Сравнивает значение с диапазоном.

```
public int CompareTo(T other)
```

## Parameters

**other** T

Значение для сравнения.

## Returns

[int](#)

Возвращает 0, если значение находится в диапазоне; отрицательное значение, если оно меньше от границы; положительное значение, если оно больше до границы.

## Contains(T)

Проверяет, находится ли значение в пределах диапазона.

```
public bool Contains(T value)
```

## Parameters

**value** T

Значение для проверки.

## Returns

[bool](#)

Возвращает true, если значение находится в диапазоне; иначе false.

# Class TaskCriteria

Namespace: [Application.Specifications](#)

Assembly: Application.dll

Содержит критерии для фильтрации задач. Поддерживает фильтрацию по ID, статусу, приоритету, названию, описанию и датам.

```
public class TaskCriteria
```

## Inheritance

[object](#) ← TaskCriteria

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Properties

## CreatedAt

Критерий фильтрации по дате создания задачи.

```
public CriteriaRangeObj<RangeObj<DateTime>, DateTime?> CreatedAt { get; init; }
```

## Property Value

[CriteriaRangeObj<RangeObj<DateTime>, DateTime>](#)

## Deadline

Критерий фильтрации по сроку выполнения задачи.

```
public CriteriaRangeObj<RangeObj<DateTime>, DateTime?> Deadline { get; init; }
```

## Property Value

[CriteriaRangeObj<RangeObj<DateTime>, DateTime>](#)

## Description

Критерий фильтрации по описанию задачи.

```
public CriteriaObj<string>? Description { get; init; }
```

## Property Value

[CriteriaObj<string>](#)

## Name

Критерий фильтрации по названию задачи.

```
public CriteriaObj<string>? Name { get; init; }
```

## Property Value

[CriteriaObj<string>](#)

## PriorityLevel

Критерий фильтрации по уровню приоритета задачи.

```
public CriteriaRangeObj<RangeObj<int>, int>? PriorityLevel { get; init; }
```

## Property Value

[CriteriaRangeObj<RangeObj<int>, int>](#)

## ProfileId

Критерий фильтрации по ID профиля владельца задачи.

```
public CriteriaId<Guid>? ProfileId { get; init; }
```

## Property Value

[CriteriaId<Guid>](#)

## StateId

Критерий фильтрации по ID состояния задачи.

```
public CriteriaId<int>? StateId { get; init; }
```

## Property Value

[CriteriaId<int>](#)

## TaskId

Критерий фильтрации по ID задачи.

```
public CriteriaId<Guid>? TaskId { get; init; }
```

## Property Value

[CriteriaId<Guid>](#)

## Methods

### Add(TaskCriteria)

Объединяет текущие критерии с другими критериями, приоритет имеют текущие критерии.

```
public TaskCriteria Add(TaskCriteria other)
```

## Parameters

**other** [TaskCriteria](#)

Другие критерии для объединения.

## Returns

[TaskCriteria](#)

Новый объект TaskCriteria с объединенными критериями.

## ByCreatedAt(DateTime?, DateTime?)

Создает критерий фильтрации задач по диапазону дат создания.

```
public static TaskCriteria ByCreatedAt(DateTime? from, DateTime? to)
```

## Parameters

**from** [DateTime?](#)

Начальная дата. Null означает отсутствие нижней границы.

**to** [DateTime?](#)

Конечная дата. Null означает отсутствие верхней границы.

## Returns

[TaskCriteria](#)

Новый объект TaskCriteria с установленным критерием даты создания.

## ByDeadline(DateTime?, DateTime?)

Создает критерий фильтрации задач по диапазону сроков выполнения.

```
public static TaskCriteria ByDeadline(DateTime? from, DateTime? to)
```

## Parameters

`from` [DateTime](#)?

Начальная дата. Null означает отсутствие нижней границы.

`to` [DateTime](#)?

Конечная дата. Null означает отсутствие верхней границы.

## Returns

[TaskCriteria](#)

Новый объект TaskCriteria с установленным критерием срока выполнения.

## ByDescription(string)

Создает критерий фильтрации задач по описанию.

```
public static TaskCriteria ByDescription(string description)
```

## Parameters

`description` [string](#)?

Описание для поиска.

## Returns

[TaskCriteria](#)

Новый объект TaskCriteria с установленным критерием описания.

## ByName(string)

Создает критерий фильтрации задач по названию.

```
public static TaskCriteria ByName(string name)
```

## Parameters

`name string`

Название для поиска.

## Returns

[TaskCriteria](#)

Новый объект TaskCriteria с установленным критерием названия.

## ByPriorityLevel(int?, int?)

Создает критерий фильтрации задач по диапазону уровня приоритета.

```
public static TaskCriteria ByPriorityLevel(int? from, int? to)
```

## Parameters

`from int?`

Минимальный уровень приоритета. Null означает отсутствие нижней границы.

`to int?`

Максимальный уровень приоритета. Null означает отсутствие верхней границы.

## Returns

[TaskCriteria](#)

Новый объект TaskCriteria с установленным критерием приоритета.

## ByProfileId(Guid)

Создает критерий фильтрации задач по ID профиля владельца.

```
public static TaskCriteria ByProfileId(Guid profileId)
```

## Parameters

`profileId` [Guid](#)

ID профиля для поиска.

## Returns

[TaskCriteria](#)

Новый объект TaskCriteria с установленным критерием профиля.

## ByStateId(int)

Создает критерий фильтрации задач по ID состояния.

```
public static TaskCriteria ByStateId(int stateId)
```

## Parameters

`stateId` [int](#)

ID состояния для поиска.

## Returns

[TaskCriteria](#)

Новый объект TaskCriteria с установленным критерием состояния.

## ByTaskId(Guid)

Создает критерий фильтрации задачи по ID.

```
public static TaskCriteria ByTaskId(Guid taskId)
```

## Parameters

`taskId` [Guid](#)

ID задачи для поиска.

## Returns

### [TaskCriteria](#)

Новый объект TaskCriteria с установленным критерием ID.

## Contains()

Устанавливает тип сравнения на "Содержит" для текстовых критериев.

```
public TaskCriteria Contains()
```

## Returns

### [TaskCriteria](#)

Новый объект TaskCriteria с критериями поиска по вхождению подстроки.

## EndWith()

Устанавливает тип сравнения на "Заканчивается на" для текстовых критериев.

```
public TaskCriteria EndWith()
```

## Returns

### [TaskCriteria](#)

Новый объект TaskCriteria с критериями поиска по окончанию подстроки.

## Equals()

Устанавливает тип сравнения на "Точное совпадение" для текстовых критериев.

```
public TaskCriteria Equals()
```

## Returns

### [TaskCriteria](#)

Новый объект TaskCriteria с критериями поиска по точному совпадению.

## Not()

Инвертирует все критерии поиска.

```
public TaskCriteria Not()
```

## Returns

### [TaskCriteria](#)

Новый объект TaskCriteria с инвертированными критериями.

## StartsWith()

Устанавливает тип сравнения на "Начинается с" для текстовых критериев.

```
public TaskCriteria StartsWith()
```

## Returns

### [TaskCriteria](#)

Новый объект TaskCriteria с критериями поиска по началу подстроки.

## Operators

### operator +(TaskCriteria, TaskCriteria)

Оператор сложения для объединения двух критериев задач.

```
public static TaskCriteria operator +(TaskCriteria left, TaskCriteria right)
```

## Parameters

### **left** [TaskCriteria](#)

Левый операнд (имеет приоритет).

### **right** [TaskCriteria](#)

Правый операнд.

## Returns

### [TaskCriteria](#)

Новый объект TaskCriteria с объединенными критериями.

# Namespace Application.Specifications.Criterias

## Classes

### [CriteriaId<T>](#)

Критерий для фильтрации по точному совпадению значения. Строит выражение `Expression Func TEntity, bool` по селектору свойства.

### [CriteriaObj<T>](#)

Критерий для фильтрации по значению объекта с поддержкой различных типов сравнения. Строит выражение `Expression Func TEntity, bool` по селектору свойства.

### [CriteriaRangeObj<TRange, TInRange>](#)

Критерий для фильтрации по диапазону значений. Строит выражение `Expression Func TEntity, bool` по селектору свойства.

## Enums

### [SearchTypes](#)

# Class CriteriaId<T>

Namespace: [Application.Specifications.Criteria](#)

Assembly: Application.dll

Критерий для фильтрации по точному совпадению значения. Строит выражение Expression Func TEntity, bool по селектору свойства.

```
public class CriteriaId<T> where T : IComparable<T>
```

## Type Parameters

T

Тип значения для сравнения, должен реализовать интерфейс IComparable<T>

### Inheritance

[object](#) ← CriteriaId<T>

### Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

### Extension Methods

[CriteriaToExpression.IsSatisfiedBy< TEntity, T >\(CriteriaId<T>, Expression< Func< TEntity, T >>\)](#)

## Constructors

### CriteriaId(T)

Критерий для фильтрации по точному совпадению значения. Строит выражение Expression Func TEntity, bool по селектору свойства.

```
public CriteriaId(T value)
```

## Parameters

**value** T

## Fields

### Value

```
public readonly T Value
```

### Field Value

T

## Properties

### NotFlag

```
public bool NotFlag { get; }
```

### Property Value

[bool](#) ↗

## Methods

### Not()

```
public CriteriaId<T> Not()
```

### Returns

[CriteriaId](#)<T>

# Class CriteriaObj<T>

Namespace: [Application.Specifications.Criteria](#)

Assembly: Application.dll

Критерий для фильтрации по значению объекта с поддержкой различных типов сравнения.  
Строит выражение `Expression Func TEntity, bool` по селектору свойства.

```
public class CriteriaObj<T> where T : IComparable<T>
```

## Type Parameters

T

Тип значения для сравнения, должен реализовать интерфейс `IComparable<T>`

## Inheritance

[object](#) ← CriteriaObj<T>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[CriteriaToExpression.IsSatisfiedBy< TEntity, T >\(CriteriaObj<T>, Expression<Func< TEntity, T >>\)](#)

## Constructors

### CriteriaObj(T)

Критерий для фильтрации по значению объекта с поддержкой различных типов сравнения.  
Строит выражение `Expression Func TEntity, bool` по селектору свойства.

```
public CriteriaObj(T value)
```

## Parameters

value T

## Fields

### CompareType

`public SearchTypes CompareType`

#### Field Value

[SearchTypes](#)

#### Value

`public readonly T Value`

#### Field Value

T

## Properties

### NotFlag

`public bool NotFlag { get; }`

#### Property Value

[bool](#) ↗

## Methods

### Contains()

```
public CriteriaObj<T> Contains()
```

Returns

[CriteriaObj<T>](#)

## EndWith()

```
public CriteriaObj<T> EndWith()
```

Returns

[CriteriaObj<T>](#)

## Equals()

```
public CriteriaObj<T> Equals()
```

Returns

[CriteriaObj<T>](#)

## Not()

```
public CriteriaObj<T> Not()
```

Returns

[CriteriaObj<T>](#)

## StartsWith()

```
public CriteriaObj<T> Startswith()
```

## Returns

[CriteriaObj<T>](#)

# Class CriteriaRangeObj<TRange, TInRange>

Namespace: [Application.Specifications.Criteria](#)

Assembly: Application.dll

Критерий для фильтрации по диапазону значений. Строит выражение `Expression Func TEntity, bool` по селектору свойства.

```
public class CriteriaRangeObj<TRange, TInRange> where TRange : IComparable<TInRange>
```

## Type Parameters

### TRange

Тип диапазона, должен реализовать интерфейс `IComparable<TInRange>`

### TInRange

Тип значения для сравнения с диапазоном.

### Inheritance

[object](#) ← CriteriaRangeObj<TRange, TInRange>

### Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

### Extension Methods

[CriteriaToExpression.IsSatisfiedBy< TEntity, TRange, TInRange >\(CriteriaRangeObj< TRange, TInRange >, Expression< Func< TEntity, TInRange > \)](#)

## Constructors

### CriteriaRangeObj(TRange)

Критерий для фильтрации по диапазону значений. Строит выражение `Expression Func TEntity, bool` по селектору свойства.

```
public CriteriaRangeObj(TRange value)
```

## Parameters

**value** TRange

## Fields

### Value

```
public readonly TRange Value
```

### Field Value

TRange

## Properties

### NotFlag

```
public bool NotFlag { get; }
```

### Property Value

[bool](#)

## Methods

### Not()

```
public CriteriaRangeObj<TRange, TInRange> Not()
```

## Returns

## CriteriaRangeObj<TRange, TInRange>

# Enum SearchTypes

Namespace: [Application.Specifications.Criteria](#)

Assembly: Application.dll

```
public enum SearchTypes
```

## Fields

Contains = 0

EndsWith = 3

Equals = 2

StartsWith = 1

# Namespace Application.UseCase.ProfileUseCases

## Classes

[AddNewProfileUseCase](#)

[ChangeProfileUseCase](#)

[DeletionProfileUseCase](#)

[UpdateProfileUseCase](#)

# Class AddNewProfileUseCase

Namespace: [Application.UseCase.ProfileUseCases](#)

Assembly: Application.dll

```
public class AddNewProfileUseCase : ICommandWithUndo, ICommandUseCase,  
IOperation<int>
```

## Inheritance

[object](#) ← AddNewProfileUseCase

## Implements

[ICommandWithUndo](#), [ICommandUseCase](#), [IOperation<int>](#)

## Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#),  
[object.ToString\(\)](#)

## Constructors

AddNewProfileUseCase(IProfileRepository,  
IPasswordHasher, ProfileCreateDto)

```
public AddNewProfileUseCase(IProfileRepository repository, IPasswordHasher hashed,  
ProfileDto.ProfileCreateDto profileCreate)
```

## Parameters

repository [IProfileRepository](#)

hashed [IPasswordHasher](#)

profileCreate [ProfileDto.ProfileCreateDto](#)

## Methods

## Execute()

Выполняет операцию и возвращает результат.

```
public Task<int> Execute()
```

### Returns

[Task](#)<int>

Результат выполнения операции типа TResult.

