

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Кабардино-Балкарский государственный университет  
им. Х.М. Бербекова» (КБГУ)

ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ЦИФРОВЫХ ТЕХНОЛОГИЙ  
КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННОЙ  
БЕЗОПАСНОСТИ

СОГЛАСОВАНО

УТВЕРЖДАЮ

Руководитель образовательной  
программы \_\_\_\_\_ Т.Ю.Хаширова

Директор института ИИиЦТ  
\_\_\_\_\_ А.Х. Шапсигов

«\_\_\_\_» \_\_\_\_\_ 2022 г.

«\_\_\_\_» \_\_\_\_\_ 2022 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**«Программирование на C#»**

Направление подготовки  
**09.03.01 Информатика и вычислительная техника**

Профиль  
Автоматизированные системы обработки информации и управления

Квалификация (степень) выпускника  
бакалавр

Форма обучения  
очная

Нальчик 2022

Рабочая программа дисциплины «Программирование на C#» /сост. Е.А. Акбашева –  
Нальчик: ФГБОУ КБГУ, 2022. – 31 с.

Рабочая программа предназначена для преподавания факультативной дисциплины студентам очной формы обучения по направлению подготовки 09.03.01 Информатика и вычислительная техника в 1 семестре 1 курса.

Рабочая программа дисциплины составлена с учетом Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденного приказом Министерства образования и науки Российской Федерации от «19» сентября 2017 г. №929 (зарегистрировано в Минюсте России 10 октября 2017 г. № 48489).

## СОДЕРЖАНИЕ

1. ЦЕЛЬ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	4
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО.....	4
3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	4
4. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ.....	6
5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО И РУБЕЖНОГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....	11
6. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ.....	21
7. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	22
8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	28
9. ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ.....	30
ПРИЛОЖЕНИЕ.....	31

## 1. ЦЕЛЬ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

**Целью** изучения дисциплины «Программирование на С#» является теоретическая и практическая подготовка студентов в области информационных технологий в такой степени, чтобы они могли выбирать необходимые технические, алгоритмические, программные и технологические решения, уметь объяснить принципы их функционирования и правильно их использовать.

### **Задачи:**

формирование у студентов знаний по дисциплине, достаточных для самостоятельного освоения новых систем и языков программирования;

выработка практических навыков написания программ на языках высокого уровня в современных средах программирования.

Изучение дисциплины направлено на подготовку специалистов, способных решать проблемы, возникающие при эксплуатации автоматизированных систем обработки информации и управления с учетом области, типов и задач профессиональной деятельности в соответствии с профессиональными стандартами:

- 06.001 – «Программист», утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 18 ноября 2013 г. № 679н (зарегистрирован Министерством юстиции Российской Федерации 18 декабря 2013 г., регистрационный № 30635), с изменением, внесенным приказом Министерства труда и социальной защиты Российской Федерации от 12 декабря 2016 г. № 727н (зарегистрирован Министерством юстиции Российской Федерации 13 января 2017 г., регистрационный № 45230);
- 06.022 – «Системный аналитик», утвержденный приказом Минтруда России от 28.10.2014 № 809н (зарегистрирован в Минюсте России 24.11.2014 № 34882);

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина относится к факультативным дисциплинам учебного плана по направлению 09.03.01 Информатика и вычислительная техника, профиль «Автоматизированные системы обработки информации и управления».

В результате освоения дисциплины обучающийся сможет частично продемонстрировать следующие обобщенные трудовые функции (ОТФ):

- Разработка требований и проектирование программного обеспечения (профессиональный стандарт 06.001 – «Программист», код D, уровень квалификации – 6).
- Концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности (профессиональный стандарт 06.022 – «Системный аналитик», код C, уровень квалификации – 6).

## 3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В совокупности с другими дисциплинами профиля дисциплина «Программирование на С#» направлена на формирование следующих компетенций в соответствии с ФГОС ВО и ОПОП ВО по направлению подготовки 09.03.01 – Информатика и вычислительная техника (ИиВТ) (уровень бакалавриата):

- Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности (ПКС-2);

**Коды и наименования индикаторов достижения компетенции:**

ПКС-2.1. Способен освоить методы планирования проектных работ; методы классического системного анализа; теорию управления бизнес-процессами; шаблоны оформления бизнес-требований; методы концептуального проектирования; методы оценки качества программных систем.

