

20483 В: Программирование на С#

20483 В: Программирование на С#

Содержание

Краткая информация	2
Обзор.....	2
О курсе	2
Профиль аудитории.....	2
По окончании курса.....	2
Детальная информация о курсе.....	2
Предварительные требования	6
Дополнительная информация.....	6

Краткая информация

Длительность:	5 дней (40 ак.часов)
Аудитория:	ИТ-профессионалы
Технология:	Visual Studio 2012
Тип:	Учебный курс
Способ обучения:	Под руководством инструктора

Обзор

О курсе

Курс позволяет слушателям приобрести знания и навыки программирования, необходимые для разработки Windows приложений с использованием языка программирования C#. Курс охватывает основы построения программ на языке C#, синтаксис языка и детали реализации приложений на платформе .NET Framework 4.5, а также знакомит слушателей с методами и технологиями, используемыми в современных настольных и корпоративных приложениях. В курсе рассматриваются обработка событий, программирование пользовательского интерфейса, доступ к базам данных, асинхронное выполнение операций, создание пользовательских атрибутов и шифрование/дешифрование данных. Курс использует среду разработки Visual Studio 2012 на платформе Windows 8.

Профиль аудитории

Этот курс предназначен для опытных разработчиков, которые уже имеют опыт программирования в C, C++, JavaScript, Objective-C, Microsoft Visual Basic или Java и понимают концепции объектно-ориентированного программирования.

По окончании курса

По окончании курса слушатели смогут:

- Описать синтаксис и основные особенности C#;
- Создавать и вызывать методы, перехватывать и обрабатывать исключения, описывать требования к мониторингу масштабируемых приложений;
- Реализовать базовую структуру и основные элементы типичного настольного приложения;
- Создавать классы, определять и реализовать интерфейсы, а также создавать и использовать коллекции;
- Использовать наследование для создания иерархии классов, расширять классы .NET Framework, а также создавать универсальные классы и методы;
- Читать и записывать данные с помощью методов файлового ввода/вывода, потоков, сериализовать и десериализовать данные в различных форматах;
- Создавать и использовать модель данных (Entity Data Model) для доступа к базам данных и с помощью LINQ-запросов, обновлять данные;
- Создавать графический интерфейс пользователя средствами XAML;
- Повышать удобство и снижать время отклика приложений с помощью задач и асинхронных операций;
- Интегрировать неуправляемые библиотеки (unmanaged libraries) и динамические компоненты в приложения C#;
- Использовать метаданные с помощью отражения (reflection), создавать и использовать пользовательские атрибуты, генерировать код во время выполнения, а также управлять версиями сборок;
- Шифровать и расшифровывать данные с помощью симметричного и асимметричного шифрования.

Детальная информация о курсе

Модуль 1. Обзор синтаксиса C#

- Обзор способов разработки приложений с использованием C#
- Типы данных, операторы и выражения

- Программирование языковых конструкций на C#

Лабораторная работа: Разработка приложения Class Enrolment

- Реализация функциональности редактирования списка студентов
- Реализация функциональности добавления в список студентов
- Реализация функциональности удаления из списка студентов
- Отображение возраста студента

По окончании этого модуля слушатели смогут:

- Описать архитектуру .NET Framework приложения и использовать функции Visual Studio 2012 и C# для поддержки разработки в .NET Framework
- Использовать основные типы данных, операторы и выражения C#
- Использовать стандартные конструкции C#

Модуль 2. Создание методов, обработка исключений и мониторинг приложений

- Создание и вызов методов
- Создание перегруженных методов и использование опциональных (optional) и параметров, возвращающих значение (Output)
- Обработка исключений
- Мониторинг приложений

Лабораторная работа: Расширение функциональности приложения Class Enrolment

- Рефакторинг существующего кода
- Проверка корректности (validation) информации о студенте
- Сохранение изменений списка классов

По окончании этого модуля слушатели смогут:

- Создавать и вызывать методы, передавать параметры в методы, возвращать значения из методов
- Создавать перегруженные методы, а также использовать опциональные (optional), а также возвращающие значение (Output) параметры
- Перехватывать и обрабатывать исключения, записывать информацию в журнал событий
- Объяснить требования для регистрации, трассировки и профилирования в масштабируемых приложениях

Модуль 3. Разработка кода для графического приложения

- Реализация структур и перечислений
- Организация данных в коллекции
- Обработка событий

Лабораторная работа: Написание кода для приложения Grades Prototype

- Добавление логики навигации
- Создание типов данных для хранения информации о пользователях и классах
- Отображение информации о пользователях и классах

По окончании этого модуля слушатели смогут:

- Определять и использовать структуры и перечисления
- Создавать и использовать простые коллекции для хранения данных в памяти
- Создавать, подписывать и генерировать события

Модуль 4. Создание классов и реализация коллекций

- Создание классов
- Определение и реализация интерфейсов
- Реализация коллекции

Лабораторная работа: Добавление проверки данных и коллекции в приложение Grades

- Реализация Teacher, Student и Grade Types как классов
- Добавление проверки данных в класс Grades
- Вывод отсортированного списка студентов

- Добавление информации об учителях

По окончании этого модуля слушатели смогут:

- Создавать и использовать классы
- Определять и реализовывать интерфейсы
- Создавать универсальные (generic) коллекций

Модуль 5. Создание иерархии классов при помощи наследования

- Создание иерархии классов
- Расширение классов .NET Framework
- Создание универсальных (generic) типов

Лабораторная работа: Рефакторинг основных функций класса User

- Создание и наследование от базового класса пользователя (User)
- Реализация сложного пароля с помощью абстрактного метода
- Создание собственного исключения ClassFullException