## Undo()

Отменяет выполненную операцию.

```
public Task<int> Undo()
```

### Returns

[Task](#)<int>

Количество затронутых записей при отмене.

# Class ChangeProfileUseCase

Namespace: [Application.UseCase.ProfileUseCases](#)

Assembly: Application.dll

```
public class ChangeProfileUseCase : ICommandWithUndo, ICommandUseCase,  
IOperation<int>
```

## Inheritance

[object](#) ← ChangeProfileUseCase

## Implements

[ICommandWithUndo](#), [ICommandUseCase](#), [IOperation<int>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

ChangeProfileUseCase(IProfileRepository, IPasswordHasher, IUserContext, Guid, string)

```
public ChangeProfileUseCase(IProfileRepository repository, IPasswordHasher hashed,  
IUserContext userContext, Guid newProfile, string password)
```

## Parameters

repository [IProfileRepository](#)

hashed [IPasswordHasher](#)

userContext [IUserContext](#)

newProfile [Guid](#)

password [string](#)

# Methods

## Execute()

Выполняет операцию и возвращает результат.

```
public Task<int> Execute()
```

## Returns

[Task](#)<int>

Результат выполнения операции типа TResult.

## Undo()

Отменяет выполненную операцию.

```
public Task<int> Undo()
```

## Returns

[Task](#)<int>

Количество затронутых записей при отмене.

# Class DeletionProfileUseCase

Namespace: [Application.UseCase.ProfileUseCases](#)

Assembly: Application.dll

```
public class DeletionProfileUseCase : ICommandWithUndo, ICommandUseCase,  
IOperation<int>
```

## Inheritance

[object](#) ← DeletionProfileUseCase

## Implements

[ICommandWithUndo](#), [ICommandUseCase](#), [IOperation<int>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

DeletionProfileUseCase(IProfileRepository,  
IPasswordHasher, Guid, string)

```
public DeletionProfileUseCase(IProfileRepository repository, IPasswordHasher hashed,  
Guid idProfile, string password)
```

## Parameters

repository [IProfileRepository](#)

hashed [IPasswordHasher](#)

idProfile [Guid](#)

password [string](#)

# Methods

## Execute()

Выполняет операцию и возвращает результат.

```
public Task<int> Execute()
```

## Returns

[Task](#)<int>

Результат выполнения операции типа TResult.

## Undo()

Отменяет выполненную операцию.

```
public Task<int> Undo()
```

## Returns

[Task](#)<int>

Количество затронутых записей при отмене.

# Class UpdateProfileUseCase

Namespace: [Application.UseCase.ProfileUseCases](#)

Assembly: Application.dll

```
public class UpdateProfileUseCase : ICommandWithUndo, ICommandUseCase,  
IOperation<int>
```

## Inheritance

[object](#) ← UpdateProfileUseCase

## Implements

[ICommandWithUndo](#), [ICommandUseCase](#), [IOperation<int>](#)

## Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#),  
[object.ToString\(\)](#)

## Constructors

UpdateProfileUseCase(IProfileRepository, IPasswordHasher, ProfileUpdateDto, string)

```
public UpdateProfileUseCase(IProfileRepository repository, IPasswordHasher hashed,  
ProfileDto.ProfileUpdateDto profileUpdate, string password)
```

## Parameters

repository [IProfileRepository](#)

hashed [IPasswordHasher](#)

profileUpdate [ProfileDto.ProfileUpdateDto](#)

password [string](#)

# Methods

## Execute()

Выполняет операцию и возвращает результат.

```
public Task<int> Execute()
```

## Returns

[Task](#)<int>

Результат выполнения операции типа TResult.

## Undo()

Отменяет выполненную операцию.

```
public Task<int> Undo()
```

## Returns

[Task](#)<int>

Количество затронутых записей при отмене.

# Namespace Application.UseCase.ProfileUseCases.Query

## Classes

### [CheckProfileExistsUseCase](#)

Use case для проверки существования профиля по критериям.

### [CountProfilesUseCase](#)

Use case для подсчета профилей по критериям.

### [FindProfilesUseCase](#)

Use case для поиска профилей по критериям.

### [GetAllProfilesUseCase](#)

Use case для получения всех профилей.

### [GetProfileUseCase](#)

Use case для получения одного профиля по критериям.

# Class CheckProfileExistsUseCase

Namespace: [Application.UseCase.ProfileUseCases.Query](#)

Assembly: Application.dll

Use case для проверки существования профиля по критериям.

```
public class CheckProfileExistsUseCase : IQueryUseCase<bool>, IOperation<bool>
```

## Inheritance

[object](#) ← CheckProfileExistsUseCase

## Implements

[IQueryUseCase<bool>](#), [IOperation<bool>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

CheckProfileExistsUseCase(IProfileRepository,  
ProfileSearchDto)

```
public CheckProfileExistsUseCase(IProfileRepository repository,  
ProfileDto.ProfileSearchDto searchDto)
```

## Parameters

repository [IProfileRepository](#)

searchDto [ProfileDto.ProfileSearchDto](#)

## Methods

## Execute()

Проверяет существование профиля, соответствующего критериям из DTO.

```
public Task<bool> Execute()
```

## Returns

[Task](#)<[bool](#)>

true если профиль существует, иначе false.

# Class CountProfilesUseCase

Namespace: [Application.UseCase.ProfileUseCases.Query](#)

Assembly: Application.dll

Use case для подсчета профилей по критериям.

```
public class CountProfilesUseCase : IQueryUseCase<int>, IOperation<int>
```

## Inheritance

[object](#) ← CountProfilesUseCase

## Implements

[IQueryUseCase<int>](#), [IOperation<int>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### CountProfilesUseCase(IProfileRepository, ProfileSearchDto)

```
public CountProfilesUseCase(IProfileRepository repository,  
    ProfileDto.ProfileSearchDto searchDto)
```

## Parameters

repository [IProfileRepository](#)

searchDto [ProfileDto.ProfileSearchDto](#)

## Methods

### Execute()

Подсчитывает количество профилей, соответствующих критериям из DTO.

```
public Task<int> Execute()
```

## Returns

[Task<int>](#)

Количество найденных профилей.

# Class FindProfilesUseCase

Namespace: [Application.UseCase.ProfileUseCases.Query](#)

Assembly: Application.dll

Use case для поиска профилей по критериям.

```
public class FindProfilesUseCase :  
    IQueryUseCase<IEnumerable<ProfileDto.ProfileDetailsDto>>,  
    IOperation<IEnumerable<ProfileDto.ProfileDetailsDto>>
```

**Inheritance**

[object](#) ← FindProfilesUseCase

**Implements**

[IQueryUseCase](#)<[IEnumerable](#)<ProfileDto.ProfileDetailsDto>>,  
[IOperation](#)<[IEnumerable](#)<ProfileDto.ProfileDetailsDto>>

**Inherited Members**

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### FindProfilesUseCase(IProfileRepository, ProfileSearchDto)

```
public FindProfilesUseCase(IProfileRepository repository,  
    ProfileDto.ProfileSearchDto searchDto)
```

## Parameters

repository [IProfileRepository](#)

searchDto [ProfileDto.ProfileSearchDto](#)

## Methods

## Execute()

Выполняет поиск профилей по критериям из DTO.

```
public Task<IEnumerable<ProfileDto.ProfileDetailsDto>> Execute()
```

## Returns

[Task](#)<[IEnumerable](#)<[ProfileDto](#).[ProfileDetailsDto](#)>>

Коллекция найденных профилей в виде ProfileDetailsDto.

# Class GetAllProfilesUseCase

Namespace: [Application.UseCase.ProfileUseCases.Query](#)

Assembly: Application.dll

Use case для получения всех профилей.

```
public class GetAllProfilesUseCase :  
    IQueryUseCase<IEnumerable<ProfileDto.ProfileDetailsDto>>,  
    IOperation<IEnumerable<ProfileDto.ProfileDetailsDto>>
```

**Inheritance**

[object](#) ← GetAllProfilesUseCase

**Implements**

[IQueryUseCase](#)<[IEnumerable](#)<ProfileDto.ProfileDetailsDto>>,  
[IOperation](#)<[IEnumerable](#)<ProfileDto.ProfileDetailsDto>>

**Inherited Members**

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### GetAllProfilesUseCase(IProfileRepository)

```
public GetAllProfilesUseCase(IProfileRepository repository)
```

**Parameters**

repository [IProfileRepository](#)

## Methods

### Execute()

Получает все профили из репозитория.

```
public Task<IEnumerable<ProfileDto.ProfileDetailsDto>> Execute()
```

## Returns

[Task](#)<[IEnumerable](#)<[ProfileDto](#).[ProfileDetailsDto](#)>>

Коллекция всех профилей в виде ProfileDetailsDto.

# Class GetProfileUseCase

Namespace: [Application.UseCase.ProfileUseCases.Query](#)

Assembly: Application.dll

Use case для получения одного профиля по критериям.

```
public class GetProfileUseCase : IQueryUseCase<ProfileDto.ProfileDetailsDto?>,  
IOperation<ProfileDto.ProfileDetailsDto?>
```

**Inheritance**

[object](#) ← GetProfileUseCase

**Implements**

[IQueryUseCase<ProfileDto.ProfileDetailsDto>](#), [IOperation<ProfileDto.ProfileDetailsDto>](#)

**Inherited Members**

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### GetProfileUseCase(IProfileRepository, ProfileSearchDto)

```
public GetProfileUseCase(IProfileRepository repository, ProfileDto.ProfileSearchDto  
searchDto)
```

## Parameters

repository [IProfileRepository](#)

searchDto [ProfileDto.ProfileSearchDto](#)

## Methods

### Execute()

Выполняет поиск одного профиля по критериям из DTO.

```
public Task<ProfileDto.ProfileDetailsDto?> Execute()
```

## Returns

[Task](#)<[ProfileDto](#).[ProfileDetailsDto](#)>

Найденный профиль в виде ProfileDetailsDto или null.

# Namespace Application.UseCase.TodoTask UseCases

## Classes

[AddNewTaskUseCase](#)

[DeletionTaskUseCase](#)

[UpdateTaskUseCase](#)

# Class AddNewTaskUseCase

Namespace: [Application.UseCase.TodoTaskUseCases](#)

Assembly: Application.dll

```
public class AddNewTaskUseCase : ICommandWithUndo, ICommandUseCase, IOperation<int>
```

## Inheritance

[object](#) ← AddNewTaskUseCase

## Implements

[ICommandWithUndo](#), [ICommandUseCase](#), [IOperation<int>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

AddNewTaskUseCase(ITodoTaskRepository, IUserContext, TodoTaskCreateDto)

```
public AddNewTaskUseCase(ITodoTaskRepository repository, IUserContext userContext,  
TodoTaskDto.TodoTaskCreateDto taskCreate)
```

## Parameters

repository [ITodoTaskRepository](#)

userContext [IUserContext](#)

taskCreate [TodoTaskDto.TodoTaskCreateDto](#)

## Methods

## Execute()

Выполняет операцию и возвращает результат.

```
public Task<int> Execute()
```

### Returns

[Task](#)<int>

Результат выполнения операции типа TResult.

## Undo()

Отменяет выполненную операцию.

```
public Task<int> Undo()
```

### Returns

[Task](#)<int>

Количество затронутых записей при отмене.

# Class DeletionTaskUseCase

Namespace: [Application.UseCase.TodoTaskUseCases](#)

Assembly: Application.dll

```
public class DeletionTaskUseCase : ICommandWithUndo, ICommandUseCase,  
IOperation<int>
```

## Inheritance

[object](#) ← DeletionTaskUseCase

## Implements

[ICommandWithUndo](#), [ICommandUseCase](#), [IOperation<int>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

DeletionTaskUseCase(ITodoTaskRepository, Guid)

```
public DeletionTaskUseCase(ITodoTaskRepository repository, Guid idTodoTask)
```

## Parameters

repository [ITodoTaskRepository](#)

idTodoTask [Guid](#)

## Methods

Execute()

Выполняет операцию и возвращает результат.

```
public Task<int> Execute()
```

## Returns

[Task](#)<int>

Результат выполнения операции типа TResult.