ПКС-2.2. Способен разрабатывать технико-экономическое обоснование; разрабатывать техническое задание на систему; разрабатывать требования к подсистемам системы и осуществлять контроль их качества; организовать оценку соответствия требованиям существующих систем и их аналогов; выполнять сопровождение приемочных испытаний и ввод в эксплуатацию системы; обрабатывать запросы на изменение требований к системе.

ПКС-2.3. Способен применить навыки составления графика контрольных мероприятий; приемами разработки бизнес-требований к системе; способностью определять ключевые свойства и ограничения системы; навыками выделения подсистем системы.

В результате освоения дисциплины обучающийся должен:

**знать:**

- основные теоретические положения структурного и объектно-ориентированного метода программирования и его свойства;
- состав, структуру и основные характеристики сред программирования, основанных на объектно-ориентированном методе;

**уметь:**

- выбирать наиболее оптимальную систему программирования исходя из постановки задачи, анализа аппаратной и программной платформы, а также финансовых ресурсов;
- иметь навыки по установке и настройке программного обеспечения одного из вариантов объектно-ориентированных сред программирования (например, Visual Studio).
- уверенно работать в визуальных средах программирования и разрабатывать с их помощью компьютерные программы для решения широкого класса задач;

**владеть:**

- Способностью овладеть основными методами, способами и средствами получения, хранения, переработки информации, иметь навыки работы с компьютером как средством управления информацией;
- Способностью использовать в познавательной и профессиональной деятельности навыки работы с информацией из различных источников;
- Способностью использовать в познавательной и профессиональной деятельности базовые знания в области информатики и современных информационных технологий, навыки использования программных средств и навыков работы в компьютерных сетях; умением создавать базы данных использовать ресурсы Интернет;
- способностью использовать базовые теоретические знания для решения профессиональных задач;
- способностью применять на практике базовые профессиональные навыки.

#### 4. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

В таблице 1 приводится описание содержания дисциплины, структурированное по разделам, с указанием по каждому разделу формы текущего контроля: защита лабораторной работы (ЛР), коллоквиум (К), рубежный контроль (РК), тестирование (Т).

Таблица 1

##### Содержание дисциплины «Программирование на C#»

№ раздела	Наименование раздела	Содержание раздела	Код контролируемой компетенции (или ее части)	Форма текущего контроля
1	Теоретические основы объектно-ориентированного программирования	Основные подходы к разработке программного обеспечения. Понятие объектно-ориентированного программирования (ООП). Понятие объекта и класса. Основные принципы ООП. Абстрагирование. Инкапсуляция. Агрегирование. Наследование. Объектная модель программы. Универсальный язык моделирования UML.	ПКС-2	ТК, К, Т
2	Основные понятия платформы Microsoft .Net	Основные идеи и компоненты платформы .NET Framework. Новый тип приложений – сборка (assembly). Метаданные. Промежуточный код (Intermediate Language). Единая среда выполнения (Common Language Runtime). Пространства имен (namespaces). Единая библиотека типов (классов, интерфейсов, структур) платформы - Microsoft Framework Library, основные пространства имен.	ПКС-2	ТК, К, Т, ЛР
3	Основные понятия языка программирования C#	Типы приложений. Общая структура программы. Пример простой программы. Использование командной строки для компиляции программы. Оператор using. Точка входа программы функция Main(). Статические методы. Базовые классы для консольных приложений. Класс Console. Класс Math. Класс Convert. Переменные методов. Типы данных. Стек. Куча. Система типов языка C#. Встроенные типы и преобразование типов. Константы.	ПКС-2	ТК, К, Т, ЛР
4	Операции и	Понятие операции. Приоритеты	ПКС-2	ТК, К, Т,

