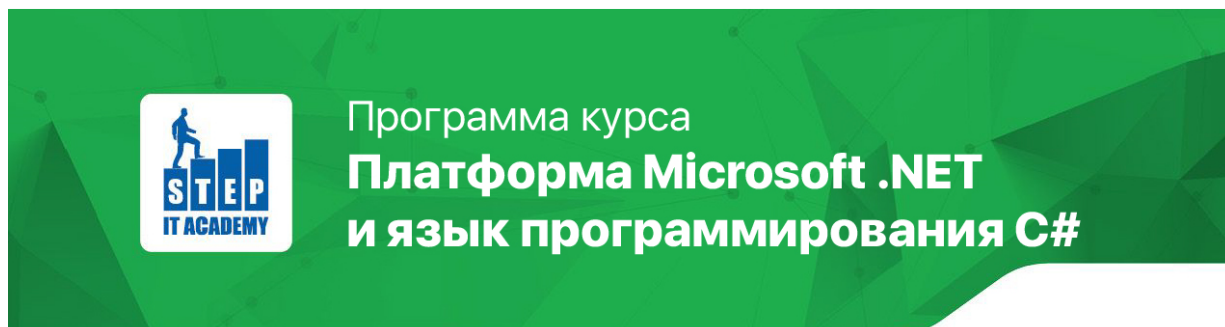


Microsoft .NET Framework and C# Programming Language

Lessons



Для групп стационара

Версия 4.0.0

Продолжительность курса – 42 пары (21 день)

Цель курса

Обучить слушателя основам разработки приложений с использованием платформы Microsoft .NET и языка программирования C#. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

По окончании курса слушатель будет:

- Понимать причины возникновения платформы Microsoft .NET
- Оперировать базовыми терминами платформы Microsoft .NET: CLR, CLS, CTS, BCL
- Уметь использовать рефлексоры и дотфускаторы
- Разбираться в тонкостях реализации ООП в C#
- Уметь создавать классы пользовательских исключений
- Уметь создавать пользовательские делегаты и события
- Взаимодействовать со сборщиком мусора
- Уметь использовать механизмы сериализации
- Изучит основы использования LINQ
- Изучит основы использования паттернов проектирования и UML

По окончании данного курса студент сдаёт практическое задание и теоретический экзамен по материалам курса. Для допуска к экзамену, должны быть

ных разделов курса.

Тематический план

Модуль 1. Введение в платформу Microsoft .NET. Основы языка программирования C#	2 пары
Модуль 2. Массивы и строки	2 пары
Модуль 3. Top level statements. Введение в классы. Обработка исключений	4 пары
Модуль 4. Пространства имен. Структуры и перечисления. Nullable типы	2 пары
Модуль 5. Перегрузка операторов. Индексаторы и свойства	2 пары
Модуль 6. Наследование	2 пары
Модуль 7. Интерфейсы	2 пары
Модуль 8. Делегаты, события, записи	4 пары
Модуль 9. Введение в Generics	4 пары
Модуль 10. Сборка мусора	2 пары
Модуль 11. Взаимодействие с файловой системой	2 пары
Модуль 12. Введение в LINQ	4 пары
Модуль 13. Сериализация объектов. Логирование	4 пары
Модуль 14. Введение в UML и паттерны проектирования	4 пары

Модуль 1.

Введение в платформу Microsoft .NET. Основы языка программирования C#

1. Введение в платформу Microsoft .NET.

- История и этапы развития технологий программирования.
- Причины возникновения платформы Microsoft .NET.
- Сравнительный анализ преимуществ и недостатков платформы Microsoft .NET.

2. Базовые понятия платформы Microsoft .NET.

- Архитектура платформы Microsoft .NET.
- Общеязыковая среда исполнения CLR (common language runtime).
- Стандартная система типов CTS (common type system).
- Стандартная языковая спецификация CLS (common language specification).
- Библиотека классов FCL (BCL).
- Языки платформы Microsoft .NET.
- Схема компиляции и исполнения приложения платформы Microsoft .NET.
- Язык MSIL (Microsoft Intermediate Language).
- Понятия метаданных, манифеста, сборки.

3. .NET Core, .NET Standard.