## Undo()

Отменяет выполненную операцию.

```
public Task<int> Undo()
```

## Returns

[Task](#)<int>

Количество затронутых записей при отмене.

# Class UpdateTaskUseCase

Namespace: [Application.UseCase.TodoTaskUseCases](#)

Assembly: Application.dll

```
public class UpdateTaskUseCase : ICommandWithUndo, ICommandUseCase, IOperation<int>
```

## Inheritance

[object](#) ← UpdateTaskUseCase

## Implements

[ICommandWithUndo](#), [ICommandUseCase](#), [IOperation<int>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

UpdateTaskUseCase(ITodoTaskRepository,  
TodoTaskUpdateDto)

```
public UpdateTaskUseCase(ITodoTaskRepository repository,  
TodoTaskDto.TodoTaskUpdateDto updateDto)
```

## Parameters

repository [ITodoTaskRepository](#)

updateDto [TodoTaskDto.TodoTaskUpdateDto](#)

## Methods

### Execute()

Выполняет операцию и возвращает результат.

```
public Task<int> Execute()
```

## Returns

[Task](#)<int>

Результат выполнения операции типа TResult.

## Undo()

Отменяет выполненную операцию.

```
public Task<int> Undo()
```

## Returns

[Task](#)<int>

Количество затронутых записей при отмене.

# Namespace Application.UseCase.TodoTask UseCases.Query

## Classes

### [CheckTaskExistsUseCase](#)

Use case для проверки существования задачи по критериям.

### [CountTasksUseCase](#)

Use case для подсчета задач по критериям.

### [FindTasksUseCase](#)

Use case для поиска задач по критериям.

### [GetAllTasksUseCase](#)

Use case для получения всех задач.

### [GetTaskUseCase](#)

Use case для получения одной задачи по критериям.

# Class CheckTaskExistsUseCase

Namespace: [Application.UseCase.TodoTaskUseCases.Query](#)

Assembly: Application.dll

Use case для проверки существования задачи по критериям.

```
public class CheckTaskExistsUseCase : IQueryUseCase<bool>, IOperation<bool>
```

## Inheritance

[object](#) ← CheckTaskExistsUseCase

## Implements

[IQueryUseCase<bool>](#), [IOperation<bool>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

CheckTaskExistsUseCase(ITodoTaskRepository,  
TodoTaskSearchDto)

```
public CheckTaskExistsUseCase(ITodoTaskRepository repository,  
TodoTaskDto.TodoTaskSearchDto searchDto)
```

## Parameters

repository [ITodoTaskRepository](#)

searchDto [TodoTaskDto.TodoTaskSearchDto](#)

## Methods

## Execute()

Проверяет существование задачи, соответствующей критериям из DTO.

```
public Task<bool> Execute()
```

## Returns

[Task](#)<[bool](#)>

true если задача существует, иначе false.

# Class CountTasksUseCase

Namespace: [Application.UseCase.TodoTaskUseCases.Query](#)

Assembly: Application.dll

Use case для подсчета задач по критериям.

```
public class CountTasksUseCase : IQueryUseCase<int>, IOperation<int>
```

## Inheritance

[object](#) ← CountTasksUseCase

## Implements

[IQueryUseCase<int>](#), [IOperation<int>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

CountTasksUseCase(ITodoTaskRepository,  
TodoTaskSearchDto)

```
public CountTasksUseCase(ITodoTaskRepository repository,  
TodoTaskDto.TodoTaskSearchDto searchDto)
```

## Parameters

repository [ITodoTaskRepository](#)

searchDto [TodoTaskDto.TodoTaskSearchDto](#)

## Methods

## Execute()

Подсчитывает количество задач, соответствующих критериям из DTO.

```
public Task<int> Execute()
```

## Returns

[Task](#)<int>

Количество найденных задач.

# Class FindTasksUseCase

Namespace: [Application.UseCase.TodoTaskUseCases.Query](#)

Assembly: Application.dll

Use case для поиска задач по критериям.

```
public class FindTasksUseCase :  
    IQueryUseCase<IEnumerable<TodoTaskDto.TodoTaskDetailsDto>>,  
    IOperation<IEnumerable<TodoTaskDto.TodoTaskDetailsDto>>
```

Inheritance

[object](#) ← FindTasksUseCase

Implements

```
IQueryUseCase<IEnumerable<TodoTaskDto.TodoTaskDetailsDto>>,  
IOperation<IEnumerable<TodoTaskDto.TodoTaskDetailsDto>>
```

Inherited Members

```
object.Equals\(object\) , object.Equals\(object, object\) , object.GetHashCode\(\) ,  
object.GetType\(\) , object.MemberwiseClone\(\) , object.ReferenceEquals\(object, object\) ,  
object.ToString\(\)
```

## Constructors

FindTasksUseCase(ITodoTaskRepository,  
TodoTaskSearchDto)

```
public FindTasksUseCase(ITodoTaskRepository repository,  
TodoTaskDto.TodoTaskSearchDto searchDto)
```

## Parameters

repository [ITodoTaskRepository](#)

searchDto [TodoTaskDto.TodoTaskSearchDto](#)

# Methods

## Execute()

Выполняет поиск задач по критериям из DTO.

```
public Task<IEnumerable<TodoTaskDto.TodoTaskDetailsDto>> Execute()
```

## Returns

[Task](#)<[IEnumerable](#)<[TodoTaskDto](#).[TodoTaskDetailsDto](#)>>

Коллекция найденных задач в виде TodoTaskDetailsDto.

# Class GetAllTasksUseCase

Namespace: [Application.UseCase.TodoTaskUseCases.Query](#)

Assembly: Application.dll

Use case для получения всех задач.

```
public class GetAllTasksUseCase :  
    IQueryUseCase<IEnumerable<TodoTaskDto.TodoTaskDetailsDto>>,  
    IOperation<IEnumerable<TodoTaskDto.TodoTaskDetailsDto>>
```

Inheritance

[object](#) ← GetAllTasksUseCase

Implements

[IQueryUseCase](#)<[IEnumerable](#)<[TodoTaskDto.TodoTaskDetailsDto](#)>>,  
[IOperation](#)<[IEnumerable](#)<[TodoTaskDto.TodoTaskDetailsDto](#)>>

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### GetAllTasksUseCase(ITodoTaskRepository)

```
public GetAllTasksUseCase(ITodoTaskRepository repository)
```

Parameters

repository [ITodoTaskRepository](#)

## Methods

### Execute()

Получает все задачи из репозитория.

```
public Task<IEnumerable<TodoTaskDto.TodoTaskDetailsDto>> Execute()
```

## Returns

[Task](#)<[IEnumerable](#)<[TodoTaskDto](#).[TodoTaskDetailsDto](#)>>

Коллекция всех задач в виде TodoTaskDetailsDto.

# Class GetTaskUseCase

Namespace: [Application.UseCase.TodoTaskUseCases.Query](#)

Assembly: Application.dll

Use case для получения одной задачи по критериям.

```
public class GetTaskUseCase : IQueryUseCase<TodoTaskDto.TodoTaskDetailsDto?>,  
IOperation<TodoTaskDto.TodoTaskDetailsDto?>
```

Inheritance

[object](#) ← GetTaskUseCase

Implements

[IQueryUseCase<TodoTaskDto.TodoTaskDetailsDto>](#), [IOperation<TodoTaskDto.TodoTaskDetailsDto>](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

GetTaskUseCase(ITodoTaskRepository, TodoTaskSearchDto)

```
public GetTaskUseCase(ITodoTaskRepository repository, TodoTaskDto.TodoTaskSearchDto  
searchDto)
```

Parameters

repository [ITodoTaskRepository](#)

searchDto [TodoTaskDto.TodoTaskSearchDto](#)

## Methods

Execute()

Выполняет поиск одной задачи по критериям из DTO.

```
public Task<TodoTaskDto.TodoTaskDetailsDto?> Execute()
```

## Returns

[Task](#)<[TodoTaskDto](#).[TodoTaskDetailsDto](#)>

Найденная задача в виде TodoTaskDetailsDto или null.

# Namespace Domain.Entities.ProfileEntity

## Classes

[Profile](#)

# Class Profile

Namespace: [Domain.Entities.ProfileEntity](#)

Assembly: Domain.dll

```
public class Profile
```

## Inheritance

[object](#) ← Profile

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[ProfileDto.ToCreateDto\(Profile\)](#) , [ProfileDto.ToDetailsDto\(Profile\)](#) , [ProfileDto.ToShortDto\(Profile\)](#) ,  
[ProfileDtoToUpdateDto\(Profile\)](#) , [ProfileMapper.ToEntity\(Profile\)](#)

# Constructors

## Profile(string, string, DateTime, string)

```
public Profile(string firstName, string lastName, DateTime dateOfBirth,  
string passwordHash)
```

## Parameters

firstName [string](#)

lastName [string](#)

dateOfBirth [DateTime](#)

passwordHash [string](#)

# Properties

## CreatedAt

```
public DateTime CreatedAt { get; }
```

## Property Value

[DateTime](#)

## DateOfBirth

```
public DateTime DateOfBirth { get; }
```

## Property Value

[DateTime](#)

## FirstName

```
public string FirstName { get; }
```

## Property Value

[string](#)

## LastName

```
public string LastName { get; }
```

## Property Value

[string](#)

## PasswordHash

```
public string PasswordHash { get; }
```

## Property Value

[string](#)

## ProfileId

```
public Guid ProfileId { get; }
```

## Property Value

[Guid](#)

## Methods

### CreateUpdateObj(Guid, string, string, DateTime)

```
public static Profile CreateUpdateObj(Guid profileId, string firstName, string lastName, DateTime dateOfBirth)
```

## Parameters

profileId [Guid](#)

firstName [string](#)

lastName [string](#)

dateOfBirth [DateTime](#)

## Returns

[Profile](#)

## Restore(Guid, string, string, DateTime, DateTime, string)

```
public static Profile Restore(Guid profileId, string firstName, string lastName,  
DateTime dateOfBirth, DateTime createdAt, string passwordHash)
```

### Parameters

profileId [Guid](#)

firstName [string](#)

lastName [string](#)

dateOfBirth [DateTime](#)

createdAt [DateTime](#)

passwordHash [string](#)

### Returns

[Profile](#)

# Namespace Domain.Entities.TaskEntity

## Classes

[CompletionIndex](#)

[TaskPriority](#)

[TaskPriority.ListPriority](#)

[TaskState](#)

[TaskState.ListState](#)

[TodoTask](#)

# Class CompletionIndex

Namespace: [Domain.Entities.TaskEntity](#)

Assembly: Domain.dll

```
public class CompletionIndex
```

## Inheritance

[object](#) ← CompletionIndex

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Fields

### Default

```
public static readonly CompletionIndex Default
```

### Field Value

[CompletionIndex](#)

### Max

```
public static readonly CompletionIndex Max
```

### Field Value

[CompletionIndex](#)

### Min

```
public static readonly CompletionIndex Min
```

## Field Value

[CompletionIndex](#)

## \_defaultCompletionIndex

```
public static readonly short _defaultCompletionIndex
```

## Field Value

[short](#)

## \_maxCompletionIndex

```
public const short _maxCompletionIndex = 10
```

## Field Value

[short](#)

## Properties

### Completion

```
public short Completion { get; }
```

## Property Value

[short](#)

## Methods

## Percentage()

```
public decimal Percentage()
```

### Returns

[decimal](#)

## Operators

### operator +(CompletionIndex, short)

```
public static CompletionIndex operator +(CompletionIndex left, short right)
```

### Parameters

[left](#) [CompletionIndex](#)

[right](#) [short](#)

### Returns

[CompletionIndex](#)

### operator /(CompletionIndex, short)

```
public static CompletionIndex operator /(CompletionIndex left, short right)
```

### Parameters

[left](#) [CompletionIndex](#)

[right](#) [short](#)

### Returns

[CompletionIndex](#)

## operator \*(CompletionIndex, short)

```
public static CompletionIndex operator *(CompletionIndex left, short right)
```

### Parameters

left [CompletionIndex](#)

right [short](#)

### Returns

[CompletionIndex](#)

## operator -(CompletionIndex, short)

```
public static CompletionIndex operator -(CompletionIndex left, short right)
```

### Parameters

left [CompletionIndex](#)

right [short](#)

### Returns

[CompletionIndex](#)

# Class TaskPriority

Namespace: [Domain.Entities.TaskEntity](#)

Assembly: Domain.dll

```
public class TaskPriority
```

## Inheritance

[object](#) ← TaskPriority

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Fields

### Critical

```
public static readonly TaskPriority Critical
```

### Field Value

[TaskPriority](#)

### High

```
public static readonly TaskPriority High
```

### Field Value

[TaskPriority](#)

### Low

```
public static readonly TaskPriority Low
```

## Field Value

[TaskPriority](#)

## Medium

```
public static readonly TaskPriority Medium
```

## Field Value

[TaskPriority](#)

# Properties

## Level

```
public int Level { get; }
```

## Property Value

[int](#)

## Name

```
public string Name { get; }
```

## Property Value

[string](#)

