

Белорусский государственный университет



**РАЗРАБОТКА ПРОГРАММНЫХ ПРИЛОЖЕНИЙ НА С#**

**Учебная программа учреждения высшего образования  
по учебной дисциплине для специальности:**

1-25 01 12 Экономическая информатика

2020 г.

2021/6/24 10:33

Учебная программа составлена на основе ОСВО 1-25 01 12-2013, учебная программа рег. № Е25-289/уч. от 16.03.2020

**СОСТАВИТЕЛИ:**

А.В. Канаш – старший преподаватель кафедры цифровой экономики экономического факультета БГУ

**РЕЦЕНЗЕНТЫ:**

Л.М. Канаш, главный специалист отдела проектирования и внедрения новых информационных технологий ЗАО «ОРГСТРОЙ»

**РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:**

Кафедрой цифровой экономики

(протокол № 10 от 22.05.2020г.);

Научно-методическим Советом БГУ

(протокол № 5 от 17.06.2020 г.)

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### Цели и задачи учебной дисциплины

**Цель** учебной дисциплины – обеспечение студентов теоретическими знаниями основ и инструментария для разработки программных приложений на языке программирования С#; формирование устойчивого алгоритмического мышления; получение студентами практических навыков разработки программных приложений в среде программирования Visual Studio (SharpDevelop).

### Задачи учебной дисциплины:

Формирование:

1. навыков разработки ПО с использованием линейных и нелинейных структур данных;
2. понятия потока, безопасного многопоточного приложения;
3. умения и навыков организации файлового ввода-вывода на языке С#;
4. системы знаний и умений разработки приложений прикладного характера с использованием технологии ADO.NET, используя связный и автономный уровни доступа к серверной базе данных.

**Место учебной дисциплины** в системе подготовки специалиста с высшим образованием (магистра).

Учебная дисциплина относится к циклу социально-гуманитарных дисциплин (компонент учреждения образования)

**Связи** с другими учебными дисциплинами, включая учебные дисциплины компонента учреждения высшего образования, дисциплины специализации и др.

В рамках учебной дисциплины «Разработка программных приложений на С#» расширяются, углубляются знания и практические навыки, полученные при изучении учебных дисциплин «Компьютерные информационные технологии», «Web-программирование», «Алгоритмы и структуры данных»

### Требования к компетенциям

Освоение учебной дисциплины «Разработка программных приложений на С#» должно обеспечить формирование следующих академических, социально-личностных и профессиональных компетенций:

*академические* компетенции:

АК-1. Уметь применять базовые научно-теоретические знания для решения теоретических и практических задач.

АК-2. Владеть системным и сравнительным анализом.

АК-3. Владеть исследовательскими навыками.

АК-4. Уметь работать самостоятельно.

АК-5. Быть способным порождать новые идеи (обладать креативностью)

АК-6. Владеть междисциплинарным подходом при решении проблем

АК-7. Иметь навыки, связанные с использованием технических устройств, управлением информацией и работой с компьютером.

АК-9. Уметь учиться, повышать свою квалификацию в течение всей жизни.

**социально-личностные** компетенции:

СЛК-2. Быть способным к социальному взаимодействию.

СЛК-3. Обладать способностью к межличностным коммуникациям.

СЛК-4. Владеть навыками здоровьесбережения.

СЛК-5. Быть способным к критике и самокритике.

СЛК-6. Уметь работать в команде.

**профессиональные** компетенции:

ПК-8. Владеть современными средствами телекоммуникаций, организовывать и вести электронный бизнес.

ПК-13. Оценивать эффективность решений в сфере информатизации.

ПК-14. Использовать информационные технологии для повышения эффективности обработки исходных данных, проведения математических и статистических расчётов, ведения документооборота и маркетинговых исследований.

ПК-26. Осуществлять проектирование, тестирование, сопровождение и эксплуатацию информационных систем, разрабатывать техническую документацию к программному обеспечению и требования к внедрению тиражируемых информационных систем.

ПК-27. Проводить научные исследования в области использования информационных технологий в экономике.

