

**ЧАСТНОЕ УЧРЕЖДЕНИЕ
ОРГАНИЗАЦИЯ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ «УЧЕБНЫЙ ЦЕНТР «ЗВЕЗДЫ И С»**

УТВЕРЖДАЮ

Директор

ЧУ ОДПО «Учебный центр «Звезды и С»

Стародубцев В.Н.

«19» мая 2021г.



Дополнительная профессиональная образовательная программа
повышения квалификации
М20483 «Программирование на C#»

Москва, 2021г.

1. Целевая установка

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

Разработчики, посещающие этот курс, уже должны были иметь некоторый ограниченный опыт использования C# для выполнения основных задач программирования. В частности, студенты должны иметь практический опыт использования C#, который демонстрирует их понимание следующего:

- Как назвать, объявить, инициализировать и присвоить значения переменным в приложении.
- Как использовать: арифметические операторы для выполнения арифметических вычислений с участием одной или нескольких переменных; реляционные операторы для проверки отношений между двумя переменными или выражениями; логические операторы для объединения выражений, содержащих реляционные операторы.
- Как создать синтаксис кода для простых операторов программирования с использованием ключевых слов языка C# и распознать синтаксические ошибки с помощью Visual Studio IDE.
- Как создать простую ветвящуюся структуру, используя оператор IF.
- Как создать простую циклическую структуру, используя оператор For для итерации массива данных. Как использовать Visual Studio IDE для поиска простых логических ошибок.
- Как создать функцию, которая принимает аргументы (параметры) и возвращает значение указанного типа.
- Как спроектировать и построить простой пользовательский интерфейс, используя стандартные элементы управления из набора инструментов Visual Studio.
- Как подключиться к базе данных SQL Server. Основы получения и хранения данных.
- Как отсортировать данные в цикле.
- Как распознать классы и методы, используемые в программе.

2. Планируемые результаты обучения

Реализация Программы направлена на повышение профессионального уровня в рамках имеющейся квалификации, определяемой профессиональным стандартом «06.001 Программист», утвержденным Приказом Минтруда

России от 18.11.2013 N 679н"Об утверждении профессионального стандарта "Программист".

Совершенствуемые компетенции

Разработка и отладка программного кода

№	Компетенция	Код компетенции
1	Формализация и алгоритмизация поставленных задач	A/01.3
2	Написание программного кода с использованием языков программирования, определения и манипулирования данными	A/02.3
3	Оформление программного кода в соответствии с установленными требованиями	A/03.3
4	Работа с системой контроля версий	A/04.3
5	Проверка и отладка программного кода	A/05.3

Приобретенные навыки

- Дать описание основного синтаксиса и функций Visual C
- Создание методов, обработка исключений и описание требований к мониторингу крупномасштабных приложений.
- Реализация базовой структуры и основных элементов типичного настольного приложения.
- Создание классов, определение и внедрение интерфейсов, а также создание и использование общих коллекций.
- Использование наследование, чтобы создать иерархию классов и расширить класс .NET Framework.
- Чтение и запись данных с использованием файлов ввода/вывода и потоков, а также сериализация и десериализация данных в различных форматах.
- Создание и использование модели данных объекта для доступа к базе данных и использование LINQ для запроса данных.
- Получение доступа к удаленным данным и их запрос с использованием типов в пространстве имен System.Net и службах данных WCF.

- Создание графического интерфейса пользователя с помощью XAML.
- Улучшение пропускной способности и времени отклика приложений, используя задачи и асинхронные операции.
- Интеграция неуправляемых библиотек и динамических компонентов в приложение Visual C
- Изучение метаданные типов с помощью отражения, создание и использование пользовательских атрибутов, генерация кода во время выполнения и управление версиями сборок.
- Шифрование и дешифрование данных с использованием симметричного и асимметричного шифрования.

3. Учебный план.