	операторы	операций. Операция присваивания. Специальные варианты присваивания. Арифметические операции. Вычисление выражений. Операции отношения. Логические операции. Условная операция. Понятие оператора. Оператор присваивания. Операторы выбора (if, switch). Операторы перехода (goto, break, continue). Операторы цикла (for, while, foreach). Обработка исключений.		ЛР
5	Работа с массивами	Типы массивов. Одномерные массивы. Многомерные массивы. Ступенчатые массивы. Массивы как коллекции. Методы класса Array. Создание и использование массивов.	ПКС-2	ТК, К, Т, ЛР
6	Коллекции и словари	Понятие коллекции – класса, предназначенного для хранения и доступа к упорядоченному списку объектов произвольного вида. Свойства и методы коллекций (Add, Remove, Item, Count). Создание коллекций объектов. Работа с объектами в коллекции - перечисление элементов коллекции (foreach). Нетипизированные коллекции. Типизированные коллекции.	ПКС-2	ТК, К, Т, ЛР
7	Встроенный язык поисковых запросов LINQ	Язык Language INTeGrated Query. Архитектура LINQ. Разновидности LINQ. Преимущества LINQ. Технология LINQ to Objects. Интерфейс IEnumerable<T>. Общий вид LINQ запроса. Операции LINQ запроса. Преобразование выборки в коллекции. Анонимные типы. LINQ и обобщенные коллекции.	ПКС-2	ТК, К, Т, ЛР
8	Описание классов	Основные элементы классов. Режимы доступа. Поля класса. Методы класса. Тело метода. Вызов метода. Перегрузка методов. Конструкторы класса. Свойства класса. Автоматически реализуемые свойства. Инициализация объектов класса. Индексаторы. Статические поля и методы класса. Переопределение операций класса. Определение преобразования типов.	ПКС-2	ТК, К, Т, ЛР
9	Делегаты и события классов	Новый тип данных – делегат (delegate). Объявление делегатов и создание экземпляров. Использование экземпляров делегатов. Взаимодействия объекта с вызывающей его программой с	ПКС-2	ТК, К, Т, ЛР

		помощью событий. Примеры событий. Описание событий в классе. Объявление в программе объектов с событиями. Создание обработчиков событий (event handler).		
10	Описание отношений между классами	Отношение вложенности. Отношение наследования. Описание производных классов. Конструкторы производного класса. Добавление методов в производном классе. Абстрактные классы. Присвоение ссылок с учетом наследования. Полиморфизм.	ПКС-2	ТК, К, Т, ЛР
11	Дополнительные пользовательские типы	Структуры. Перечисления. Интерфейсы. Два способа реализации интерфейса. Приведение к типу интерфейса. Встроенные интерфейсы.	ПКС-2	ТК, К, Т, ЛР
12	Создание форм и работа с ними	Пространство имен System.Windows.Forms. Основной класс окон – Form. Отображение и закрытие форм. Элементы управления – controls. Работа с элементами управления. Основные элементы управления, Виды меню. Основные классы по работе с меню. Основное и контекстное меню. Работа с диалоговыми окнами. Понятие диалогового окна. Типы диалоговых окон. Классы общих диалогов. Использование общих диалогов. Создание собственных диалогов. Рисование в окне программы. Пространство имен System.Windows.Drawing. Класс Graphics. Основные классы для рисования – Pen, Brush, Color, Font. Внешние устройства – клавиатура, мышь, таймер. События внешних устройств. Обработка событий формы от внешних устройств.	ПКС-2	ТК, К, Т, ЛР
13	Работа с файлами	Файловая систем. Структура файла. Классы для работы с каталогами Directory и DirectoryInfo. Классы для работы с файлами File и FileInfo. Классы для работы с содержанием файлов FileStream, StreamReader и StreamWriter, BinaryReader и BinaryWriter.	ПКС-2	ТК, К, Т, ЛР
14	Работа с базами данными	Назначение технологии ADO.Net. Реляционная модель данных. Язык работы с данными SQL. Основные операторы языка SQL (Select, Update, Delete). Понятие источника данных и его создание. Технологии ODBC и	ПКС-2	ТК, К, Т, ЛР



		<p>OleDb.  <b>Присоединенная работа с базами данных.</b> Провайдеры ADO.Net. Объектная модель провайдеров ADO.Net. Основные классы провайдеров. Класс Connection. Строка соединения с БД. Класс Command. Выполнение команд к базе данных с использованием объектов Command. Класс DataReader. Методы класса DataReader. Чтение, корректировка и добавление новых данных.  <b>Отсоединенная работа с БД.</b> Класс DataAdapter. Класс DataSet. Заполнение данных в DataSet из базы данных. Классы DataTable и DataRow. Работа с данными объекта DataTable. Сохранение изменений DataSet в базе данных. Создание типизированных классов DataSet. Использование объектов типизированного класса DataSet. Выполнение LINQ запросов к объектам DataSet.</p>		
--	--	--	--	--