ПК-28. Проводить научные исследования с целью совершенствования методов проектирования, тестирования, оценки качества, внедрения и сопровождения прикладного программного обеспечения.

В результате освоения учебной дисциплины студент должен:

**знать:** логические и арифметические основы обработки информации в ПЭВМ; основные типы данных и их внутреннее представление в языке С#; синтаксис и все конструкции языка программирования С# – операторы, операции, управляющие структуры, файлы; способы вызова функций и передачи параметров; организацию разработки программного продукта на языке С#, функциональные возможности и специфику разработки приложений в среде визуального программирования Visual Studio (Sharp Developer); основные операции с файлами в языке С#; основы многопоточного программирования; сущность технологии доступа к данным ADO.NET, ее

объектную модель; принципы построения приложений с использованием данных серверной СУБД;

**уметь:** разрабатывать на языке С# программы для решения несложных алгоритмических задач; решать задачи хранения информации в виде файлов разных типов на языке С#; разрабатывать безопасные многопоточные приложения; разрабатывать приложения, в которых реализован доступ к базе данных с помощью технологии ADO.NET.

**владеть:** навыками подготовки и отладки программ средствами визуального программирования Visual Studio (SharpDevelop); процедурного программирования, которые основаны на применении и вычислении функций; методами обработки файлов разных типов на языке С#; технологией доступа ADO.NET к серверной СУБД; приемами разработки современных Windows-приложений прикладного характера, использующих данные серверной СУБД и их тестирования.

### **Структура учебной дисциплины**

Дисциплина изучается в 5 семестре. Всего на изучение учебной дисциплины «Разработка программных приложений на С#» отведено:

для очной формы получения высшего образования – 90 часов, в том числе 50 аудиторных часов, из них: лекции – 20 часов, лабораторные занятия – 26 часов, управляемая самостоятельная работа – 4 часа.

Трудоемкость учебной дисциплины составляет 2 зачетные единицы.

Форма текущей аттестации – *зачет*.

# СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

## Раздел 1. Базовый синтаксис языка C#

### Тема 1.1. Общая характеристика и состав языка C#. Типы данных

История возникновения языка программирования C#. Понятие компонента. Платформа .NET. Этапы создания приложения на C# в среде .NET. Пространство имен. Алфавит и лексемы. Идентификаторы. Ключевые слова. Знаки операций и разделители. Литералы (константы). Управляющая последовательность. Комментарии. Типы данных. Встроенные типы. Типы литералов. Типы-значения и ссылочные типы. Упаковка и распаковка.

### Тема 1.2. Конструкции языка C#

Переменные. Именованные константы. Выражения. Преобразования встроенных арифметических типов-значений. Исключения. Основные операции C#: инкремент и декремент, new, операции отрицания, явное преобразование типа, умножение, деление, остаток от деления, сложение и вычитание, операции сдвига, поразрядные логические операции, условные логические операции, операции присваивания. Консольный ввод-вывод. Выражения, блоки и пустые операторы. Базовые конструкции структурного программирования. Операторы ветвления. Оператор выбора switch. Операторы цикла и передачи управления: цикл с предусловием, цикл с постусловием, цикл с параметром for, цикл перебора foreach, операторы goto, break, return. Обработка исключительных ситуаций: операторы try и throw.

## Раздел 2. Классы

### Тема 2.1. Классы: основные понятия

Описание классов. Состав класса. Присваивание объектов. Данные: поля и константы. Методы. Параметры метода. Вызов метода. Ключевое слово this. Конструкторы. Свойства.

### Тема 2.2. Массивы, символы и строки

Тип данных Массив в C#. Одномерные массивы. Прямоугольные массивы. Ступенчатые массивы. Класс System.Array. Оператор foreach для работы с массивом. Массивы объектов. Символы. Массивы символов. Строки типа string. Класс System.String. Форматирование строк. Строки типа StringBuilder

### Тема 2.3. Классы: подробности

Перегрузка методов. Рекурсивные методы. Методы с переменным количеством аргументов. Метод Main. Индексаторы. Операции класса. Унарные и бинарные операции в классах. Операции преобразования типа. Деструкторы. Вложенные типы. Наследование классов. Описание класса-потомка. Раннее и позднее связывания. Виртуальные методы. Абстрактные

классы. Бесплодные классы. Взаимоотношения между классами. UML. Класс System.Object.