№ п/п	Наименование модулей по программе	В том числе аудиторных			Форма контроля
		Всего	<i>Лекций</i>	<i>Практических занятий</i>	
1	Обзор синтаксиса C#	2	1	1	Прак. занятие
2	Создание методов, обработка исключений и мониторинг приложений	4	2	2	Прак. занятие
3	Основные типы и конструкции языка C#	3	1	2	Прак. занятие
4	Создание классов и реализация коллекций	4	2	2	Прак. занятие
5	Создание иерархии классов при помощи наследования	4	2	2	Прак. занятие
6	Чтение и запись локальных данных	4	2	2	Прак. занятие
7	Доступ к базе данных	2	1	1	Прак. занятие

8	Доступ к удаленным данным	4	2	2	Прак. занятие
9	Проектирование пользовательского интерфейса для графических приложений	4	2	2	Прак. занятие
10	Повышение производительности и отклика приложений	2	1	1	Прак. занятие
11	Интеграция с неуправляемым (unmanaged) кодом	2	1	1	Прак. занятие
12	Создание повторно используемых типов и сборок	3	2	1	Прак. занятие
13	Шифрование данных	2	1	1	Прак. занятие
14	Итоговая аттестация: (Лабораторная работа)	2	-	2	Прак. занятие
	Итого:	40	20	20	

4. Календарный учебный график

Календарный учебный график составляется в форме расписания занятий при наборе группы и прилагается к программе повышения квалификации.

Форма обучения: очная, очная с применением дистанционных технологий.

Трудоемкость программы: 40 часа.

Сроки освоения программы: 5 дней.

Режим занятий: дневной, вечерний.

5. Рабочие программы дисциплин

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

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

Модуль 3. Основные типы и конструкции языка C#

- 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

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

Модуль 5. Создание иерархии классов при помощи наследования

- Создание иерархии классов
- Расширение классов .NET Framework
- **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

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

Модуль 7. Доступ к базе данных

- Создание и использование моделей данных (Entity Data Models)
- Запрос 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

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

Модуль 9. Проектирование пользовательского интерфейса для графических приложений

- Использование XAML для проектирования пользовательского интерфейса
- Привязка элементов управления к данным
- **Лабораторная работа:** Настройка студенческой фотографии и стилизация приложений
- Настройка внешнего вида студенческих фотографий
- Стилизация формы входа в приложение
- Добавление анимации к элементу управления StudentPhoto

Модуль 10. Повышение производительности и отклика приложений

- Реализация многопоточности
- 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

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

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

Модуль 13. Шифрование данных

- Реализация симметричного шифрования
- Реализация асимметричного шифрования
- **Лабораторная работа:** Шифрование отчета Grades
- Шифрование отчета Grades

Расшифровка отчета Grades

6. Организационно-педагогические условия реализации программы

6.1. Материально-технические условия реализации программы

Исполнитель обеспечивает для проведения обучения следующие средства вычислительной техники:

- персональный компьютер для преподавателя – 1 шт.;
- персональный компьютер для каждого Слушателя;
- проектор и экран – 1 комплект;
- доска – 1 шт.

Персональные компьютеры объединены в локальную вычислительную сеть. Технические характеристики персональных компьютеров:

- процессор 4 ядра 3,1 ГГц;
- оперативная память - 32 Гб;
- SSD + 2 HDD в RAID0 не менее 500Гб;

- два монитора (24' + 22' FullHD);
- комплект клавиатура и мышь.

6.2. Учебно-методическое обеспечение программы

Каждый Слушатель обеспечивается авторизованным учебным пособием на английском языке.

7. Требования к профессорско-преподавательскому составу

Высшее профессиональное образование и стаж работы в образовательном учреждении не менее 1 года. Статус Microsoft Certified Trainer.

8. Форма аттестации

Текущий контроль успеваемости и качества подготовки, промежуточная и итоговая аттестации слушателей осуществляются в процессе изучения, освоения данной профессиональной образовательной программы повышения квалификации.