Таблица 2

## Структура дисциплины

Вид работы	Трудоемкость, часы
	1 семестр
<b>Общая трудоемкость (в зачетных единицах)</b>	<b>3</b>
<b>Контактная работа (в часах):</b>	34
Лекции (Л)	17
Практические занятия (ПЗ)	17
Семинарские занятия (СЗ)	—
Лабораторные работы (ЛР)	—
<b>Самостоятельная работа (в часах), в том числе контактная работа:</b>	675
Курсовой проект (КП), курсовая работа (КР)	—
Расчетно-графическое задание (РГЗ)	—
Реферат (Р)	—
Эссе (Э)	—
Самостоятельное изучение разделов	65
Контрольная работа (К)	—
<b>Подготовка и прохождение промежуточной аттестации</b>	9
Вид промежуточной аттестации	зачет

Таблица 3

## Лекции

№ раз-дела	Наименование разделов
1	Теоретические основы объектно-ориентированного программирования
2	Основные понятия платформы Microsoft .Net
3	Основные понятия языка программирования C#
4	Операции и операторы в C#
5	Работа с массивами в C#
6	Коллекции и словари в C#
7	Встроенный язык поисковых запросов LINQ в C#
8	Описание классов в C#
9	Делегаты и события классов в C#
10	Описание отношений между классами в C#
11	Дополнительные пользовательские типы в C#
12	Создание форм и работа с ними в C#
13	Работа с файлами в C#
14	Работа с базами данных в C#

Таблица 4

## Практические работы

№ занятия	Наименование лабораторных работ
1.	Разработка элементарных программ на языке программирования C#
2.	Объекты и классы (наследование, конструкторы, деструкторы).
3.	Программирование полиморфных методов.
4.	Расширенные возможности языка программирования.
5.	Событийно-ориентированное программирование.

6.	Проектирование и реализация гетерогенных приложений
----	---

Таблица 5

## Самостоятельное изучение разделов дисциплины

№ раздела	Вопросы, выносимые на самостоятельное изучение
1	Обобщенные интерфейсы в C#.
2	Делегаты в C#.
3	События в C#.
4	Анонимные методы в C#.
5	Лямбда-выражения в C#.
6	Коллекции в C#.
7	Необобщенные коллекции в C#.
8	Обобщенные коллекции в C#.

## 5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО И РУБЕЖНОГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Формы контроля текущих, рубежных и промежуточных знаний студентов по дисциплине определяются в соответствии с учебным планом образовательной программы и в соответствии с действующим Положением о балльно-рейтинговой системе оценки успеваемости студентов КБГУ.

От обучающихся требуется посещение занятий, выполнение лабораторных работ, знакомство с рекомендованной литературой.

При аттестации обучающихся оценивается качество работы на занятиях (умение вести дискуссию, способность четко и ёмко формулировать свои мысли), уровень подготовки к самостоятельной деятельности, качество выполнения заданий (презентаций, докладов, выполнение лабораторных работ и др.).

Конечными результатами освоения программы дисциплины являются сформированные когнитивные дескрипторы «знать», «уметь», «владеть», расписанные по отдельным компетенциям. Формирование этих дескрипторов происходит в течение всего семестра по этапам в рамках различного вида занятий и самостоятельной работы.

В ходе изучения дисциплины предусматриваются текущий, рубежный контроль и промежуточная аттестация.

### 5.1. Оценочные материалы для текущего контроля

Цель текущего контроля – оценка результатов работы в семестре и обеспечение своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы обучающегося. Объектом текущего контроля являются конкретизированные результаты обучения (учебные достижения) по дисциплине.

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины «Программирование на C#» и включает: отчет по результатам выполнения лабораторных работ, самостоятельное выполнение заданий с отчетом (защитой) в установленный срок.

Оценка качества подготовки на основании выполненных заданий ведется преподавателем (с обсуждением результатов), баллы начисляются в зависимости от сложности задания.

### Критерии формирования оценок (оценивания) устного опроса

Устный опрос является одним из основных способов учёта знаний обучающегося по дисциплине. Развёрнутый ответ должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения.