# Methods

## Deescalate()

```
public TaskPriority Deescalate()
```

### Returns

[TaskPriority](#)

## Escalate()

```
public TaskPriority Escalate()
```

### Returns

[TaskPriority](#)

# Class TaskPriority.ListPriority

Namespace: [Domain.Entities.TaskEntity](#)

Assembly: Domain.dll

```
public static class TaskPriority.ListPriority
```

## Inheritance

[object](#) ← TaskPriority.ListPriority

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### GetByLevel(int)

```
public static TaskPriority GetByLevel(int level)
```

## Parameters

level [int](#)

## Returns

[TaskPriority](#)

# Class TaskState

Namespace: [Domain.Entities.TaskEntity](#)

Assembly: Domain.dll

```
public class TaskState
```

## Inheritance

[object](#) ← TaskState

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Fields

### Completed

```
public static readonly TaskState Completed
```

#### Field Value

[TaskState](#)

### InProgress

```
public static readonly TaskState InProgress
```

#### Field Value

[TaskState](#)

### NotCompleted

```
public static readonly TaskState NotCompleted
```

## Field Value

[TaskState](#)

## Uncertain

```
public static readonly TaskState Uncertain
```

## Field Value

[TaskState](#)

## Properties

### CompletionIndex

```
public CompletionIndex CompletionIndex { get; }
```

## Property Value

[CompletionIndex](#)

## Description

```
public string Description { get; }
```

## Property Value

[string](#) ↗

## Name

```
public string Name { get; }
```

Property Value

[string](#) ↗

StateId

```
public int StateId { get; }
```

Property Value

[int](#) ↗

# Class TaskState.ListState

Namespace: [Domain.Entities.TaskEntity](#)

Assembly: Domain.dll

```
public static class TaskState.ListState
```

## Inheritance

[object](#) ← TaskState.ListState

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### GetById(int)

```
public static TaskState GetById(int stateId)
```

## Parameters

stateId [int](#)

## Returns

[TaskState](#)

# Class TodoTask

Namespace: [Domain.Entities.TaskEntity](#)

Assembly: Domain.dll

```
public class TodoTask
```

## Inheritance

[object](#) ← TodoTask

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[TodoTaskDto.ToCreateDto\(TodoTask, IUserContext\)](#) , [TodoTaskDto.ToDetailsDto\(TodoTask\)](#) ,  
[TodoTaskDto.ToShortDto\(TodoTask\)](#) , [TodoTaskDtoToUpdateDto\(TodoTask\)](#) ,  
[TodoTaskMapper.ToEntity\(TodoTask\)](#)

## Constructors

TodoTask(Guid, string, string?, DateTime?, TaskState?, TaskPriority?)

```
public TodoTask(Guid profileId, string name, string? description = null, DateTime? deadline = null, TaskState? state = null, TaskPriority? priority = null)
```

## Parameters

profileId [Guid](#)

name [string](#)

description [string](#)

deadline [DateTime](#)?

state [TaskState](#)

priority [TaskPriority](#)

## Properties

### CreatedAt

```
public DateTime CreatedAt { get; }
```

### Property Value

[DateTime](#)

### Deadline

```
public DateTime? Deadline { get; }
```

### Property Value

[DateTime](#)?

### Description

```
public string? Description { get; }
```

### Property Value

[string](#)

### Name

```
public string Name { get; }
```

## PropertyValue

[string](#) ↗

## Priority

```
public TaskPriority Priority { get; }
```

## PropertyValue

[TaskPriority](#)

## ProfileId

```
public Guid ProfileId { get; }
```

## PropertyValue

[Guid](#) ↗

## State

```
public TaskState State { get; }
```

## PropertyValue

[TaskState](#)

## TaskId

```
public Guid TaskId { get; }
```

## PropertyValue

## Methods

CreateUpdateObj(Guid, TaskState, TaskPriority, string, string?, DateTime?)

```
public static TodoTask CreateUpdateObj(Guid taskId, TaskState state, TaskPriority priority, string name, string? description, DateTime? deadline)
```

### Parameters

taskId [Guid ↗](#)

state [TaskState](#)

priority [TaskPriority](#)

name [string ↗](#)

description [string ↗](#)

deadline [DateTime ↗](#)?

### Returns

[TodoTask](#)

Restore(Guid, int, int, Guid, string, string?, DateTime, DateTime?)

```
public static TodoTask Restore(Guid taskId, int stateId, int priorityLevel, Guid profileId, string name, string? description, DateTime createdAt, DateTime? deadline)
```

### Parameters

taskId [Guid ↗](#)

stateId [int](#)

priorityLevel [int](#)

profileId [Guid](#)

name [string](#)

description [string](#)

createdAt [DateTime](#)

deadline [DateTime](#)?

Returns

[TodoTask](#)

# Namespace Infrastructure

## Classes

[CommandManager](#)

[UserContext](#)

# Class CommandManager

Namespace: [Infrastructure](#)

Assembly: Infrastructure.dll

```
public class CommandManager : ICommandManager
```

## Inheritance

[object](#) ← CommandManager

## Implements

[ICommandManager](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### ExecuteCommandAsync(ICommandWithUndo)

```
public Task<int> ExecuteCommandAsync(ICommandWithUndo command)
```

#### Parameters

command [ICommandWithUndo](#)

#### Returns

[Task](#)<[int](#)>

### Redo()

```
public Task<int> Redo()
```

## Returns

[Task](#)<[int](#)>

## Undo()

```
public Task<int> Undo()
```

## Returns

[Task](#)<[int](#)>

# Class UserContext

Namespace: [Infrastructure](#)

Assembly: Infrastructure.dll

```
public class UserContext : IUserContext
```

## Inheritance

[object](#) ← UserContext

## Implements

[IUserContext](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### UserContext(Guid?)

```
public UserContext(Guid? idProfile = null)
```

## Parameters

`idProfile` [Guid](#)?

## Properties

### UserId

```
public Guid? UserId { get; }
```

## Property Value

[Guid](#)?

## Methods

### Clear()

```
public void Clear()
```

### Set(Guid?)

```
public void Set(Guid? userId)
```

## Parameters

userId [Guid](#)?

# Namespace Infrastructure.Database Classes

[DatabaseInitialization](#)

[TodoContext](#)

# Class DatabaseInitialization

Namespace: [Infrastructure.Database](#)

Assembly: Infrastructure.dll

```
public class DatabaseInitialization
```

## Inheritance

[object](#) ← DatabaseInitialization

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### DatabaseInitialization(TodoContext)

```
public DatabaseInitialization(TodoContext context)
```

## Parameters

context [TodoContext](#)

## Methods

### InitializeAsync()

```
public Task InitializeAsync()
```

## Returns

[Task](#)

# Class TodoContext

Namespace: [Infrastructure.Database](#)

Assembly: Infrastructure.dll

```
public class TodoContext : DbContext, IInfrastructure<IServiceProvider>,  
 IDbContextDependencies, IDbSetCache, IDbContextPoolable, IResettableService,  
 IDisposable, IAsyncDisposable
```

## Inheritance

[object](#) ← [DbContext](#) ← TodoContext

## Implements

[IInfrastructure](#)<[IServiceProvider](#)>, [IDbContextDependencies](#), [IDbSetCache](#),  
[IDbContextPoolable](#), [IResettableService](#), [IDisposable](#), [IAsyncDisposable](#)

## Inherited Members

[DbContext.Set< TEntity >\(\)](#) , [DbContext.Set< TEntity >\(string\)](#) ,  
[DbContext.ConfigureConventions\(ModelConfigurationBuilder\)](#) , [DbContext.SaveChanges\(\)](#) ,  
[DbContext.SaveChanges\(bool\)](#) , [DbContext.SaveChangesAsync\(CancellationToken\)](#) ,  
[DbContext.SaveChangesAsync\(bool, CancellationToken\)](#) , [DbContext.Dispose\(\)](#) ,  
[DbContext.DisposeAsync\(\)](#) , [DbContext.Entry< TEntity >\(TEntity\)](#) , [DbContext.Entry\(object\)](#) ,  
[DbContext.Add< TEntity >\(TEntity\)](#) , [DbContext.AddAsync< TEntity >\(TEntity, CancellationToken\)](#) ,  
[DbContext.Attach< TEntity >\(TEntity\)](#) , [DbContext.Update< TEntity >\(TEntity\)](#) ,  
[DbContext.Remove< TEntity >\(TEntity\)](#) , [DbContext.Add\(object\)](#) ,  
[DbContext.AddAsync\(object, CancellationToken\)](#) , [DbContext.Attach\(object\)](#) ,  
[DbContext.Update\(object\)](#) , [DbContext.Remove\(object\)](#) ,  
[DbContext.AddRange\(params object\[\]\)](#) , [DbContext.AddRangeAsync\(params object\[\]\)](#) ,  
[DbContext.AttachRange\(params object\[\]\)](#) , [DbContext.UpdateRange\(params object\[\]\)](#) ,  
[DbContext.RemoveRange\(params object\[\]\)](#) , [DbContext.AddRange\(IEnumerable< object >\)](#) ,  
[DbContext.AddRangeAsync\(IEnumerable< object >, CancellationToken\)](#) ,  
[DbContext.AttachRange\(IEnumerable< object >\)](#) ,  
[DbContext.UpdateRange\(IEnumerable< object >\)](#) ,  
[DbContext.RemoveRange\(IEnumerable< object >\)](#) , [DbContext.Find\(Type, params object\[\]\)](#) ,  
[DbContext.FindAsync\(Type, params object\[\]\)](#) ,  
[DbContext.FindAsync\(Type, object\[\], CancellationToken\)](#) ,  
[DbContext.Find< TEntity >\(params object\[\]\)](#) , [DbContext.FindAsync< TEntity >\(params object\[\]\)](#) ,  
[DbContext.FindAsync< TEntity >\(object\[\], CancellationToken\)](#) ,  
[DbContext.FromExpression< TResult >\(Expression< Func< IQueryable< TResult > > \)](#) ,

[DbContext.Database](#) , [DbContext.ChangeTracker](#) , [DbContext.Model](#) ,  
[DbContext.ContextId](#) , [DbContext.SavingChanges](#) , [DbContext.SavedChanges](#) ,  
[DbContext.SaveChangesFailed](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Properties

### Profiles

```
public DbSet<ProfileEntity> Profiles { get; set; }
```

### Property Value

[DbSet](#)<[ProfileEntity](#)>

### Tasks

```
public DbSet<TodoTaskEntity> Tasks { get; set; }
```

### Property Value

[DbSet](#)<[TodoTaskEntity](#)>

## Methods

### OnConfiguring(DbContextOptionsBuilder)

Override this method to configure the database (and other options) to be used for this context. This method is called for each instance of the context that is created. The base implementation does nothing.

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
```

## Parameters

## optionsBuilder [DbContextOptionsBuilder](#)

A builder used to create or modify options for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure the context.

## Remarks

In situations where an instance of [DbContextOptions](#) may or may not have been passed to the constructor, you can use [IsConfigured](#) to determine if the options have already been set, and skip some or all of the logic in [OnConfiguring\(DbContextOptionsBuilder\)](#).

See [DbContext lifetime, configuration, and initialization](#) for more information and examples.

## OnModelCreating(ModelBuilder)

Override this method to further configure the model that was discovered by convention from the entity types exposed in [DbSet< TEntity >](#) properties on your derived context. The resulting model may be cached and re-used for subsequent instances of your derived context.

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

## Parameters

### modelBuilder [ModelBuilder](#)

The builder being used to construct the model for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure aspects of the model that are specific to a given database.

## Remarks

If a model is explicitly set on the options for this context (via [UseModel\(IModel\)](#)) then this method will not be run. However, it will still run when creating a compiled model.

See [Modeling entity types and relationships](#) for more information and examples.

# Namespace Infrastructure.Database.Entity Classes

[ProfileEntity](#)

[TodoTaskEntity](#)

# Class ProfileEntity

Namespace: [Infrastructure.Database.Entity](#)

Assembly: Infrastructure.dll

```
public class ProfileEntity
```

## Inheritance

[object](#) ← ProfileEntity

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[ProfileMapper.ToDomain\(ProfileEntity\)](#)

# Properties

## CreatedAt

```
public DateTime CreatedAt { get; set; }
```

## Property Value

[DateTime](#)

## DateOfBirth

```
public DateTime DateOfBirth { get; set; }
```

## Property Value

[DateTime](#)

## FirstName

```
public string FirstName { get; set; }
```

## Property Value

[string ↗](#)

## LastName

```
public string LastName { get; set; }
```

## Property Value

[string ↗](#)

## PasswordHash

```
public string PasswordHash { get; set; }
```

## Property Value

[string ↗](#)

## ProfileId

```
public string ProfileId { get; set; }
```

## Property Value

[string ↗](#)

## Tasks

```
public virtual ICollection<TodoTaskEntity>? Tasks { get; set; }
```

## Property Value

[ICollection](#)<[TodoTaskEntity](#)>

# Class TodoTaskEntity

Namespace: [Infrastructure.Database.Entity](#)

Assembly: Infrastructure.dll

```
public class TodoTaskEntity
```

## Inheritance

[object](#) ← TodoTaskEntity

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[TodoTaskMapper.ToDomain\(TodoTaskEntity\)](#)

# Properties

## CreateAt

```
public DateTime CreateAt { get; set; }
```

## Property Value

[DateTime](#)

## Deadline

```
public DateTime? Deadline { get; set; }
```

## Property Value

[DateTime](#)?

## Description

```
public string? Description { get; set; }
```

## Property Value

[string](#)

## Name

```
public string Name { get; set; }
```

## Property Value

[string](#)

## PriorityLevel

```
public int PriorityLevel { get; set; }
```

## Property Value

[int](#)

## ProfileId

```
public string ProfileId { get; set; }
```

## Property Value

[string](#)

## StateId

```
public int StateId { get; set; }
```

## Property Value

[int ↗](#)

## TaskCreator

```
public virtual ProfileEntity? TaskCreator { get; set; }
```

## Property Value

[ProfileEntity](#)

## TaskId

```
public string TaskId { get; set; }
```

## Property Value

[string ↗](#)

# Namespace Infrastructure.EfRepository

## Classes

[CriteriaToExpression](#)

[EfProfileRepository](#)

[EfTodoTaskRepository](#)

# Class CriteriaToExpression

Namespace: [Infrastructure.EfRepository](#)

Assembly: Infrastructure.dll

```
public static class CriteriaToExpression
```

## Inheritance

[object](#) ← CriteriaToExpression

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

**IsSatisfiedBy< TEntity, T >(CriteriaId< T >, Expression< Func< TEntity, T > >)**

Строит выражение для фильтрации сущности по селектору свойства.