### **Раздел 3. Создание графических приложений C# в Windows Forms**

#### **Тема 3.1. Введение в Windows Forms**

Технологии создания графических интерфейсов. Создание графического приложения. Запуск приложения. Основы форм. Структура формы. Основные свойства форм. Программная настройка свойств. Установка размеров формы. Начальное расположение формы. Фон и цвета формы. Добавление форм. Взаимодействие между формами. События в Windows Forms. События формы.

#### **Тема 3.2. Контейнеры в Windows Forms**

Понятие контейнера. Динамическое добавление элементов. Элементы GroupBox, Panel, FlowLayoutPanel, TableLayoutPanel. Размеры элементов и их позиционирование в контейнере: позиционирование, установка размеров, свойства Anchor и Dock. Панели вкладок TabControl и SplitContainer. Управление вкладками в коде. Элементы управления.

### **Раздел 4. Программирование ввода/вывода. Многопоточное программирование**

#### **Тема 4.1. Организация C#-системы ввода-вывода: байтовые и символьные потоки. Встроенные потоки**

Понятие потока. Стандартные потоки Console.In, Console.Out, Console.Error. Классы потоков. Класс Stream. Классы байтовых потоков. Классы символьных потоков. Классы двоичных потоков. Класс FileStream и файловый ввод-вывод на побайтовой основе. Открытие и закрытие файла. Запись данных в файл. Считывание байтов из объекта класса FileStream. Файловый ввод-вывод с ориентацией на символы. Использование классов StreamWriter и StreamReader. Считывание и запись двоичных данных. Классы BinaryWriter и BinaryReader. Навигация по файловой системе: классы для работы с файловой системой, получение сведений о файле, копирование файлов, перечисление файлов в папке, изменение расширения файлов.

#### **Тема 4.2. Многопоточное программирование в языке C#**

Многопоточность программы. Создание потока в C#. Класс Thread. Запуск и остановка потока в C#. Приоритеты потоков в C#. Изменение типов потоков в C#. Синхронизация потоков. Управление временем существования потоков.

## **Раздел 5. Управление данными с помощью технологии ADO.NET (ActiveX Data Objects)**

### **Тема 5.1. Базовые типы ADO.NET. Работа с БД Microsoft SQL Server без разрыва соединения (связный уровень ADO.NET). Использование отсоединенных (автономных) наборов данных**

Краткий обзор ADO.NET. Пространство имен System.Data. Поставщики данных ADO.NET. Поведение объектов подключения. Объектная модель ADO.NET. Объекты ADO .NET: таблицы, поля, строки, отношения, DataAdapter, DataView. Интерфейсы соединения, транзакции, команды, адаптеров данных, чтения данных. Создание соединения с источником данных. Командная строка соединения ConnectionString. Управление соединением. Объект Connection. Обработка исключений при работе с MS SQL. Работа с базой данных с помощью SQL-запросов. Объект Command: создание и инициализация. Командная строка SQL-запроса CommandText. Выборка, добавление, удаление и обновление данных. Вывод информации с использованием объекта чтения данных. Объекты параметризованных команд. Хранимые процедуры. Отображение информации из базы данных с помощью элементов управления ListBox, ComboBox, ListView, DataGridView, DataTable. Использование отсоединенных (автономных) наборов данных. Создание подключения к базе данных. Управляемые провайдеры. Обновление базы данных с помощью адаптера данных. Динамическое получение объектов команд во время работы с помощью построителя команд SqlCommandBuilder.

## УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Дневная форма получения образования

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
<b>1</b>	<b>Базовый синтаксис языка C#</b>	<b>4</b>			<b>6</b>			
1.1	Общая характеристика и состав языка C#. Типы данных	2			2			опрос, защита лабораторных работ
1.2	Конструкции языка C#	2			4			
<b>2</b>	<b>Классы</b>	<b>6</b>			<b>6</b>			
2.1	Классы: основные понятия	2			2			опрос, защита лабораторных работ
2.2	Массивы, символы и строки	2			2			
2.3	Классы: подробности	2			2			
<b>3</b>	<b>Создание графических приложений C# в Windows Forms</b>	<b>4</b>					<b>2</b>	
3.1	Введение в Windows Forms	2						опрос, защита лабораторных работ, контрольная работа
3.2	Контейнеры в Windows Forms	2					2	
<b>4</b>	<b>Программирование ввода/вывода. Многопоточное программирование</b>	<b>4</b>			<b>4</b>			
4.1	Организация C#-системы ввода-вывода: байтовые и символьные потоки. Встроенные потоки	2			2			опрос, защита лабораторных работ

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
4.2	Многопоточное программирование в языке С#	2			2			опрос, защита лабораторных работ
<b>5</b>	<b>Управление данными с помощью технологии ADO.NET (ActiveX Data Objects)</b>	<b>2</b>			<b>10</b>		<b>2</b>	
5.1	Базовые типы ADO.NET. Работа с БД Microsoft SQL Server без разрыва соединения (связный уровень ADO.NET).	2			6			защита лабораторных работ
5.2	Использование отсоединенных (автономных) наборов данных				4		2	защита лабораторных работ, защита проекта
	<b>Итого</b>	<b>20</b>			<b>26</b>		<b>4</b>	

## **ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ**

### **Перечень основной литературы**

1. Hello World – интерактивный учебник по основам С# [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/tutorials/intro-to-csharp/hello-world>. – Дата доступа: 01.10.2020
2. Албахри, Д. С# 8.0. Карманный справочник / Джозеф Албахри, Бен Албахри – СПб.: ООО «Диалектика», 2020. – 240 с.
3. Стивен, К. Конкурентность в С#. Асинхронное, параллельное и многопоточное программирование / К. Стивен – СПб.: Питер, 2020. – 272 с.
4. Мюллер, Д. С# для чайников / Джон Поль Мюллер, Билл Семпф, Чак Сфер – СПб.: ООО «Диалектика», 2019. – 608 с.
5. Фленов, М.Е. Библия С#. – 4-е изд., перераб. и доп. / М.Е. Фленов – СПб.: БХВ-Петербург, 2019. – 512 с.

### **Перечень дополнительной литературы**

1. Бельков, С.А. Прикладное программирование с использованием языка С-Шарп: учебно-методическое пособие / С. А. Бельков. – Екатеринбург: Изд-во Урал. ун-та, 2017. – 120 с.
2. Евсева, О.Н. Работа с базами данных на языке С#. Технология ADO.NET: учебное пособие / О.Н. Евсева, А.Б. Шамшев. – Ульяновск: УлГТУ, 2009. – 170 с.

## **Перечень рекомендуемых средств диагностики и методика формирования итоговой оценки**

Оценка за письменные и устные ответы на лекциях (опрос) включает в себя корректность и полноту ответа, обоснованность аргументов, наличие примеров из практики.

Оценка за выполнение лабораторных работ формируется на основе следующих критериев: корректность полученных результатов и их интерпретацию, умение воспроизвести выполнение заданий, защиты выполненных индивидуальных заданий, полнота ответов на вопросы.

Итоговая оценка за лабораторные работы рассчитывается путем усреднения оценок за запланированные к выполнению лабораторные работы.

Контрольные тесты оцениваются исходя из доли правильно выполненных заданий по методике, представленной в таблице:

Методика выставления оценки по тесту										
Доля	[0-15)	[15-25)	[25-35)	[35-45)	[45-55)	[55-65)	[65-75)	[75-85)	[85-95)	[95-100]
Оценка	1	2	3	4	5	6	7	8	9	10

Контрольные работы и проект оцениваются исходя из полноты выполнения, корректности полученных результатов, качества исполнения, проявления креативности.

Итоговая оценка за контрольные работы (тесты) рассчитывается путем усреднения оценок по всем контрольным работам (тестам).

Формой текущей аттестации по дисциплине «Разработка программных приложений на C#» учебным планом предусмотрен зачет.