В результате устного опроса знания, обучающегося оцениваются по следующей шкале:

Таблица 6

3 балла	2 балла	1 балл	0 баллов
ставится, если обучающийся: 1) полно излагает изученный материал, даёт правильное определенное экономических понятий; 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только по учебнику, но и самостоятельно составленные; 3) излагает материал последовательно и правильно с точки зрения норм литературного языка.	ставится, если обучающийся даёт ответ, удовлетворяющий тем же требованиям, что и для балла «1», но допускает 1-2 ошибки, которые сам же исправляет, и 1-2 недочёта в последовательности и языковом оформлении излагаемого.	ставится, если обучающийся обнаруживает знание и понимание основных положений данной темы, но: 1) излагает материал неполно и допускает неточности в определении понятий; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.	ставится, если обучающийся обнаруживает незнание большей части соответствующего раздела изучаемого материала, допускает ошибки в формулировке.

Баллы «1», «2», «3» могут ставиться не только за единовременный ответ, но и за рассредоточенный во времени, т.е. за сумму ответов, данных на протяжении занятия. начисляются в зависимости от сложности задания.

#### 5.1.1. Оценочные материалы для самостоятельной работы обучающегося

Рабочая программа предусматривает проведение лекционных, лабораторных занятий, а также самостоятельную работу обучающихся. В ФГБОУ ВО «Кабардино-Балкарский государственный университет» действует балльно-рейтинговая система оценки учебных достижений обучающихся по образовательным программам, реализуемым на основании федеральных государственных образовательных стандартов. Балльно-рейтинговая система оценки знаний является одной из составляющих системы управления качеством образовательной деятельности в университете.

#### Образцы заданий для самостоятельной работы

##### Задание 1

Составить на любом языке программирования консольное приложение, которое содержит описание класса **Time** (время), который должен содержать:

Класс должен включать:

- Закрытые свойства для хранения часов и минут
- Методы доступа к закрытым свойствам
- Конструктор или несколько конструкторов, для создания экземпляров класса
- Метод отображения на экране времени в формате (чч:мм)

Программа должна делать следующее:

1. В функции main() нужно объявить и создать массив из 3 объектов описанного класса
2. Задать им следующие значения (2ч 30м, 5ч 15м, 3ч 45м)
3. Вывести на экран время, хранящееся во всех объектах.
4. Рассчитать разницу в днях между 1 и 2 объектами и вывести ее на экран.

## Задание 2

Составить на любом языке программирования консольную программу, которая содержит описание класса

Date - дата (год, месяц, день)

Класс должен включать:

- Закрытые свойства для хранения год, месяц, день.
  - Методы доступа к закрытым свойствам.
  - Конструктор или несколько конструкторов, для создания объектов класса.
  - Метод - показать на экране время в формате (дд/мм/гг)
  - Метод - рассчитать количество дней с начала года до даты
- public int Days( )

Программа должна делать следующее:

1. В функции main() нужно объявить и создать массив из 3 объектов описанного класса
2. Задать им следующие значения (1.5.2001 5.2.2002 13.7.2001)
3. Вывести на экран даты, хранящиеся во всех объектах.
4. Рассчитать разницу в днях между 1 и 3 объектами и вывести ее на экран.

## **Критерии формирования оценок по заданиям для самостоятельной работы студента**

«Отлично» (5 баллов) – обучающийся показал глубокие знания материала по поставленным вопросам, грамотно, логично его излагает, структурировал и детализировал информацию, избегая простого повторения информации из текста, информация представлена в переработанном виде. Свободно использует необходимые формулы при решении задач;

«Хорошо» (4 балла) – обучающийся твердо знает материал, грамотно его излагает, не допускает существенных неточностей в процессе решения задач;

«Удовлетворительно» (3 балла) – обучающийся имеет знания основного материала по поставленным вопросам, но не усвоил его деталей, допускает отдельные неточности при решении задач;

«Неудовлетворительно» (менее 3 баллов) – обучающийся допускает грубые ошибки в ответе на поставленные вопросы и при решении задач

## **5.2. Оценочные материалы для рубежного контроля**

Рубежный контроль осуществляется по более или менее самостоятельным разделам – учебным модулям курса и проводится по окончании изучения материала модуля в заранее установленное время. Рубежный контроль проводится с целью определения качества усвоения материала учебного модуля в целом. В течение семестра проводится три таких контрольных мероприятия по графику.

В качестве форм рубежного контроля используется тестирование (письменное или компьютерное), проведение коллоквиума.

Рубежный контроль освоения студентом дисциплины осуществляется в рамках балльно-рейтинговой системы. Распределение баллов в соответствии с действующим Положением о балльно-рейтинговой системе оценки успеваемости студентов КБГУ приведено в таблице 7.