```
public static Expression<Func< TEntity, bool>> IsSatisfiedBy< TEntity, T >
(this CriteriaId< T > criteriaId, Expression< Func< TEntity, T > > selector) where T
: IComparable< T >
```

## Parameters

criteriaId [CriteriaId](#)< T >

selector [Expression](#)< [Func](#)< TEntity, T > >

## Returns

[Expression](#)< [Func](#)< TEntity, bool > >

## Type Parameters

**TEntity**

**T**

## IsSatisfiedBy<TEntity, T>(CriteriaObj<T>, Expression<Func<TEntity, T>>)

Строит выражение для фильтрации сущности по селектору свойства.

```
public static Expression<Func<TEntity, bool>> IsSatisfiedBy<TEntity, T>(this  
CriteriaObj<T> criteriaObj, Expression<Func<TEntity, T>> selector) where T  
: IComparable<T>
```

### Parameters

**criteriaObj** [CriteriaObj<T>](#)

**selector** [Expression<Func<TEntity, T>>](#)

### Returns

[Expression<Func<TEntity, bool>>](#)

### Type Parameters

**TEntity**

**T**

## IsSatisfiedBy<TEntity, TRange, TInRange> (CriteriaRangeObj<TRange, TInRange>, Expression<Func<TEntity, TInRange>>)

Строит выражение для фильтрации сущности по селектору свойства.

```
public static Expression<Func<TEntity, bool>> IsSatisfiedBy<TEntity, TRange,  
TInRange>(this CriteriaRangeObj<TRange, TInRange> criteriaRangeObj,  
Expression<Func<TEntity, TInRange>> selector) where TRange : IComparable<TInRange>
```

## Parameters

criteriaRangeObj [CriteriaRangeObj](#)<TRange, TInRange>

selector [Expression](#)<[Func](#)<TEntity, TInRange>>

## Returns

[Expression](#)<[Func](#)<TEntity, [bool](#)>>

## Type Parameters

TEntity

TRange

TInRange

# Class EfProfileRepository

Namespace: [Infrastructure.EfRepository](#)

Assembly: Infrastructure.dll

```
public class EfProfileRepository : IProfileRepository, IBaseRepository<Profile>,  
IFilterByCriteria<Profile, ProfileCriteria>
```

## Inheritance

[object](#) ← EfProfileRepository

## Implements

[IProfileRepository](#), [IBaseRepository<Profile>](#), [IFilterByCriteria<Profile, ProfileCriteria>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### EfProfileRepository(TodoContext)

```
public EfProfileRepository(TodoContext context)
```

## Parameters

context [TodoContext](#)

## Methods

### AddAsync(Profile)

```
public Task<int> AddAsync(Profile profile)
```

## Parameters

`profile` [Profile](#)

## Returns

[Task](#)<[int](#)>

## CountAsync(ProfileCriteria)

```
public Task<int> CountAsync(ProfileCriteria profileCriteria)
```

## Parameters

`profileCriteria` [ProfileCriteria](#)

## Returns

[Task](#)<[int](#)>

## DeleteAsync(Guid)

```
public Task<int> DeleteAsync(Guid id)
```

## Parameters

`id` [Guid](#)

## Returns

[Task](#)<[int](#)>

## ExistsAsync(ProfileCriteria)

```
public Task<bool> ExistsAsync(ProfileCriteria profileCriteria)
```

## Parameters

profileCriteria [ProfileCriteria](#)

## Returns

[Task](#)<[bool](#)>

## FindAsync(ProfileCriteria)

```
public Task<IEnumerable<Profile>> FindAsync(ProfileCriteria profileCriteria)
```

## Parameters

profileCriteria [ProfileCriteria](#)

## Returns

[Task](#)<[IEnumerable](#)<[Profile](#)>>

## FindSingleAsync(ProfileCriteria)

```
public Task<Profile?> FindSingleAsync(ProfileCriteria profileCriteria)
```

## Parameters

profileCriteria [ProfileCriteria](#)

## Returns

[Task](#)<[Profile](#)>

## GetAllAsync()

```
public Task<IEnumerable<Profile>> GetAllAsync()
```

## Returns

[Task](#)<[IEnumerable](#)<[Profile](#)>>

## GetByIdAsync(Guid)

`public Task<Profile?> GetByIdAsync(Guid id)`

## Parameters

`id` [Guid](#)

## Returns

[Task](#)<[Profile](#)>

## UpdateAsync(Profile)

`public Task<int> UpdateAsync(Profile profile)`

## Parameters

`profile` [Profile](#)

## Returns

[Task](#)<[int](#)>

# Class EfTodoTaskRepository

Namespace: [Infrastructure.EfRepository](#)

Assembly: Infrastructure.dll

```
public class EfTodoTaskRepository : ITodoTaskRepository, IBaseRepository<TodoTask>,  
IFilterByCriteria<TodoTask, TaskCriteria>
```

## Inheritance

[object](#) ← EfTodoTaskRepository

## Implements

[ITodoTaskRepository](#), [IBaseRepository<TodoTask>](#), [IFilterByCriteria<TodoTask, TaskCriteria>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### EfTodoTaskRepository(TodoContext)

```
public EfTodoTaskRepository(TodoContext context)
```

## Parameters

context [TodoContext](#)

## Methods

### AddAsync(TodoTask)

```
public Task<int> AddAsync(TodoTask todo)
```

## Parameters

todo [TodoTask](#)

## Returns

[Task](#)<[int](#)>

## CountAsync(TaskCriteria)

```
public Task<int> CountAsync(TaskCriteria profileCriteria)
```

## Parameters

profileCriteria [TaskCriteria](#)

## Returns

[Task](#)<[int](#)>

## DeleteAsync(Guid)

```
public Task<int> DeleteAsync(Guid id)
```

## Parameters

id [Guid](#)

## Returns

[Task](#)<[int](#)>

## ExistsAsync(TaskCriteria)

```
public Task<bool> ExistsAsync(TaskCriteria profileCriteria)
```

## Parameters

profileCriteria [TaskCriteria](#)

## Returns

[Task](#)<[bool](#)>

## FindAsync(TaskCriteria)

```
public Task<IEnumerable<TodoTask>> FindAsync(TaskCriteria profileCriteria)
```

## Parameters

profileCriteria [TaskCriteria](#)

## Returns

[Task](#)<[IEnumerable](#)<[TodoTask](#)>>

## FindSingleAsync(TaskCriteria)

```
public Task<TodoTask?> FindSingleAsync(TaskCriteria profileCriteria)
```

## Parameters

profileCriteria [TaskCriteria](#)

## Returns

[Task](#)<[TodoTask](#)>

## GetAllAsync(IUserContext)

```
public Task<IEnumerable<TodoTask>> GetAllAsync(IUserContext userContext)
```

## Parameters

userContext [IUserContext](#)

## Returns

[Task](#)<[IEnumerable](#)<[TodoTask](#)>>

## GetByIdAsync(Guid)

```
public Task<TodoTask?> GetByIdAsync(Guid id)
```

## Parameters

id [Guid](#)

## Returns

[Task](#)<[TodoTask](#)>

## UpdateAsync(TodoTask)

```
public Task<int> UpdateAsync(TodoTask todo)
```

## Parameters

todo [TodoTask](#)

## Returns

[Task](#)<[int](#)>

# Namespace Infrastructure.EfRepository.Mapper

## Classes

[ProfileMapper](#)

[TodoTaskMapper](#)

# Class ProfileMapper

Namespace: [Infrastructure.EfRepository.Mapper](#)

Assembly: Infrastructure.dll

```
public static class ProfileMapper
```

## Inheritance

[object](#) ← ProfileMapper

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### ToDomain(ProfileEntity)

```
public static Profile ToDomain(this ProfileEntity profileEntity)
```

#### Parameters

profileEntity [ProfileEntity](#)

#### Returns

[Profile](#)

### ToEntity(Profile)

```
public static ProfileEntity ToEntity(this Profile profile)
```

#### Parameters

`profile` [Profile](#)

Returns

[ProfileEntity](#)

# Class TodoTaskMapper

Namespace: [Infrastructure.EfRepository.Mapper](#)

Assembly: Infrastructure.dll

```
public static class TodoTaskMapper
```

## Inheritance

[object](#) ← TodoTaskMapper

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### ToDomain(TodoTaskEntity)

```
public static TodoTask ToDomain\(this TodoTaskEntity todoTaskEntity)
```

#### Parameters

todoTaskEntity [TodoTaskEntity](#)

#### Returns

[TodoTask](#)

### ToEntity(TodoTask)

```
public static TodoTaskEntity ToEntity\(this TodoTask todoTask)
```

#### Parameters

todoTask [TodoTask](#)

Returns

[TodoTaskEntity](#)

# Namespace Infrastructure.Encryption

## Classes

[PasswordHasher](#)

# Class PasswordHasher

Namespace: [Infrastructure.Encryption](#)

Assembly: Infrastructure.dll

```
public class PasswordHasher : IPasswordHasher
```

## Inheritance

[object](#) ← PasswordHasher

## Implements

[IPasswordHasher](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### HashedAsync(string)

```
public Task<string> HashedAsync(string password)
```

#### Parameters

password [string](#)

#### Returns

[Task](#)<[string](#)>

### VerifyAsync(string, string)

```
public Task<bool> VerifyAsync(string password, string hash)
```

## Parameters

password [string](#)

hash [string](#)

## Returns

[Task](#)<[bool](#)>

# Namespace Presentation

## Classes

[Launch](#)

# Class Launch

Namespace: [Presentation](#)

Assembly: Presentation.dll

```
public static class Launch
```

## Inheritance

[object](#) ← Launch

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Methods

## CyclicRun()

```
public static Task<short> CyclicRun()
```

## Returns

[Task](#)<[short](#)>

## RunOnce(string[])

```
public static Task<short> RunOnce(string[] args)
```

## Parameters

args [string](#)[]

## Returns

[Task](#) <[short](#)>

UpdateRepositories(IProfileRepository,  
ITodoTaskRepository, IUserContext, IPasswordHasher,  
ICommandManager)

```
public static Task UpdateRepositories(IProfileRepository profileRepository,  
ITodoTaskRepository todoTaskRepository, IUserContext userContextService,  
IPasswordHasher passwordHasher, ICommandManager commandManager)
```

## Parameters

profileRepository [IProfileRepository](#)

todoTaskRepository [ITodoTaskRepository](#)

userContextService [IUserContext](#)

passwordHasher [IPasswordHasher](#)

commandManager [ICommandManager](#)

## Returns

[Task](#)

# Namespace Presentation.Adapters

## Classes

### [IOFacade](#)

Фасад для объединенной работы с вводом и выводом информации. Позволяет использовать единый интерфейс для взаимодействия с пользователем.

### [InputAdapter](#)

Адаптер для работы только с вводом информации. Реализует интерфейсы ввода текста, чисел, паролей и кнопок.

### [OutputAdapter](#)

Адаптер для работы только с выводом информации. Реализует интерфейс вывода ошибок и цветного вывода.

# Class IOFacade

Namespace: [Presentation.Adapters](#)

Assembly: Presentation.dll

Фасад для объединенной работы с вводом и выводом информации. Позволяет использовать единый интерфейс для взаимодействия с пользователем.

```
public class IOFacade
```

## Inheritance

[object](#) ← IOFacade

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### IOFacade(IInputProvider?, IErrorOutput?)

Инициализирует фасад с провайдерами ввода и вывода.