- Что такое .NET Core?
- Цели и задачи .NET Core
- Что такое .NET Standard?
- Цели и задачи .NET Standard
- Сравнение .NET Core, .NET Standard, .NET Framework

4. Введение в язык программирования C#.

- Плюсы и минусы языка программирования C#.
- Простейшая программа на языке программирования C#.

Программа курса Платформа Microsoft .NET и язык программирования C# Версия 4.0.0



- Необходимость использования рефлексора.
- Обзор существующих рефлексоров.
- Что такое дотфускатор?
- Необходимость использования дотфускаторов.
- Обзор существующих дотфускаторов.

6. Типы данных.

- Целочисленные типы данных.
- Типы данных для чисел с плавающей точкой.
- Символьный тип данных.
- Другие типы данных.

7. Литералы.

8. Переменные.

- Понятие переменной.
- Правила именования переменных.
- Область видимости переменных.

9. Анонимные типы.

10. Ввод, вывод в консольном приложении.

11. Структурные и ссылочные типы.

12. Преобразование типов.

- Явное преобразование.
- Неявное преобразование.

13. Операторы.

- Арифметические операторы.
- Операторы отношений.
- Логические операторы.
- Битовые операторы.
- Оператор присваивания.
- Приоритет операторов.

14. Условия.

- Условный оператор if.
- Условный оператор if else.
- switch:
 - switch statement
 - switch expression
- Оператор ?:

Программа курса Платформа Microsoft .NET и язык программирования C# Версия 4.0.0



- Цикл while.
- Цикл do while.
- Цикл foreach.
- Инструкция break.
- Инструкция continue.
- Инструкция goto.

16. Использование nameoff.

Модуль 2.

Массивы и строки

1. Массивы.

- Одномерные массивы.
- Многомерные массивы.
- Рваные массивы.
- Использование цикла foreach.

2. Null-conditional и Null-coalescing операторы.

3. Строки.

- Создание строки.
- Операции со строками.
- Особенности использования строк.

4. Использование аргументов командной строки.

Модуль 3.

Top level statements. Введение в классы.

Обработка исключений

1. Top level statements.

- Что такое top level statements?
- Правила по использованию top level statements.
- Работа с командной строкой.
- Практические примеры.

2. Синтаксис объявления класса.

3. Спецификаторы доступа языка программирования C#.

4. Поля класса.

5. Методы класса.

- Передача параметров.

Программа курса Платформа Microsoft .NET и язык программирования C# Версия 4.0.0



- Реализация тела метода в виде выражения.

6. Конструкторы.

- Понятие конструктора.
- Параметризованный конструктор.
- Перегруженные конструкторы.
- Статические конструкторы.

7. Ключевое слово `this`.

8. Использование `ref` и `out` параметров.

- Использование модификатора `ref`.
- Использование модификатора `out`.
- Кортежи.

9. Создание методов с переменным количеством аргументов.

10. Деконструкторы.

11. Частичные типы (`partial types`).

12. Иерархия исключений.

- Базовый класс `System.Exception`.
- Анализ иерархии стандартных исключений.

13. Основы обработки исключений.

- Ключевое слово `try`.
- Ключевое слово `catch`.
- Ключевое слово `throw`.
- Ключевое слово `finally`.

14. Тонкости обработки исключений.

- Перехват всех исключений.
- Вложенные блоки `try`.
- Повторное генерирование исключений.

15. Применение конструкций `checked` и `unchecked`.

16. Фильтры исключений.

Модуль 4.

Пространства имен. Структуры и перечисления. `Nullable` типы.

1. Пространства имен.

- Что такое пространство имен?

Программа курса Платформа Microsoft .NET и язык программирования C# Версия 4.0.0



- Вложенные пространства имен.
- Разбиение пространства имен на части.
- Пространство имен по умолчанию.
- Вторая форма using.
- Использование using для подключения статических членов.

2. Структуры, перечисления.

- Структуры:
 - Понятие структуры.
 - Синтаксис объявления структуры.
 - Необходимость и особенности применения структур.
 - Конструктор без параметров и структуры
- Перечисления (enum):
 - Понятие перечисления.
 - Синтаксис объявления перечисления.
 - Необходимость и особенности применения перечисления.
 - Установка базового типа перечисления.
 - Использование методов для перечислений.