Текущий контроль успеваемости и качества подготовки осуществляется в пределах времени, отведенного на учебные занятия, и выполняет одновременно обучающую функцию. Текущий контроль успеваемости проводится в процессе изучения каждого раздела (темы, подтемы) внутри модуля данной дополнительной профессиональной программы и проводится в форме устного опроса преподавателя. Промежуточная и итоговая аттестации проводятся в форме лабораторных работ на персональном компьютере слушателя, который использовался во время обучения, в классе под наблюдением преподавателя. По окончании каждого модуля рабочей программы проводится промежуточная аттестация в виде промежуточной лабораторной работы по теме каждого модуля данной профессиональной образовательной программы.

Итоговая аттестация проводится в форме итоговой лабораторной работы. В итоговой лабораторной работе задействуются материалы из всех модулей пройденной программы.

Аттестация считается пройденной в случае успешного завершения лабораторной работы, а именно: выполнения поставленной задачи: «Создание графического модуля программы статистики для предвыборной компании».

Время выполнения итоговой аттестации – 2 ак. часа.

9. Оценочные материалы к итоговой аттестации

Итоговая аттестация проводится в форме выполнения задания. Результаты итоговой аттестации слушателей выставляются по двух бальной шкале («зачтено\не зачтено»). Итоговая аттестация считается пройденной

(«зачтено»), если слушатель выполнил все лабораторные работы и итоговое задание.

Пример решения задач (Официальное учебное пособие Microsoft, язык – английский):

Customizing Student Photographs and Styling the Application

Scenario

Now that you and The School of Fine Arts are happy with the basic functionality of the application, you

need to improve the appearance of the interface to give the user a nicer experience through the use of

animations and a consistent look and feel.

You decide to create a **StudentPhoto** control that will enable you to display photographs of students in

the student list and other views. You also decide to create a fluid method for a teacher to remove a

student from their class. Finally, you want to update the look of the various views, keeping their look

consistent across the application.

Objectives

After completing this lab, you will be able to:

- Create and use user controls.
- Use styles and animations.

Lab Setup

Estimated Time: **90 minutes**

You will find the high-level steps on the following page:

<https://github.com/MicrosoftLearning/20483->

[Programming-in-C-](https://github.com/MicrosoftLearning/20483-)

[Sharp/blob/master/Instructions/20483C_MOD09_LAB_MANUAL.md](https://github.com/MicrosoftLearning/20483-).

You will find the detailed steps on the following page:

<https://github.com/MicrosoftLearning/20483->

[Programming-in-C-Sharp/blob/master/Instructions/20483C_MOD09_LAK.md](https://github.com/MicrosoftLearning/20483-).

Exercise 1: Customizing the Appearance of Student Photographs

Scenario

In this exercise, you will customize the appearance of student photographs in the production application.

You will begin by creating a **StudentPhoto** user control that will host the photographs on the various

pages in the UI. Then you will lay out the user controls and write code to raise the **Student_Click** method

when a user clicks a student photograph.

Next, you will add a remove button with a red X to the user control that users can click to remove a

student from a class. When a user hovers over the button, the opacity of the button and the photograph

will change.

Finally, you will run the application to verify that the student's image is displayed correctly on the

StudentsPage view.

Exercise 2: Styling the Logon View

Scenario

In this exercise, you will update the **LogonPage** control to have the same look and feel as the rest of the

application.

First, you will define styles for the username and password text boxes on the **LogonPage** of the

application. You will use the **Style** property of each control to apply the styles that you have defined.

Then you will define some global styles for use across the entire application. You will define a style for

labels and a style for text. Finally, you will run the application to verify that the styling of the text elements

has changed throughout the application.

MCT USE ONLY. STUDENT USE PROHIBITED

9-26 Designing the User Interface for a Graphical Application

Exercise 3: Animating the StudentPhoto Control (If Time Permits)

Scenario

In this exercise, you will update the **StudentPhoto** control to animate when a user hovers over it.

First you will define an animation for the **StudentPhoto** control, which will cause a student's photograph

to pulse when a user hovers over it. You will then add event handlers for this animation and apply the

animation to the control. Finally, you will run the application to verify that the animation executes

correctly.