```
public IOFacade(IInputProvider? input = null, IErrorOutput? output = null)
```

## Parameters

**input** [IInputProvider](#)

Провайдер ввода (опционально).

**output** [IErrorOutput](#)

Провайдер вывода (опционально).

## Properties

## Input

Получить провайдер ввода.

```
public IInputProvider Input { get; }
```

## Property Value

[IInputProvider](#)

## Output

Получить провайдер вывода.

```
public IErrorOutput Output { get; }
```

## Property Value

[IErrorOutput](#)

## Methods

### AskNumeric(string)

Вывести вопрос и получить число

```
public int AskNumeric(string question)
```

#### Parameters

question [string](#)

#### Returns

[int](#)

## AskSelection(string, IEnumerable<string>, int)

Вывести вопрос и получить выбор из списка

```
public string AskSelection(string question, IEnumerable<string> options, int  
pageSize = 3)
```

### Parameters

question [string](#)

options [IEnumerable](#)<[string](#)>

pageSize [int](#)

### Returns

[string](#)

## AskText(string, bool)

Вывести вопрос и получить строку от пользователя.

```
public string AskText(string question, bool notNull = true)
```

### Parameters

question [string](#)

Вопрос для пользователя.

notNull [bool](#)

Требовать непустой ответ.

### Returns

[string](#)

Введённая строка.

## AskYesNo(string)

Вывести вопрос и получить да/нет

```
public bool AskYesNo(string question)
```

### Parameters

question [string](#)

### Returns

[bool](#)

## ShowError(string)

Вывести ошибку

```
public void ShowError(string error)
```

### Parameters

error [string](#)

## ShowException(Exception)

Вывести исключение

```
public void ShowException(Exception exception)
```

### Parameters

exception [Exception](#)

## ShowInfo(string)

Вывести информацию

```
public void ShowInfo(string message)
```

## Parameters

message [string](#)

## ShowSeparator(string)

Вывести разделитель

```
public void ShowSeparator(string title = "")
```

## Parameters

title [string](#)

## ShowSuccess(string)

Вывести успешное сообщение

```
public void ShowSuccess(string message)
```

## Parameters

message [string](#)

## ShowWarning(string)

Вывести предупреждение

```
public void ShowWarning(string message)
```

## Parameters

message [string](#)

# Class InputAdapter

Namespace: [Presentation.Adapters](#)

Assembly: Presentation.dll

Адаптер для работы только с вводом информации. Реализует интерфейсы ввода текста, чисел, паролей и кнопок.

```
public class InputAdapter : IInputProvider, ITextInput, INumericInput,  
IPasswordInput, IButtonInput
```

## Inheritance

[object](#) ← InputAdapter

## Implements

[IInputProvider](#), [ITextInput](#), [INumericInput](#), [IPasswordInput](#), [IButtonInput](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

**InputAdapter(IColoredOutput?, ITextInput?,  
INumericInput?, IPasswordInput?, IButtonInput?)**

Конструктор адаптера ввода.

```
public InputAdapter(IColoredOutput? output = null, ITextInput? textInput = null,  
INumericInput? numericInput = null, IPasswordInput? passwordInput = null,  
IButtonInput? buttonInput = null)
```

## Parameters

**output [IColoredOutput](#)**

Вывод информации (опционально).

## **textInput** [ITextInput](#)

Ввод текста (опционально).

## **numericInput** [INumericInput](#)

Ввод чисел (опционально).

## **passwordInput** [IPasswordInput](#)

Ввод паролей (опционально).

## **buttonInput** [IButtonInput](#)

Ввод кнопок (опционально).

# Methods

## GetCheckedPassword()

Получить пароль с подтверждением.

```
public string GetCheckedPassword()
```

## Returns

[string](#)

## GetKeyFromSet(string, ConsoleKey, params ConsoleKey[])

Получить нажатую клавишу из набора.

```
public ConsoleKey GetKeyFromSet(string prompt, ConsoleKey defaultKey = ConsoleKey.Y,  
params ConsoleKey[] allowedKeys)
```

## Parameters

**prompt** [string](#)

Подсказка.

`defaultKey` [ConsoleKey](#)

Клавиша по умолчанию.

`allowedKeys` [ConsoleKey](#)[]

Разрешённые клавиши.

## Returns

[ConsoleKey](#)

## GetLongText(string)

Получить длинный текст от пользователя.

```
public string GetLongText(string prompt)
```

## Parameters

`prompt` [string](#)

## Returns

[string](#)

## GetNonEmptyText(string)

Получить непустой текст от пользователя.

```
public string GetNonEmptyText(string prompt)
```

## Parameters

`prompt` [string](#)

## Returns

[string](#)

## GetNumeric(string)

Получить числовое значение от пользователя.

```
public int GetNumeric(string prompt)
```

### Parameters

**prompt** [string](#)

### Returns

[int](#)

## GetNumericInRange(string, int, int)

Получить числовое значение в заданном диапазоне.

```
public int GetNumericInRange(string prompt, int min, int max)
```

### Parameters

**prompt** [string](#)

**min** [int](#)

**max** [int](#)

### Returns

[int](#)

## GetNumericWithMax(string, int)

Получить числовое значение с максимальным ограничением.

```
public int GetNumericWithMax(string prompt, int max)
```

## Parameters

prompt [string](#)

max [int](#)

## Returns

[int](#)

## GetNumericWithMin(string, int)

Получить числовое значение с минимальным ограничением.

```
public int GetNumericWithMin(string prompt, int min)
```

## Parameters

prompt [string](#)

min [int](#)

## Returns

[int](#)

## GetPassword(string)

Получить пароль от пользователя.

```
public string GetPassword(string prompt)
```

## Parameters

prompt [string](#)

## Returns

[string](#)

## GetPositiveNumeric(string)

Получить положительное числовое значение.

```
public int GetPositiveNumeric(string prompt)
```

## Parameters

[prompt string](#)

## Returns

[int](#)

## GetSelectionFromDictionary(Dictionary<int, string>, string?, int)

Получить выбор из словаря.

```
public KeyValuePair<int, string> GetSelectionFromDictionary(Dictionary<int, string> options, string? title = null, int pageSize = 3)
```

## Parameters

[options Dictionary<int, string>](#)

Словарь вариантов.

[title string](#)

Заголовок (опционально).

[pageSize int](#)

Размер страницы.

## Returns

[KeyValuePair](#)<int, string>

## GetSelectionFromList(IEnumerable<string>, string?, int)

Получить выбор из списка.

```
public string GetSelectionFromList(IEnumerable<string> options, string? title = null, int pageSize = 3)
```

## Parameters

`options` [IEnumerable](#)<string>

Список вариантов.

`title` [string](#)

Заголовок (опционально).

`pageSize` [int](#)

Размер страницы.

## Returns

[string](#)

## GetShortText(string, bool)

Получить короткий текст от пользователя.

```
public string GetShortText(string prompt, bool notNull = true)
```

## Parameters

`prompt` [string](#)

`notNull` [bool](#)

## Returns

[string](#)

## GetYesNoChoice(string)

Получить выбор "Да/Нет" от пользователя.

```
public bool GetYesNoChoice(string prompt)
```

## Parameters

[prompt](#) [string](#)

## Returns

[bool](#)

## ValidatePasswordLength(string, int)

Проверить длину пароля.

```
public bool ValidatePasswordLength(string password, int minLength = 8)
```

## Parameters

[password](#) [string](#)

Пароль.

[minLength](#) [int](#)

Минимальная длина.

## Returns

[bool](#)

True, если длина достаточна.

# Class OutputAdapter

Namespace: [Presentation.Adapters](#)

Assembly: Presentation.dll

Адаптер для работы только с выводом информации. Реализует интерфейс вывода ошибок и цветного вывода.

```
public class OutputAdapter : IErrorOutput, IColoredOutput, IOutputProvider
```

## Inheritance

[object](#) ← OutputAdapter

## Implements

[IErrorOutput](#), [IColoredOutput](#), [IOutputProvider](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### OutputAdapter(IErrorOutput?, IColoredOutput?)

Конструктор адаптера вывода.

```
public OutputAdapter(IErrorOutput? errorOutput = null, IColoredOutput? coloredOutput  
= null)
```

## Parameters

### errorOutput [IErrorOutput](#)

Интерфейс вывода ошибок (опционально).

### coloredOutput [IColoredOutput](#)

Интерфейс цветного вывода (опционально).

# Methods

## WriteColoredLine(string, ConsoleColor)

Вывести цветное сообщение с переносом строки.

```
public void WriteColoredLine(string message, ConsoleColor color)
```

### Parameters

message [string](#)

color [ConsoleColor](#)

## WriteColoredMessage(string, ConsoleColor)

Вывести цветное сообщение без переноса строки.

```
public void WriteColoredMessage(string message, ConsoleColor color)
```

### Parameters

message [string](#)

color [ConsoleColor](#)

## WriteDetailedException(Exception)

Вывести подробное исключение.

```
public void WriteDetailedException(Exception exception)
```

### Parameters

exception [Exception](#)

## WriteEmptyLine()

Вывести пустую строку.

```
public void WriteEmptyLine()
```

## WriteError(string)

Вывести сообщение об ошибке.

```
public void WriteError(string message)
```

### Parameters

message [string](#)

## WriteErrorLines(params string[])

Вывести несколько сообщений об ошибке.

```
public void WriteErrorLines(params string[] messages)
```

### Parameters

messages [string](#)[]

## WriteException(Exception)

Вывести исключение.

```
public void WriteException(Exception exception)
```

### Parameters

exception [Exception](#)

## WriteInfo(string)

Вывести информационное сообщение.

```
public void WriteInfo(string message)
```

### Parameters

message [string](#)

## WriteLine(string)

Вывести текст с переносом строки.

```
public void WriteLine(string message)
```

### Parameters

message [string](#)

## WriteList<T>(IEnumerable<T>, Func<T, string>)

Вывести список с нумерацией.

```
public void WriteList<T>(IEnumerable<T> items, Func<T, string> format)
```

### Parameters

items [IEnumerable](#)<T>

Коллекция элементов.

format [Func](#)<T, [string](#)>

Функция форматирования элемента.

### Type Parameters

## T

Тип элемента.

## WriteMenu(string, params (string key, string description)[])

Вывести меню.

```
public void WriteMenu(string title, params (string key, string description)
[] options)
```

### Parameters

**title** [string](#)

Заголовок меню.

**options** ([string](#) [key](#), [string](#) [description](#))[]

Опции меню (ключ и описание).

## WriteSuccess(string)

Вывести сообщение об успешном выполнении.

```
public void WriteSuccess(string message)
```

### Parameters

**message** [string](#)

## WriteTable<T>(IEnumerable<T>, Func<T, string>)

Вывести таблицу из данных.

```
public void WriteTable<T>(IEnumerable<T> items, Func<T, string> format)
```

## Parameters

`items` [IEnumerable](#)<T>

Коллекция элементов.

`format` [Func](#)<T, string>

Функция форматирования элемента.

## Type Parameters

T

Тип элемента.

## WriteText(string)

Вывести текст без переноса строки.

```
public void WriteText(string message)
```

## Parameters

`message` string

## WriteWarning(string)

Вывести предупреждение.