3. Nullable типы.

- Что такое Nullable тип.
- Цели и задачи Nullable типов.
- Операции доступные для Nullable типов.
- Примеры использования.

Модуль 5.

Перегрузка операторов. Индексаторы и свойства

1. Перегрузка операторов.

- Введение в перегрузку операторов.
- Перегрузка унарных операторов.
- Перегрузка бинарных операторов.
- Перегрузка операторов отношений.
- Перегрузка логических операторов.
- Перегрузка операторов true и false.
- Перегрузка операторов преобразования.

2. Индексаторы.

- Перегрузка индексаторов.

3. Свойства.

- Что такое свойства?
- Синтаксис объявления свойств.
- get, set, init.
- Краткий синтаксис однострочных свойств.
- Примеры использования свойств.
- Автоматические свойства (auto-property):
 - Что такое автоматические свойства?
 - Примеры использования автоматических свойств.
 - Инициализация автоматических свойств.

Модуль 6.

Наследование

1. Наследование в C#.

- Анализ механизма наследования в C#.
- Спецификаторы доступа при наследовании.
- Особенности использования конструкторов при наследовании.
- Соккрытие имен при наследовании.
- Ключевое слово base.
- Наследование и исключения.
- Наследование от стандартных классов исключений.

2. Использование ключевого слова sealed.

3. Использование ссылок на базовый класс.

4. Виртуальные методы.

- Что такое виртуальный метод?
- Необходимость использования виртуальных методов.
- Переопределение виртуальных методов.

5. Абстрактный класс.

6. Анализ базового класса Object.

7. Упаковка, распаковка (boxing, unboxing).

Модуль 7.

Интерфейсы

1. Понятие интерфейса.
2. Синтаксис объявления интерфейсов.

Содержание модуля 8. Делегаты, события, записи

6. Наследование интерфейсов.

7. Проблемы сокрытия имен при наследовании интерфейсов.

8. Анализ стандартных интерфейсов.

Модуль 8.

Делегаты, события, записи

1. Делегаты.

- Понятие делегата.
- Синтаксис объявления делегата.
- Цели и задачи делегатов.
- Вызов нескольких методов через делегат (multicasting).
- Базовые классы для делегатов:
 - System.Delegate,
 - System.MulticastDelegate.

2. События.

- Понятие события.
- Синтаксис объявления события.
- Необходимость и особенности применения событий.
- Применение события для многоадресного делегата.
- Использование событийных средств доступа.

3. Анонимные методы.

4. Лямбда выражения.

5. Extension методы.

6. Записи (records).

- Что такое запись(record)?
- Синтаксис объявления записи.
- Сравнение записи и класса.
- Принципы работы операторов сравнения, Equals, GetHashCode для записей.
- Наследование записей.
- Практические примеры использования.

Модуль 9.

Введение в Generics

1. Generics.

Программа курса Платформа Microsoft .NET и язык программирования C# Версия 4.0.0



- Создание generic классов.
- Вложенные типы внутри generic класса.
- Использование ограничений.
- Создание generic интерфейсов.
- Создание generic делегатов.
- Создание generic методов.

2. Итераторы.

- Что такое итератор?
- Синтаксис и примеры использования итераторов.

3. Коллекции.

- Понятие коллекции.
- Generic коллекции.
- Классы коллекций `List<T>`, `Dictionary<TKey,TValue>`, `Stack<T>`, `Queue<T>`, `SortedList<TKey, TValue>` и другие.
- Интерфейсы коллекций `ICollection<T>`, `IEnumerator<T>`, `IEnumerable<T>`, `ICollection<T>`, `IDictionary<TKey,TValue>`, `Comparer<T>`, ...
- Примеры использования классов generic коллекций.

Модуль 10.

Сборка мусора

1. Жизненный цикл объектов.
2. Понятие сборщика мусора.
3. Деструктор и метод `Finalize`.
4. Метод `Dispose` и интерфейс `IDisposable`.
5. Класс `System.GC`.
6. Понятие поколений при сборке мусора.

Модуль 11.