При формировании итоговой оценки используется рейтинговая оценка знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая оценка предусматривает использование весовых коэффициентов для текущего контроля знаний и текущей аттестации студентов по дисциплине.

Примерные весовые коэффициенты, определяющие вклад текущего контроля знаний и текущей аттестации в рейтинговую оценку:

- лабораторные работы – 50 %;
- контрольные тесты – 20 %;
- контрольные работы и проект – 25 %;
- опросы – 5 %.

Итоговая оценка по дисциплине рассчитывается на основе оценки текущей успеваемости (рейтинговой оценки) и экзаменационной оценки с учетом их весовых коэффициентов и использования правил математического округления.

Вклад текущей успеваемости в итоговую оценку составляет 50%, экзаменационной оценки – 50 %.

### **Примерный перечень заданий для управляемой самостоятельной работы студентов**

Управляемая самостоятельная работа (консультационно-методическая поддержка и контроль) осуществляться преимущественно в дистанционной форме и обеспечивается средствами образовательного портала БГУ LMS Moodle

В отдельных случаях управляемая самостоятельная работа проводится в форме аудиторных занятий, согласно утвержденному графику.

Объем часов на составление и размещение заданий, консультации и контроль, осуществляемые с использованием технологий дистанционного обучения, планируется в пределах учебных часов, отведенных на УСР.

Приоритетным направлением для разработки УСР в дистанционной форме являются открытые задания как основной содержательный элемент эвристического обучения.

### **Примерные виды заданий для УСР**

#### **Раздел 3. Создание графических приложений C# в Windows Forms (2ч.)**

Создать графическое приложение Windows Forms (Net. Framework) C#, состоящее не менее, чем из 2-х форм с использованием компонентов MenuStrip, ToolStrip, Panel, Radiobutton, Combobox, Textbox, Button.

(Форма контроля – контрольная работа).

#### **Раздел 5. Управление данными с помощью технологии ADO.NET (ActiveX Data Objects)**

Разработать многооконный проект Windows Forms (Net. Framework) C# с использованием базы данных SQL Server и технологией доступа к данным ADO.NET (работа в группах до 4 человек)

(Форма контроля – защита проекта)

### **Примерная тематика лабораторных занятий.**

<b>№</b>	<b>Тема</b>
<b>Базовый синтаксис языка C#</b>	
1.	Изучение среды разработки Visual Studio (SharpDevelop). Линейные алгоритмы
2.	Разветвляющиеся алгоритмы
3.	Циклические алгоритмы
<b>Классы</b>	
4.	Классы и объекты. Строки

№	Тема
5.	Одномерные и многомерные массивы
6.	Методы (перегрузка, параметры по умолчанию, параметры по ссылке)
<b>Программирование ввода/вывода. Многопоточное программирование</b>	
7.	Файловый ввод/вывод в C#. Применение файлов при разработке Windows-приложений прикладного характера
8.	Многопоточное программирование в языке C#
<b>Управление данными с помощью технологии ADO.NET (ActiveX Data Objects)</b>	
9.	Создание реляционной базы данных и выполнение SQL-запросов к ней
10.	Работа с БД Microsoft SQL Server без разрыва соединения. Разработка консольных приложений баз данных, в которых работа с СУБД осуществляется «вручную»
11.	Работа с БД Microsoft SQL Server без разрыва соединения. Параметризованные запросы
12.	Работа с БД Microsoft SQL Server в автономном режиме. Создание объектов ADO.NET: DataSet, DataTable, DataRow, DataColumn, DataAdapter вручную
13.	Работа с БД Microsoft SQL Server в автономном режиме. Разработка простых Windows-приложений, в которых соединение и работа с БД осуществляется с помощью мастеров данных Visual Studio»
14.	«Работа с БД Microsoft SQL Server в автономном режиме. Использование объекта DataSet, состоящего из нескольких таблиц, а также объектов DataRelation для разработки Windows-приложений, содержащих бизнес-логику

## **Описание инновационных подходов и методов к преподаванию учебной дисциплины**

При организации образовательного процесса используются следующие подходы и методы:

**эвристический подход**, который предполагает: осуществление студентами лично-значимых открытий окружающего мира; демонстрацию многообразия решений большинства профессиональных задач и жизненных проблем; творческую самореализацию обучающихся в процессе создания образовательных продуктов; индивидуализацию обучения через возможность самостоятельно ставить цели, осуществлять рефлексию собственной образовательной деятельности.