```
public void WriteWarning(string message)
```

## Parameters

`message` string

# Namespace Presentation.Input

## Classes

### [ConsoleInput](#)

Реализация провайдера ввода информации через консоль

# Class ConsoleInput

Namespace: [Presentation.Input](#)

Assembly: Presentation.dll

Реализация провайдера ввода информации через консоль

```
public class ConsoleInput : IButtonInput, ITextInput, INumericInput,  
IPasswordInput, IInputProvider
```

## Inheritance

[object](#) ← ConsoleInput

## Implements

[IButtonInput](#), [ITextInput](#), [INumericInput](#), [IPasswordInput](#), [IInputProvider](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### ConsoleInput(IColoredOutput)

```
public ConsoleInput(IColoredOutput output)
```

## Parameters

output [IColoredOutput](#)

## Methods

### GetCheckedPassword()

Получить подтвержденный пароль (с проверкой совпадения)

```
public string GetCheckedPassword()
```

## Returns

[string](#)

## GetKeyFromSet(string, ConsoleKey, params ConsoleKey[])

Получить нажатую кнопку из набора

```
public ConsoleKey GetKeyFromSet(string prompt, ConsoleKey defaultKey = ConsoleKey.Y,  
params ConsoleKey[] allowedKeys)
```

## Parameters

`prompt` [string](#)

`defaultKey` [ConsoleKey](#)

`allowedKeys` [ConsoleKey](#)[]

## Returns

[ConsoleKey](#)

## GetLongText(string)

Получить многострочный текст

```
public string GetLongText(string prompt)
```

## Parameters

`prompt` [string](#)

## Returns

[string](#)

## GetNonEmptyText(string)

Получить текст с проверкой на пустоту

```
public string GetNonEmptyText(string prompt)
```

### Parameters

[prompt](#) [string](#)

### Returns

[string](#)

## GetNumeric(string)

Получить целое число без ограничений

```
public int GetNumeric(string prompt)
```

### Parameters

[prompt](#) [string](#)

### Returns

[int](#)

## GetNumericInRange(string, int, int)

Получить целое число в заданном диапазоне

```
public int GetNumericInRange(string prompt, int min, int max)
```

## Parameters

`prompt` `string`

`min` `int`

`max` `int`

## Returns

`int`

## GetNumericWithMax(string, int)

Получить целое число меньше или равно максимуму

```
public int GetNumericWithMax(string prompt, int max)
```

## Parameters

`prompt` `string`

`max` `int`

## Returns

`int`

## GetNumericWithMin(string, int)

Получить целое число больше или равно минимуму

```
public int GetNumericWithMin(string prompt, int min)
```

## Parameters

`prompt` `string`

`min` `int`

## Returns

[int ↗](#)

## GetPassword(string)

Получить пароль с маскированием символов

```
public string GetPassword(string prompt)
```

## Parameters

[prompt string ↗](#)

## Returns

[string ↗](#)

## GetPositiveNumeric(string)

Получить положительное целое число (>= 0)

```
public int GetPositiveNumeric(string prompt)
```

## Parameters

[prompt string ↗](#)

## Returns

[int ↗](#)

## GetSelectionFromDictionary(Dictionary<int, string>, string?, int)

Получить один вариант из словаря

```
public KeyValuePair<int, string> GetSelectionFromDictionary(Dictionary<int, string> options, string? title = null, int pageSize = 3)
```

## Parameters

options [Dictionary<int, string>](#)

title [string](#)

pageSize [int](#)

## Returns

[KeyValuePair<int, string>](#)

## GetSelectionFromList(IEnumerable<string>, string?, int)

Получить один вариант из списка

```
public string GetSelectionFromList(IEnumerable<string> options, string? title = null, int pageSize = 3)
```

## Parameters

options [IEnumerable<string>](#)

title [string](#)

pageSize [int](#)

## Returns

[string](#)

## GetShortText(string, bool)

Получить однострочный текст

```
public string GetShortText(string prompt, bool notNull = true)
```

## Parameters

`prompt` [string](#)

`notNull` [bool](#)

## Returns

[string](#)

## GetYesNoChoice(string)

Получить ответ да/нет от пользователя

```
public bool GetYesNoChoice(string prompt)
```

## Parameters

`prompt` [string](#)

## Returns

[bool](#)

## ValidatePasswordLength(string, int)

Проверить пароль на минимальную длину

```
public bool ValidatePasswordLength(string password, int minLength = 8)
```

## Parameters

`password` [string](#)

`minLength` [int](#)

## Returns

bool ↗

# Namespace Presentation.Input.Interfaces Interfaces

## [IButtonInput](#)

Интерфейс для операций ввода через кнопки и выбор вариантов

## [IIInputProvider](#)

Базовый интерфейс для провайдера ввода информации

## [INumericInput](#)

Интерфейс для операций ввода числовых значений

## [IPasswordInput](#)

Интерфейс для операций ввода пароля

## [ITextInput](#)

Интерфейс для операций ввода текста

# Interface IButtonInput

Namespace: [Presentation.Input.Interfaces](#)

Assembly: Presentation.dll

Интерфейс для операций ввода через кнопки и выбор вариантов

```
public interface IButtonInput
```

## Methods

`GetKeyFromSet(string, ConsoleKey, params ConsoleKey[])`

Получить нажатую кнопку из набора

```
ConsoleKey GetKeyFromSet(string prompt, ConsoleKey defaultKey = ConsoleKey.Y, params  
ConsoleKey[] allowedKeys)
```

### Parameters

`prompt` [string](#)

`defaultKey` [ConsoleKey](#)

`allowedKeys` [ConsoleKey\[\]](#)

### Returns

[ConsoleKey](#)

`GetSelectionFromDictionary(Dictionary<int, string>, string?, int)`

Получить один вариант из словаря

```
KeyValuePair<int, string> GetSelectionFromDictionary(Dictionary<int, string>
```

```
options, string? title = null, int pageSize = 3)
```

## Parameters

options [Dictionary<int, string>](#)

title [string](#)

pageSize [int](#)

## Returns

[KeyValuePair<int, string>](#)

## GetSelectionFromList(IEnumerable<string>, string?, int)

Получить один вариант из списка

```
string GetSelectionFromList(IEnumerable<string> options, string? title = null, int  
pageSize = 3)
```

## Parameters

options [IEnumerable<string>](#)

title [string](#)

pageSize [int](#)

## Returns

[string](#)

## GetYesNoChoice(string)

Получить ответ да/нет от пользователя

```
bool GetYesNoChoice(string prompt)
```

## Parameters

`prompt` [string](#)

## Returns

[bool](#)

# Interface IInputProvider

Namespace: [Presentation.Input.Interfaces](#)

Assembly: Presentation.dll

Базовый интерфейс для провайдера ввода информации

```
public interface IInputProvider
```

## Methods

### GetNumeric(string)

Получить целое число от пользователя

```
int GetNumeric(string prompt)
```

#### Parameters

**prompt** [string](#)

#### Returns

[int](#)

### GetPassword(string)

Получить пароль от пользователя

```
string GetPassword(string prompt)
```

#### Parameters

**prompt** [string](#)

#### Returns

[string](#)

## GetText(string)

Получить строку от пользователя

```
string GetText(string prompt)
```

### Parameters

**prompt** [string](#)

### Returns

[string](#)

# Interface INumericInput

Namespace: [Presentation.Input.Interfaces](#)

Assembly: Presentation.dll

Интерфейс для операций ввода числовых значений

```
public interface INumericInput
```

## Methods

### GetNumeric(string)

Получить целое число без ограничений

```
int GetNumeric(string prompt)
```

#### Parameters

**prompt** [string](#)

#### Returns

[int](#)

### GetNumericInRange(string, int, int)

Получить целое число в заданном диапазоне

```
int GetNumericInRange(string prompt, int min, int max)
```

#### Parameters

**prompt** [string](#)

**min** [int](#)

`max` [int ↗](#)

Returns

[int ↗](#)

## GetNumericWithMax(string, int)

Получить целое число меньше или равно максимуму

```
int GetNumericWithMax(string prompt, int max)
```

Parameters

`prompt` [string ↗](#)

`max` [int ↗](#)

Returns

[int ↗](#)

## GetNumericWithMin(string, int)

Получить целое число больше или равно минимуму

```
int GetNumericWithMin(string prompt, int min)
```

Parameters

`prompt` [string ↗](#)

`min` [int ↗](#)

Returns

[int ↗](#)

## GetPositiveNumeric(string)

Получить положительное целое число ( $\geq 0$ )

```
int GetPositiveNumeric(string prompt)
```

### Parameters

**prompt** string

### Returns

int

# Interface IPasswordInput

Namespace: [Presentation.Input.Interfaces](#)

Assembly: Presentation.dll

Интерфейс для операций ввода пароля

```
public interface IPasswordInput
```

## Methods

### GetCheckedPassword()

Получить подтвержденный пароль (с проверкой совпадения)

```
string GetCheckedPassword()
```

#### Returns

[string](#)

### GetPassword(string)

Получить пароль с маскированием символов

```
string GetPassword(string prompt)
```

#### Parameters

[prompt](#) [string](#)

#### Returns

[string](#)

## ValidatePasswordLength(string, int)

Проверить пароль на минимальную длину

```
bool ValidatePasswordLength(string password, int minLength = 8)
```

### Parameters

password [string](#)

minLength [int](#)

### Returns

[bool](#)

# Interface ITextInput

Namespace: [Presentation.Input.Interfaces](#)

Assembly: Presentation.dll

Интерфейс для операций ввода текста

```
public interface ITextInput
```

## Methods

### GetLongText(string)

Получить многострочный текст

```
string GetLongText(string prompt)
```

#### Parameters

**prompt** [string](#) ↗

#### Returns

[string](#) ↗

### GetNonEmptyText(string)

Получить текст с проверкой на пустоту

```
string GetNonEmptyText(string prompt)
```

#### Parameters

**prompt** [string](#) ↗

#### Returns

[string ↗](#)

## GetShortText(string, bool)

Получить однострочный текст

```
string GetShortText(string prompt, bool notNull = true)
```

### Parameters

[prompt string ↗](#)

[notNull bool ↗](#)

### Returns

[string ↗](#)

# Namespace Presentation.Output

## Classes

### [ConsoleOutput](#)

Реализация цветного вывода информации в консоль

# Class ConsoleOutput

Namespace: [Presentation.Output](#)

Assembly: Presentation.dll

Реализация цветного вывода информации в консоль

```
public class ConsoleOutput : IErrorOutput, IColoredOutput, IOutputProvider
```

## Inheritance

[object](#) ← ConsoleOutput

## Implements

[IErrorOutput](#), [IColoredOutput](#), [IOutputProvider](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### WriteColoredLine(string, ConsoleColor)

Вывести текст с указанным цветом и переводом на новую строку

```
public void WriteColoredLine(string message, ConsoleColor color)
```

## Parameters

message [string](#)

color [ConsoleColor](#)

### WriteColoredMessage(string, ConsoleColor)

Вывести текст с указанным цветом

```
public void WriteColoredMessage(string message, ConsoleColor color)
```

## Parameters

message [string](#)

color [ConsoleColor](#)

## WriteDetailedException(Exception)

Вывести детальную информацию об исключении с трассировкой стека

```
public void WriteDetailedException(Exception exception)
```

## Parameters

exception [Exception](#)

## WriteEmptyLine()

Вывести пустую строку

```
public void WriteEmptyLine()
```

## WriteError(string)

Вывести сообщение об ошибке

```
public void WriteError(string message)
```

## Parameters

message [string](#)

## WriteErrorLines(params string[])

Вывести несколько строк текста с логированием ошибок

```
public void WriteErrorLines(params string[] messages)
```

### Parameters

messages [string](#)[]

## WriteException(Exception)

Вывести информацию об исключении

```
public void WriteException(Exception exception)
```

### Parameters

exception [Exception](#)

## WriteInfo(string)

Вывести информацию (голубой цвет по умолчанию)

```
public void WriteInfo(string message)
```

### Parameters

message [string](#)

## WriteLine(string)

Вывести текст с новой строки

```
public void WriteLine(string message)
```

## Parameters

message [string](#)

## WriteSuccess(string)

Вывести успешное сообщение (зеленый цвет по умолчанию)

```
public void WriteSuccess(string message)
```

## Parameters

message [string](#)

## WriteText(string)

Вывести простой текст

```
public void WriteText(string message)
```

## Parameters

message [string](#)

## WriteWarning(string)

Вывести предупреждение (желтый цвет по умолчанию)

```
public void WriteWarning(string message)
```

## Parameters

message [string](#)

# Namespace Presentation.Output. Implementation

## Classes

[WriteToConsole](#)

# Class WriteToConsole

Namespace: [Presentation.Output.Implementation](#)

Assembly: Presentation.dll

```
public static class WriteToConsole
```

## Inheritance

[object](#) ← WriteToConsole

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### ColorMessage(string, ConsoleColor)

Writes the specified message to the console using the given text color.

```
public static void ColorMessage(string textError, ConsoleColor colorText  
= ConsoleColor.Red)
```

## Parameters

**textError** [string](#)

The message to display in the console output.

**colorText** [ConsoleColor](#)

The color to use for the message text. Defaults to [Red](#) if not specified.

## Remarks

This method temporarily changes the console's foreground color to the specified value while writing the message, and then restores the original color. Use this method to highlight error messages or other important information in the console output.

## LongText(params string[])

Displays each provided text string in a dark yellow color on the console.

```
public static void LongText(params string[] text)
```

### Parameters

text [string](#)[]

The variable-length array of strings to be displayed. Each string is printed on a new line.

### Remarks

This method is useful for highlighting messages in the console output, making them stand out visually.

## PrintPanel(string?, params string[])

Renders a panel to the console with optional header text and one or more lines of content.

```
public static void PrintPanel(string? header = null, params string[] textLines)
```

### Parameters

header [string](#)[]

The optional header text to display at the top of the panel. If null, the panel is rendered without a header.

textLines [string](#)[]

The lines of text to display within the panel. Each element represents a separate line.

### Remarks

The method uses AnsiConsole to output the panel. All provided text lines are joined with newline characters to form the panel's content.

## PrintTable(string[], IEnumerable<string[]>, string?)

Asynchronously displays a formatted table in the console with the specified columns and rows, and optionally includes a title above the table.

```
public static void PrintTable(string[] columns, IEnumerable<string[]> rows, string? title = null)
```

### Parameters

`columns` [string\[\]](#)

An array of strings that defines the names of the columns to display in the table. Each element represents a column header.

`rows` [IEnumerable<string\[\]>](#)

A collection of string arrays, where each array contains the values for a single row in the table. The number of elements in each row should match the number of columns.

`title` [string?](#)

An optional string specifying the title to display above the table. If null, no title is shown.

### Remarks

Await this method to ensure the table is fully rendered before executing subsequent operations. The method does not return any data and is intended for output purposes only.

## ProcExcept(Exception)

Processes and displays detailed information about an exception to the console, including the exception message, the method where the exception occurred, the stack trace, and any inner exception details.