Взаимодействие с файловой системой

1. Модель потоков в C#. Пространство `System.IO`.
2. Класс `Stream`.
3. Анализ байтовых классов потоков.
4. Анализ символьных классов потоков.
5. Анализ двоичных классов потоков.
6. Использование класса `FileStream` для файловых операций.
7. Использование класса `StreamWriter` для файловых операций.
8. Использование класса `StreamReader` для файловых операций.
9. Использование класса `BinaryWriter` для файловых операций.

12. Регулярные выражения.

Модуль 12.

Введение в LINQ

1. Что такое LINQ?

2. Цели и задачи LINQ.

3. Понятие запроса.

- Запрос в LINQ.
- Синтаксис запроса.
- Исполнение запроса.
- Сортировка.
- Группировка.
- Другие операции.

4. Использование LINQ и коллекций.

5. Примеры использования.

Модуль 13.

Сериализация объектов. Логирование

1. Понятие атрибутов.

2. Что такое сериализация?

3. Отношения между объектами.

4. Графы отношений объектов.

5. Атрибуты для сериализации [Serializable] и [NonSerialized].

6. Форматы сериализации.

- Пространство System.Runtime.Serialization.Formatters.
- Двоичное форматирование. Класс BinaryFormatter.
- Soap форматирование. Класс SoapFormatter.
- Примеры использования сериализации.
- Создание пользовательского формата сериализации. Интерфейс ISerializable.

7. Что такое логирование?

8. Цели и задачи логирования.

9. Ситуации, требующие логирования.

10. Инструменты логирования.

- log4net
- Serilog

Модуль 14.

Введение в UML и паттерны проектирования

1. UML

- Что такое UML?
- Цели и задачи диаграмм UML.
- Анализ диаграммы классов (class diagrams).
- Анализ диаграммы вариантов использования (use case diagram).
- Анализ диаграммы последовательности (sequence diagram).
- Краткий обзор других диаграмм

2. Паттерны проектирования.

- Что такое паттерны проектирования?
- Цели и задачи паттернов проектирования.
- Группы паттернов проектирования:
 - Порождающие паттерны;
 - Структурные паттерны;
 - Поведенческие паттерны (паттерны поведения).
- Анализ паттернов:
 - Singleton:
 - Цель паттерна;
 - Причины возникновения паттерна;
 - Структура паттерна;
 - Результаты использования паттерна;
 - Практический пример использования паттерна.
 - Factory Method:
 - Цель паттерна;
 - Причины возникновения паттерна;
 - Структура паттерна;
 - Результаты использования паттерна;
 - Практический пример использования паттерна.
 - Builder:
 - Цель паттерна;
 - Причины возникновения паттерна;
 - Структура паттерна;
 - Результаты использования паттерна;

- Цель паттерна;
 - Причины возникновения паттерна;
 - Структура паттерна;
 - Результаты использования паттерна;
 - Практический пример использования паттерна.
- Facade:
 - Цель паттерна;
 - Причины возникновения паттерна;
 - Структура паттерна;
 - Результаты использования паттерна;
 - Практический пример использования паттерна.
 - Observer:
 - Цель паттерна;
 - Причины возникновения паттерна;
 - Структура паттерна;
 - Результаты использования паттерна;
 - Практический пример использования паттерна.
 - Strategy:
 - Цель паттерна;
 - Причины возникновения паттерна;
 - Структура паттерна;
 - Результаты использования паттерна;
 - Практический пример использования паттерна.

Модуль 15.

Экзамен

Программа курса Платформа Microsoft .NET и язык программирования C# Версия 4.0.0



the Act of "On Copyright and Related Rights". The scope and method of the cited works are in accordance with the adopted norms, without prejudice to the normal exploitation of copyright, and do not prejudice the legitimate interests of authors and right holders. At the time of use, the cited works fragments cannot be replaced by alternative, non-copyrighted counterparts and meet the criteria for fair use. All rights reserved. Any reproduction of the materials or its part is prohibited. Use of the works or their fragments must be agreed upon with authors and rights holders. Agreed material use is only possible with reference to the source. Responsibility for unauthorized copying and commercial use of the material is determined by the current legislation.