Программа курса "Современный .NET/C#".

**Введение в .NET**

**Обзор NET Framework.**

* NET Standarl/.NET Framework/.NET Core/.NET
* Приложения, которые могут быть разработаны с использованием NET
* MSIL.
* Время выполнения общего языка (CLR)
* Управляемый и неуправляемый код.
* Общая система типов (CTS).
* Система общего языка (CLS).
* Компиляция.
  + JIT.
  + OSR (On Stack Replacement)
  + AOT.
  + PGO.
  + SIMD.

**-NET Основная точка входа, представление проекта и решения.**

* Обзор проекта и решения NET.
* Метод точки входа - Main (Program.cs).
* Восстановление, сборка, запуск приложения и упаковка приложения.
* Код выхода приложения.
* Запуск приложения с использованием входных аргументов.

**-Синтаксис языка NET/C#.**

* Структура программы.
* Операторы верхнего уровня - программа без метода Main.
* Переменные и типы данных
* Типы Псевдонимы
* Строки.
  + Форматирование,
  + Многострочная строка.
  + Строковые литералы UTF-8.
* Типы значений и ссылочные типы.
* Нулевые типы.

© Операторы Null.

Введение в HEAP и STACK-память.

Где выделяются переменные?

Неявное и явное приведение,

Кастинг и деиндексация,

Перечисления и ограничения

Операторы.

Массивы.

Инлайновые массивы.

Циклы.

© For.

© Foreach

© While,

© Do-While.

Булевы операторы.

e Тернарный оператор.

© Оператор If.

'© Switch - оператор.

'© Switch - выражение

Методы

Делегаты

© Лямбда-выражения.

Значения по умолчанию для параметров лямбда-выражений.

Статический модификатор для лямбда-выражения

Входные параметры: in, out, ref.

Объектно-ориентированное программирование (ООП).

Понятие ООП.

© Что такое пространство имен.

e Структура.

© Класс,

Свойства (get, set, init).

Свойства, доступные только для чтения.

Необходимые свойства.

Частичный класс и методы.

Герметичные класс и метод.

Статический класс, поля и методы.

Перегрузка методов,

Перегрузка операторов.

Методы расширения.

Анонимные типы.

Кортежи.

Первичные конструкторы.

© Статические конструкторы.

© Абстрактный класс.

© Virtual, Override, New, Abstract.

© Интерфейс,

Реализация методов по умолчанию.

© Статические члены.

Модификатор доступа.

Перехватчики.

Особенности ООП.

Наследование.

Полиморфизм.

Абстрагирование.

Инкапсуляция.

Функциональные техники.

Сопоставление шаблонов,

Проверка нулей.

Тесты типов.

Реляционные шаблоны.

Множественные входы.

Списочные шаблоны.

Отбрасывания.

Деконструкция типов.

Исключения/ошибки и обработка ошибок.

Исключения, бросок исключения. Встроенные исключения

Блок Try...Catch...Finally...

Пользовательские исключения.

Исключения, генерируемые компилятором.

SOLID.

Принцип единой ответственности (SRP).

Принцип открытости/закрытости (OCP).

Принцип замещения Лискова (LSP).

Принцип разделения интерфейсов (ISP).

Принцип инверсии зависимостей (DIP).

Дженерики.

© Generic Types.

© Ковариантность и контравариантность.

-NET Collections.

Введение в NET-коллекции.

Различные типы коллекций.

Коллекция.

IList

[Перечислимый.

Iterator.

Indexator.

Dictionary [Словарь.

List.

Очередь.

SortedList.

Стек.

ArrayList.

Хэш-таблица.

Stack.

Неизменяемые коллекции.

Сложность коллекций.

Отражение и атрибуты.

Введение.

Создание пользовательских атрибутов.

Generic и атрибуты.

Введение в Reflection.

Generic и Reflection

Класс сборки.

JSON.

© Введение в JSON.

System.Text.Json

Генерация кода Json

Regex.

Введение в Regex

Кэширование Regex

Генерация кода Regex.

Делегаты и события.

Что такое делегаты.

Сильно типизированные делегаты.

Общие паттерны для делегатов.

Введение в события.

Стандартные шаблоны событий NET.

Различие между делегатами и событиями.

Управляемый и неуправляемый код/ресурсы.

Что такое управляемые и неуправляемые ресурсы.

Автоматическое управление памятью.

Что такое GC и как оно работает.

Утечка памяти в .NET.

Что такое паттерн Dispose. Финализатор и деконструктор.

Что такое Stackallock, Span или как написать код с нулевым выделением.

LING.

© Что такое LINQ (Language-Integrated Query)

\* Отложенные и немедленные запросы.

\* Проекция (Select, SelectMany).

© Фильтр (Where).

© Сортировка.

© Агрегирование.

© Skip, Take.

© Группировка.

© Join

# © All, Any, Contains, First, Single, Last, Distinct.

Асинхронное программирование. Потоки и задачи.

¢ Основы управляемых потоков.

© Потоки и потокообразование.

© Планирование потоков.

© Отмена потоков.

© Уничтожение потоков.

© Исключения в управляемых потоках

Таймеры.

© Асинхронное программирование,

¢ Введение в асинхронное программирование

Сценарии асинхронного программирования.

'Модель асинхронного программирования задач

Асинхронные типы возврата.

Отмена задач.

Обработка асинхронных задач по мере их завершения.

Использование ValueTask.

TAP во время выполнения и под капотом.

Параллелизм и параллелизм.

\* lAsyncEnumerable (неблокирующее перечисление).

lAsyncEnumerable - отмена асинхронных итераций.

\AsyncEnumerable - под капотом.

База данных. SQL. PostgreSQL.

Фундаментальные понятия SQL.

Что такое SQL и зачем он нужен?

Разновидности SQL: Postgres vs SQL Server и т. д.

Таблицы, строки и столбцы базы данных.

SQL-запросы

'© SELECT. Выбор всех столбцов или определенных столбцов.

Ограничение записей.

Упорядочивание записей.

Различимые записи.

Операторы сравнения:

Фильтрация записей.

Связь между таблицами