**практико-ориентированный подход**, который предполагает: освоение содержания образования через решения практических задач; приобретение навыков эффективного выполнения разных видов профессиональной деятельности; ориентацию на генерирование идей, реализацию групповых студенческих проектов, развитие предпринимательской культуры; использованию процедур, способов оценивания, фиксирующих сформированность профессиональных компетенций.

**метод проектного обучения**, который предполагает: способ организации учебной деятельности студентов, развивающий актуальные для учебной и профессиональной деятельности навыки планирования, самоорганизации, сотрудничества и предполагающий создание собственного продукта; приобретение навыков для решения исследовательских, творческих, социальных, предпринимательских и коммуникационных задач.

**метод группового обучения**, который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями.

### **Методические рекомендации по организации самостоятельной работы обучающихся**

Для организации самостоятельной работы студентов по учебной дисциплине следует использовать современные информационные ресурсы: разместить на образовательном портале комплекс учебных и учебно-методических материалов (учебно-программные материалы, учебное издание для теоретического изучения дисциплины, методические указания к лабораторным занятиям, материалы текущего контроля и текущей аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образовательных стандартов высшего образования и учебно-программной документации, в т.ч. вопросы для подготовки к зачету, задания, тесты, вопросы для самоконтроля, список рекомендуемой литературы, информационных ресурсов).

Самостоятельная (практическая) работа студентов по изучению дисциплины «Разработка программных приложений на С#» выполняется в

форме аудиторной и внеаудиторной работы. Студентам предлагается самостоятельное изучение ряда вопросов, что предполагает углубленное изучение основной и дополнительной литературы. Эффективность самостоятельной работы студентов проверяется в ходе текущего и итогового контроля знаний. При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной (практической) работы, предоставленной в системе дистанционного обучения: поиск (подбор) и обзор литературы и электронных источников по индивидуально-заданной проблеме курса; работы, предусматривающие решение задач и выполнение упражнений, выдаваемых на практических занятиях; подготовка к контрольным работам; подготовка к зачету.

### **Примерный перечень вопросов к зачету**

1. История возникновения языка программирования C#.
2. Понятие компонента. Платформа .NET.
3. Этапы создания приложения на C# в среде .NET.
4. Пространство имен. Алфавит и лексемы.
5. Идентификаторы. Ключевые слова.
6. Знаки операций и разделители.
7. Литералы (константы).
8. Управляющая последовательность.
9. Комментарии.
10. Классификация типов данных.
11. Встроенные типы.
12. Типы литералов.
13. Типы-значения и ссылочные типы. Упаковка и распаковка.
14. Переменные. Именованные константы.
15. Выражения.
16. Преобразования встроенных арифметических типов-значений.
17. Исключения.
18. Основные операции C#: инкремент и декремент, операция new, операции отрицания, явное преобразование типа.
19. Основные операции C#: умножение, деление, остаток от деления.
20. Основные операции C#: сложение и вычитание.
21. Основные операции C#: операции сдвига, поразрядные логические операции, условные логические операции
22. Основные операции C#: операции присваивания.
23. Консольный ввод-вывод.
24. Выражения, блоки и пустые операторы.
25. Базовые конструкции структурного программирования.
26. Операторы ветвления.
27. Оператор выбора switch.
28. Цикл с предусловием, с постусловием, с параметром for, перебора foreach