Таблица 7

Распределение баллов в соответствии с действующим Положением о балльно-рейтинговой системе

№ рейтинговой точки	Коллоквиум	Лаб.практикум	Посещаемость	Тестирование	Итого
1	7	8	3	5	23
2	7	8	3	5	23
3	7	8	4	5	24

Таблица 8

#### Критерии оценки

Вид мероприятия	Критерии оценки	Баллы
Коллоквиум (устный опрос по теме)	- ясность, четкость и доказательность изложения ответов на вопросы; - владение специальными терминами; - системность знаний по тематике	0-21 балл
Лабораторное занятие	- понимание цели и задач работы - выполнение заданий и обработка результатов - отчет и защита лабораторной работы	0-24 балла
Компьютерное тестирование по разделам дисциплины	Результаты тестирования (Количество баллов = $5 \cdot \varphi$ , $\varphi$ - доля правильно отвеченных тестов по теме).	0-15 баллов
Посещение занятий	При более 3 пропусках без уважительной причины занятий аннулируются баллы	0-10 баллов
Зачет	ясность, четкость и доказательность изложения ответов на вопросы; - владение специальными терминами; - системность знаний по тематике дисциплины в целом	0-30 баллов
Итоговая оценка		0-100 баллов

#### 5.2.1. Оценочные материалы для проведения коллоквиума

##### Перечень вопросов, выносимых на рейтинговый контроль первой точки (контролируемые компетенции ПКС-2):

1. Обзор .NET. Основные понятия.
2. Принципы объектно-ориентированного программирования. Язык C# и платформа .NET
3. JIT-компиляция.
4. Компиляция в командной строке.
5. Типы данных и переменные в C#.
6. Объявление переменных в C#.
7. Неявная типизация в C#.
8. Область видимости (контекст) переменных в C#.

9. Преобразования базовых типов данных в C#.
10. Явные и неявные преобразования в C#.
11. Операции языка C#.
12. Преобразования базовых типов данных в C#.
13. Явные и неявные преобразования в C#.
14. Массивы в C#.
15. Многомерные массивы в C#.
16. Массив массивов в C#.
17. Методы и свойства массивов в C#.
18. Условные конструкции в C#.
19. Конструкция switch/case Тернарная операция в C#.
20. Циклы в C#.
21. Цикл foreach в C#.
22. Операторы continue и break в C#.

**Перечень вопросов, выносимых на рейтинговый контроль второй точки (контролируемые компетенции ПКС-2):**

1. Методы в C#.
2. Функции в C#.
3. Параметры методов в C#.
4. Передача параметров по ссылке и модификатор ref в C#.
5. Необязательные параметры методов в C#.
6. Именованные параметры методов в C#.
7. Массив параметров и ключевое слово params в C#.
8. Рекурсивные функции в C#.
9. Перечисления enum в C#.
10. Структуры в C#.
11. Конструкторы в структурах в C#.
12. Обработка исключений в C#.
13. Обработка исключений и условные конструкции в C#.
14. Фильтры исключений в C#.
15. Работа с консолью и класс Console в C#.
16. Типы значений и ссылочные типы в C#.
17. Копирование значений в C#.
18. Ссылочные типы внутри типов значений в C#.
19. Объекты классов как параметры методов в C#.
20. Классы и объекты в C#.
21. Инициализаторы объектов в C#.
22. Частичные классы в C#.