```
public static void ProcExcept(Exception ex)
```

### Parameters

`ex` [Exception](#)

The exception to process and display. Cannot be null.

## Remarks

This method is useful for logging or debugging purposes, as it highlights critical exception details in different console colors for improved readability.

# Namespace Presentation.Output.Interfaces Interfaces

## [IColoredOutput](#)

Интерфейс для цветного вывода информации

## [IErrorOutput](#)

Интерфейс для вывода ошибок и исключений

## [IOOutputProvider](#)

Базовый интерфейс для провайдера вывода информации

# Interface IColoredOutput

Namespace: [Presentation.Output.Interfaces](#)

Assembly: Presentation.dll

Интерфейс для цветного вывода информации

```
public interface IColoredOutput : IOutputProvider
```

## Inherited Members

[IOutputProvider.WriteLine\(string\)](#) , [IOutputProvider.WriteLine\(string\)](#) ,  
[IOutputProvider.WriteLine\(\)](#)

## Methods

### WriteColoredLine(string, ConsoleColor)

Вывести текст с указанным цветом и переводом на новую строку

```
void WriteColoredLine(string message, ConsoleColor color)
```

#### Parameters

message [string](#)

color [ConsoleColor](#)

### WriteColoredMessage(string, ConsoleColor)

Вывести текст с указанным цветом

```
void WriteColoredMessage(string message, ConsoleColor color)
```

#### Parameters

message [string](#)

`color` [ConsoleColor](#)

## WriteInfo(string)

Вывести информацию (голубой цвет по умолчанию)

```
void WriteInfo(string message)
```

### Parameters

`message` [string](#)

## WriteSuccess(string)

Вывести успешное сообщение (зеленый цвет по умолчанию)

```
void WriteSuccess(string message)
```

### Parameters

`message` [string](#)

## WriteWarning(string)

Вывести предупреждение (желтый цвет по умолчанию)

```
void WriteWarning(string message)
```

### Parameters

`message` [string](#)

# Interface IErrorOutput

Namespace: [Presentation.Output.Interfaces](#)

Assembly: Presentation.dll

Интерфейс для вывода ошибок и исключений

```
public interface IErrorOutput : IColoredOutput, IOutputProvider
```

## Inherited Members

[IColoredOutput.WriteColoredMessage\(string, ConsoleColor\)](#) ,  
[IColoredOutput.WriteColoredLine\(string, ConsoleColor\)](#) , [IColoredOutput.WriteSuccess\(string\)](#) ,  
[IColoredOutput.WriteWarning\(string\)](#) , [IColoredOutput.WriteInfo\(string\)](#) ,  
[IOutputProvider.WriteLine\(string\)](#) , [IOutputProvider.WriteLine\(string\)](#) ,  
[IOutputProvider.WriteLine\(\)](#)

## Methods

### WriteDetailedException(Exception)

Вывести детальную информацию об исключении с трассировкой стека

```
void WriteDetailedException(Exception exception)
```

#### Parameters

exception [Exception](#)

### WriteError(string)

Вывести сообщение об ошибке

```
void WriteError(string message)
```

#### Parameters

message [string](#)

## WriteErrorLines(params string[])

Вывести несколько строк текста с логированием ошибок

```
void WriteErrorLines(params string[] messages)
```

### Parameters

messages [string](#)[]

## WriteException(Exception)

Вывести информацию об исключении

```
void WriteException(Exception exception)
```

### Parameters

exception [Exception](#)

# Interface IOutputProvider

Namespace: [Presentation.Output.Interfaces](#)

Assembly: Presentation.dll

Базовый интерфейс для провайдера вывода информации

```
public interface IOutputProvider
```

## Methods

### WriteEmptyLine()

Вывести пустую строку

```
void WriteEmptyLine()
```

### WriteLine(string)

Вывести текст с новой строки

```
void WriteLine(string message)
```

#### Parameters

message [string](#)

### WriteText(string)

Вывести простой текст

```
void WriteText(string message)
```

#### Parameters

message [string](#)

# Namespace Program

## Classes

### Program

Главный класс программы. Содержит точку входа и инициализацию зависимостей.

# Class Program

Namespace: [Program](#)

Assembly: Program.dll

Главный класс программы. Содержит точку входа и инициализацию зависимостей.

```
public static class Program
```

## Inheritance

[object](#) ← Program

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Methods

## Main(string[])

Точка входа в программу.

```
public static Task<int> Main(string[] args)
```

## Parameters

args [string](#)[]

Аргументы командной строки.

## Returns

[Task](#)<[int](#)>

Код завершения программы.

# Namespace ShevricTodo.Formats

## Classes

[FileSerializationFormat](#)

[Json<T>](#)

[MessagePack<T>](#)

[ProtoBuf<T>](#)

[Xml<T>](#)

## Interfaces

[ISerializationFormat<T>](#)

# Class FileSerializationFormat

Namespace: [ShevricTodo.Formats](#)

Assembly: Infrastructure.dll

```
public abstract class FileSerializationFormat
```

## Inheritance

[object](#) ← FileSerializationFormat

## Derived

[Json<T>](#), [MessagePack<T>](#), [ProtoBuf<T>](#), [Xml<T>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Fields

### Path

```
public string? Path
```

### Field Value

[string](#)

## Methods

### StringInfo(string)

```
public static string StringInfo(string path)
```

### Parameters

path [string](#)

Returns

[string](#)

## StringInfoAsync()

```
public Task<string> StringInfoAsync()
```

Returns

[Task](#)<[string](#)>

# Interface ISerializationFormat<T>

Namespace: [ShevricTodo.Formats](#)

Assembly: Infrastructure.dll

```
public interface ISerializationFormat<T>
```

## Type Parameters

T

## Methods

### DeserializationAsync()

```
Task<T?> DeserializationAsync()
```

#### Returns

[Task](#)<T>

### SerializationAsync(T)

```
Task SerializationAsync(T values)
```

#### Parameters

values T

#### Returns

[Task](#)<T>

## StringInfoAsync()

Task<[string](#)> **StringInfoAsync()**

### Returns

[Task](#)<[string](#)>

# Class Json<T>

Namespace: [ShevricTodo.Formats](#)

Assembly: Infrastructure.dll

```
public class Json<T> : FileSerializationFormat, ISerializationFormat<T>
```

## Type Parameters

T

### Inheritance

[object](#) ↗ ← [FileSerializationFormat](#) ← [Json<T>](#)

### Implements

[ISerializationFormat](#)<T>

### Inherited Members

[FileSerializationFormat.Path](#) , [FileSerializationFormat.StringInfo\(string\)](#) ,  
[FileSerializationFormat.StringInfoAsync\(\)](#) , [object.Equals\(object\)](#) ↗ , [object.Equals\(object, object\)](#) ↗ ,  
[object.GetHashCode\(\)](#) ↗ , [object.GetType\(\)](#) ↗ , [object.MemberwiseClone\(\)](#) ↗ ,  
[object.ReferenceEquals\(object, object\)](#) ↗ , [object.ToString\(\)](#) ↗

## Constructors

### Json(string)

```
public Json(string path)
```

### Parameters

path [string](#) ↗

## Fields

### FileExtension

```
public const string FileExtension = ".json"
```

## Field Value

[string](#)

## SerializerOptions

```
public JsonSerializerOptions SerializerOptions
```

## Field Value

[JsonSerializerOptions](#)

## Methods

### DeserializationAsync()

```
public Task<T?> DeserializationAsync()
```

#### Returns

[Task](#)<T>

### DeserializationAsync(string)

```
public static Task<T?> DeserializationAsync(string path)
```

#### Parameters

path [string](#)

#### Returns

[Task](#) <T>

## SerializationAsync(T)

```
public Task SerializationAsync(T value)
```

### Parameters

value T

### Returns

[Task](#)

## SerializationAsync(T, string, JsonSerializerOptions?)

```
public static Task SerializationAsync(T value, string path, JsonSerializerOptions?  
serializerOptions = null)
```

### Parameters

value T

path [string](#)

serializerOptions [JsonSerializerOptions](#)

### Returns

[Task](#)

# Class MessagePack<T>

Namespace: [ShevricTodo.Formats](#)

Assembly: Infrastructure.dll

```
public class MessagePack<T> : FileSerializationFormat, ISerializationFormat<T>
```

## Type Parameters

T

### Inheritance

[object](#) ↴ ← [FileSerializationFormat](#) ← MessagePack<T>

### Implements

[ISerializationFormat](#)<T>

### Inherited Members

[FileSerializationFormat.Path](#) , [FileSerializationFormat.StringInfo\(string\)](#) ,  
[FileSerializationFormat.StringInfoAsync\(\)](#) , [object.Equals\(object\)](#) ↴ , [object.Equals\(object, object\)](#) ↴ ,  
[object.GetHashCode\(\)](#) ↴ , [object.GetType\(\)](#) ↴ , [object.MemberwiseClone\(\)](#) ↴ ,  
[object.ReferenceEquals\(object, object\)](#) ↴ , [object.ToString\(\)](#) ↴

## Fields

### FileExtension

```
public const string FileExtension = ".bin"
```

### Field Value

[string](#) ↴

## Methods

### DeserializationAsync()

```
public Task<T?> DeserializationAsync()
```

## Returns

[Task](#)<T>

## DeserializationAsync(string)

```
public static Task<T?> DeserializationAsync(string path)
```

## Parameters

path [string](#)

## Returns

[Task](#)<T>

## SerializationAsync(T)

```
public Task SerializationAsync(T value)
```

## Parameters

value T

## Returns

[Task](#)

## SerializationAsync(T, string)

```
public static Task SerializationAsync(T value, string path)
```

## Parameters

**value** T

**path** [string](#) ↗

## Returns

[Task](#) ↗

# Class ProtoBuf<T>

Namespace: [ShevricTodo.Formats](#)

Assembly: Infrastructure.dll

```
public class ProtoBuf<T> : FileSerializationFormat, ISerializationFormat<T>
```

## Type Parameters

T

### Inheritance

[object](#) ↴ ← [FileSerializationFormat](#) ← ProtoBuf<T>

### Implements

[ISerializationFormat](#)<T>

### Inherited Members

[FileSerializationFormat.Path](#) , [FileSerializationFormat.StringInfo\(string\)](#) ,  
[FileSerializationFormat.StringInfoAsync\(\)](#) , [object.Equals\(object\)](#) ↴ , [object.Equals\(object, object\)](#) ↴ ,  
[object.GetHashCode\(\)](#) ↴ , [object.GetType\(\)](#) ↴ , [object.MemberwiseClone\(\)](#) ↴ ,  
[object.ReferenceEquals\(object, object\)](#) ↴ , [object.ToString\(\)](#) ↴

## Fields

### FileExtension

```
public const string FileExtension = ".bin"
```

### Field Value

[string](#) ↴

## Methods

### DeserializationAsync()

```
public Task<T?> DeserializationAsync()
```

## Returns

[Task](#)<T>

## DeserializationAsync(string)

```
public static Task<T?> DeserializationAsync(string path)
```

## Parameters

path [string](#)

## Returns

[Task](#)<T>

## SerializationAsync(T)

```
public Task SerializationAsync(T value)
```

## Parameters

value T

## Returns

[Task](#)

## SerializationAsync(T, string)

```
public static Task SerializationAsync(T value, string path)
```

## Parameters

**value** T

**path** [string](#) ↗

## Returns

[Task](#) ↗

# Class Xml<T>

Namespace: [ShevricTodo.Formats](#)

Assembly: Infrastructure.dll

```
public class Xml<T> : FileSerializationFormat, ISerializationFormat<T>
```

## Type Parameters

T

### Inheritance

[object](#) ↴ ← [FileSerializationFormat](#) ← Xml<T>

### Implements

[ISerializationFormat](#)<T>

### Inherited Members

[FileSerializationFormat.Path](#) , [FileSerializationFormat.StringInfo\(string\)](#) ,  
[FileSerializationFormat.StringInfoAsync\(\)](#) , [object.Equals\(object\)](#) ↴ , [object.Equals\(object, object\)](#) ↴ ,  
[object.GetHashCode\(\)](#) ↴ , [object.GetType\(\)](#) ↴ , [object.MemberwiseClone\(\)](#) ↴ ,  
[object.ReferenceEquals\(object, object\)](#) ↴ , [object.ToString\(\)](#) ↴

## Fields

### FileExtension

```
public const string FileExtension = ".xml"
```

### Field Value

[string](#) ↴

## Methods

### DeserializationAsync()

```
public Task<T?> DeserializationAsync()
```

## Returns

[Task](#)<T>

## DeserializationAsync(string)

```
public static Task<T?> DeserializationAsync(string path)
```

## Parameters

path [string](#)

## Returns

[Task](#)<T>

## SerializationAsync(T)

```
public Task SerializationAsync(T value)
```

## Parameters

value T

## Returns

[Task](#)

## SerializationAsync(T, string)

```
public static Task SerializationAsync(T value, string path)
```

## Parameters

**value** T

**path** [string](#) ↗

## Returns

[Task](#) ↗