29. Операторы передачи управления goto, break, return.
30. Обработка исключительных ситуаций: операторы try и throw.
31. Описание классов.
32. Состав класса. Присваивание объектов.
33. Данные: поля и константы.
34. Методы. Параметры метода. Вызов метода. Ключевое слово this.
35. Конструкторы. Свойства.
36. Тип данных Массив в C#.
37. Одномерные массивы.
38. Прямоугольные массивы.
39. Ступенчатые массивы.
40. Класс System.Array.
41. Оператор foreach для работы с массивом.
42. Массивы объектов.
43. Символы.
44. Массивы символов.
45. Строки типа string.
46. Класс System.String.
47. Форматирование строк.
48. Строки типа StringBuilder
49. Перегрузка методов.
50. Рекурсивные методы.
51. Методы с переменным количеством аргументов.
52. Метод Main.
53. Индексаторы.
54. Операции класса. Унарные и бинарные операции в классах.
55. Операции преобразования типа.
56. Деструкторы.
57. Вложенные типы.
58. Наследование классов. Описание класса-потомка. Раннее и позднее связывания.
59. Виртуальные методы.
60. Абстрактные классы.
61. Бесплодные классы.
62. Взаимоотношения между классами.
63. UML.
64. Класс System.Object.
65. Технологии создания графических интерфейсов.
66. Создание графического приложения. Запуск приложения.
67. Основы форм. Структура формы. Основные свойства форм.
68. Программная настройка свойств. Установка размеров формы. Начальное расположение формы. Фон и цвета формы.
69. Добавление форм. Взаимодействие между формами.
70. События в Windows Forms. События формы.

71. Понятие контейнера.
72. Динамическое добавление элементов.
73. Элементы GroupBox, Panel. FlowLayoutPanel. TableLayoutPanel.
74. Размеры элементов и их позиционирование в контейнере: позиционирование, установка размеров, свойства Anchor и Dock.
75. Панели вкладок TabControl и SplitContainer. Управление вкладками в коде.
76. Элементы управления.
77. Понятие потока. Стандартные потоки Console.In, Console.Out, Console.Error.
78. Классы потоков. Класс Stream. Классы байтовых потоков. Классы символьных потоков. Классы двоичных потоков.
79. Класс FileStream и файловый ввод-вывод на побайтовой основе.
80. Открытие и закрытие файла.
81. Запись данных в файл. Считывание байтов из объекта класса FileStream.
82. Файловый ввод-вывод с ориентацией на символы.
83. Использование классов StreamWriter и StreamReader.
84. Считывание и запись двоичных данных. Классы BinaryWriter и BinaryReader.
85. Навигация по файловой системе: классы для работы с файловой системой, получение сведений о файле, копирование файлов, перечисление файлов в папке, изменение расширения файлов
86. Многопоточность программы.
87. Создание потока в C#. Класс Thread. Запуск и остановка потока в C#.
88. Приоритеты потоков в C#.
89. Изменение типов потоков в C#.
90. Синхронизация потоков. Управление временем существования потоков.
91. Краткий обзор ADO.NET.
92. Пространство имен System.Data.
93. Поставщики данных ADO.NET.
94. Поведение объектов подключения.
95. Объектная модель ADO.NET. Объекты ADO .NET: таблицы, поля, строки, отношения, DataAdapter, DataView.
96. Интерфейсы соединения, транзакции, команды, адаптеров данных, чтения данных.
97. Создание соединения с источником данных. Командная строка соединения connectionString. Управление соединением. Объект Connection.
98. Обработка исключений при работе с MS SQL.
99. Работа с базой данных с помощью SQL-запросов. Объект Command: создание и инициализация.
100. Командная строка SQL-запроса CommandText. Выборка, добавление, удаление и обновление данных.

101. Вывод информации с использованием объекта чтения данных.
102. Объекты параметризованных команд.
103. Хранимые процедуры.
104. Отображение информации из базы данных с помощью элементов управления ListBox, ComboBox, ListView, DataGridView, DataTable.
105. Использование отсоединенных (автономных) наборов данных. Создание подключения к базе данных. Управляемые провайдеры. Обновление базы данных с помощью адаптера данных.
106. Динамическое получение объектов команд во время работы с помощью построителя команд SqlCommandBuilder

## ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО  
ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**

на \_\_\_\_ / \_\_\_\_ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры  
\_\_\_\_\_ (протокол № \_\_\_\_ от \_\_\_\_\_ 202\_ г.)

Заведующий кафедрой  
\_\_\_\_\_

УТВЕРЖДАЮ  
Декан факультета  
\_\_\_\_\_