По окончании этого модуля слушатели смогут:

- Определить абстрактные классы и наследовать от базового класса для создания иерархии классов
- Наследовать классы .NET Framework и использовать методы расширения для добавления пользовательских функций к унаследованным классам
- Создавать универсальные классы и методы

Модуль 6. Чтение и запись локальных данных

- Чтение и запись файлов
- Сериализации и десериализации данных
- Реализация ввода/вывода с помощью потоков

Лабораторная работа: Создание отчета Grades

- Сериализация данных для XML отчета Grades
- Предварительный просмотр отчета Grades
- Сохранение сериализованных данных Grades в файл

По окончании этого модуля слушатели смогут:

- Читать и записывать данные из файловой системы с помощью файлового ввода/вывода
- Преобразовать данные в формат, доступный для записи или чтения из файла или другого источника данных
- Использовать потоки для передачи и приема данных из файла или другого источника данных

Модуль 7. Доступ к базе данных

- Создание и использование моделей данных (Entity Data Models)
- Запрос данных с помощью LINQ
- Обновление данных с помощью LINQ

Лабораторная работа: Получение и изменение данных Grades

- Создание модели данных (Entity Model) для базы данных
- Обновление данных Student и Grade Data при помощи Entity Framework
- Настройка модели для проверки корректности данных

По окончании этого модуля слушатели смогут:

- Создавать модели данных, описать основные классы, содержащиеся в модели, а также настраивать сгенерированный код
- Использовать LINQ-запросы и работать с данными
- Использовать LINQ для вставки, обновления и удаления данных

Модуль 8. Доступ к удаленным данным

- Доступ к данным через сеть

- Доступ к данным в облаке

Лабораторная работа: Получение и изменение данных Grades в облаке

- Создание службы WCF Data Services для баз данных SchoolGrades
- Интеграция службы WCF Data Services в приложение
- Получение студенческой фотографии через интернет

По окончании этого модуля слушатели смогут:

- Использовать классы в пространстве имен System.Net для передачи и приема данных через сеть
- Создавать и использовать WCF Data Services для доступа к данным в облаке

Модуль 9. Проектирование пользовательского интерфейса для графических приложений

- Использование XAML для проектирования пользовательского интерфейса
- Привязка элементов управления к данным
- Стилизация пользовательского интерфейса

Лабораторная работа: Настройка студенческой фотографии и стилизация приложений

- Настройка внешнего вида студенческих фотографий
- Стилизация формы входа в приложение и элемента управления StudentPhoto

По окончании этого модуля слушатели смогут:

- Определить виды и элементы XAML для дизайна графического интерфейса пользователя
- Использовать методы привязки данных для соединения XAML элементов с источниками данных и видами
- Добавлять дизайн и динамические преобразования в пользовательский интерфейс XAML

Модуль 10. Повышение производительности приложений

- Реализация многопоточности с помощью классов Tasks и лямбда выражений
- Асинхронное выполнение операций
- Синхронизация одновременного доступа к данным

Лабораторная работа: Повышение отзывчивости и производительности приложений

- Обеспечение отзывчивости пользовательского интерфейса во время доступа к объекту Teachers
- Предоставление визуальной обратной связи во время длительных операций

По окончании этого модуля слушатели смогут:

- Реализовать многозадачность с помощью классов Tasks и лямбда выражений
- Асинхронно выполнять операции
- Координировать одновременный доступ к общим данным для нескольких задач с помощью синхронных примитивов и сопутствующих коллекций

Модуль 11. Интеграция с неуправляемым (unmanaged) кодом

- Создание и использование динамических объектов
- Управление временем жизни объектов и unmanaged ресурсами

Лабораторная работа: Обновление отчета

- Создание отчета Grades с помощью Microsoft Office Word
- Управление временем жизни объектов при помощи метода Dispose

По окончании этого модуля слушатели смогут:

- Интегрировать наследуемый код в приложение C # с помощью DLR
- Контролировать время жизни unmanaged ресурсов и гарантировать их правильное освобождение

Модуль 12. Создание повторно используемых типов и сборок

- Проверка метаданных объекта
- Создание и использование пользовательских атрибутов

- Генерация управляемого кода
- Управление версиями, подпись и развертывание сборок

Лабораторная работа: Определение данных для включения в отчет Grades

- Создание собственного атрибута IncludeInReport
- Генерация отчетов
- Централизованное хранение сборки Grades.Utilities

По окончании этого модуля слушатели смогут:

- Проверять метаданные объектов во время выполнения с помощью рефлексии
- Создавать и использовать пользовательские атрибуты класса
- Создавать управляемый код во время выполнения с помощью CodeDOM
- Управлять различными версиями и размещать сборки в глобальном кэше (GAC)

Модуль 13. Шифрование данных

- Реализация симметричного шифрования
- Реализация асимметричного шифрования

Лабораторная работа: Шифрование отчета Grades

- Шифрование отчета Grades
- Расшифровка отчета Grades

По окончании этого модуля слушатели смогут:

- Выполнять симметричное шифрование с помощью классов System.Security
- Выполнять асимметричное шифрование с помощью классов System.Security

Предварительные требования

Для успешного прохождения курса необходимо обладать следующими знаниями и навыками:

- Практический опыт работы с Windows Server 2008, Windows Server 2008 R2, Windows Server 2012
- Хорошие знания в области Active Directory и сетевой инфраструктуры
- Знания в объеме курса «20410: Установка и настройка Windows Server 2012»

Дополнительная информация

Если у вас возникли вопросы, воспользуйтесь следующими ссылками:

- Информации об [учебных курсах и программах сертификации](#)
- [Расписание курсов](#)