**Перечень вопросов, выносимых на рейтинговый контроль третьей точки (контролируемые компетенции ПКС-2):**

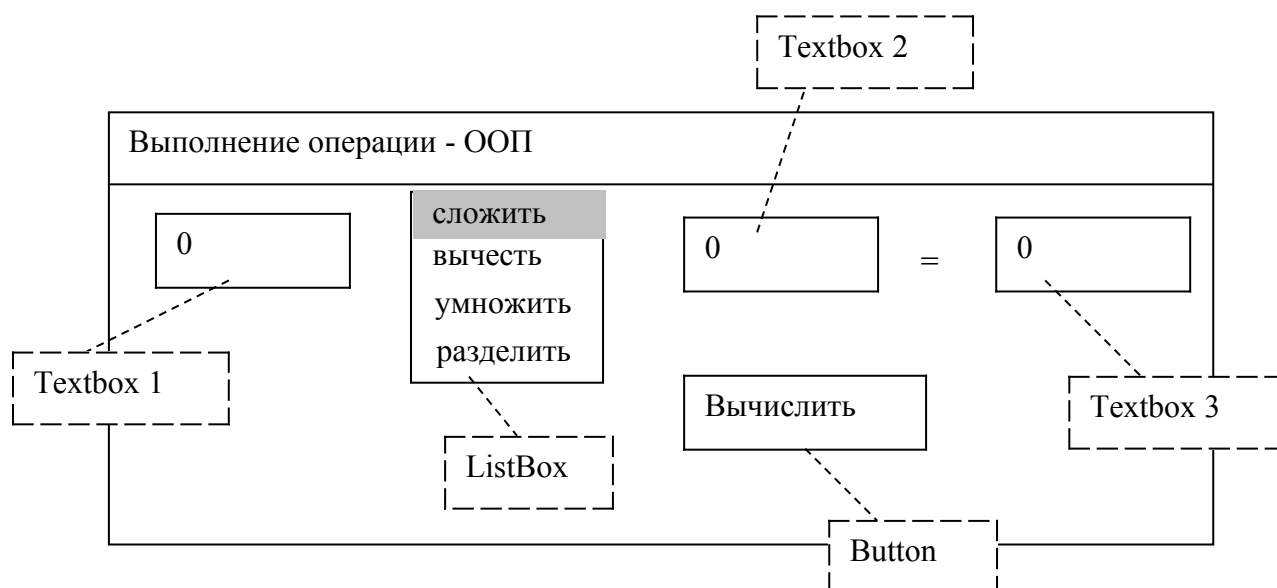
1. Модификаторы доступа в C#.
2. Константы и поля для чтения в C#.
3. Инкапсуляция в C#.
4. Автоматические свойства в C#.
5. Перегрузка методов и операторов в C#.
6. Перегрузка операторов в C#.
7. Статические члены класса в C#.
8. Статический конструктор в C#.
9. Статические классы в C#.

10. Наследование в C#.
11. Доступ к членам базового класса из класса-наследника в C#.
12. Ключевое слово base в C#.
13. Конструкторы в производных классах в C#.
14. Полиморфизм и переопределение методов в C#.
15. Абстрактные классы в C#.
16. Класс System.Object и его методы в C#.
17. Преобразование типов в C#.
18. Обобщенные типы в C#.
19. Анонимные типы в C#.
20. Методы расширения в C#.
21. Пространства имен в C#.
22. Псевдонимы в C#.
23. Интерфейсы в C#.

### Образцы заданий рубежного контроля

#### Задание 1

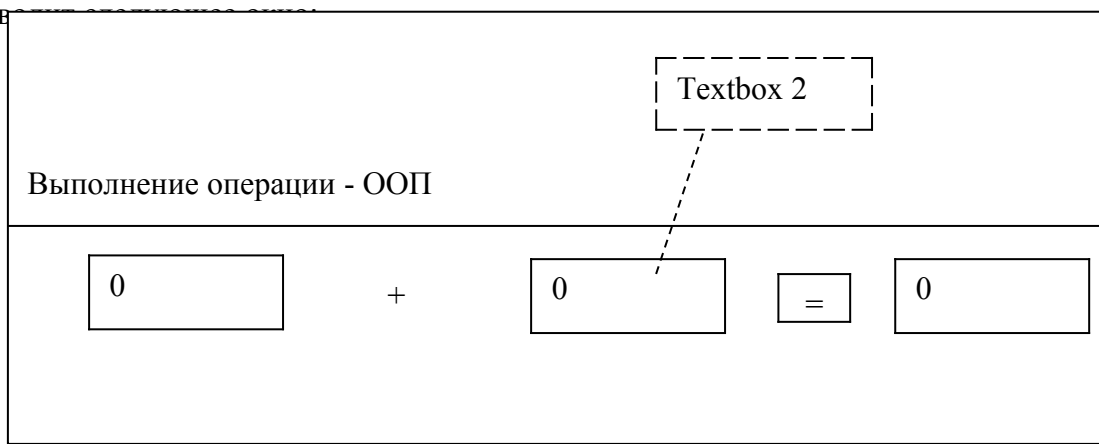
1. Написать программу, которая, используя объект класса производного от класса Form, выводит следующее окно:



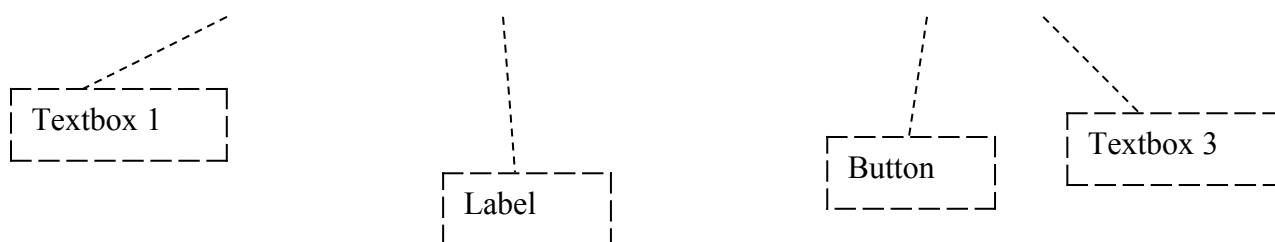
При нажатии кнопки Button, текстовое поле Textbox3 должно показать результат выбранной в ListBox операции над числами, введенными в Textbox1 и Textbox2

#### Задание 2

1. Написать программу, которая, используя объект класса производного от класса Form, выводит следующее окно:







При нажатии кнопки Button, текстовое поле Textbox3 должно показать результат выбранной в ListBox операции над числами, введенными в Textbox1 и Textbox2

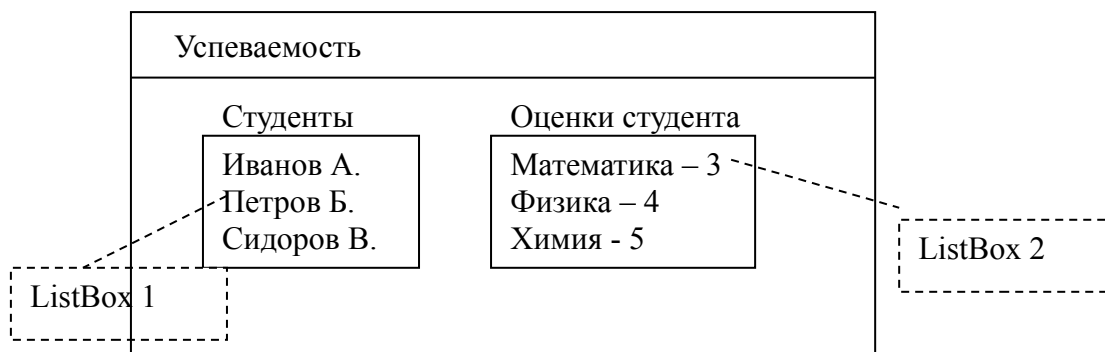
При выборе студента в ListBox1 в ListBox2 должны показываться его оценки.

### Задание 3

1. Написать программу, которая, используя объект класса производного от класса Form, позволяет просматривать информацию об успеваемости студентов по годам:

ФИО студента	Математика	Физика	Химия
Иванов А.	4	5	4
Петров Б.	3	4	5
Сидоров В.	5	5	5

Форма окна программы:



При выборе студента в ListBox1 в ListBox2 должны показываться его оценки.

### Критерии формирования оценок по контрольным точкам (коллоквиум)

4 балла ставится за работу, выполненную полностью без ошибок и недочетов; обучающийся демонстрирует знание теоретического и практического материала по теме практической работы, решено 100% задач;

3 балла ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов. Обучающийся демонстрирует знание теоретического и практического материала по теме практической работы, допуская незначительные неточности при решении задач, решено 70% задач;

2 балла ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой.

Обучающийся затрудняется с правильной оценкой предложенной задачи, дает неполный ответ, решено 55% задач

1 балл ставится за работу, если число ошибок и недочетов превысило норму или правильно выполнено менее 2/3 всей работы. Обучающийся дает неверную оценку ситуации, решено менее 50 % задач.

### 5.2.2. Оценочные материалы: тестирование

Тестирование обучающихся проводится в онлайн-режиме согласно расписанию в ЭИОС open.kbsu.ru

#### Примерные тестовые задания (контролируемые компетенции ПКС-2)

Полный перечень тестовых заданий представлен в ЭОИС

1. В контексте информационных технологий среда, обеспечивающая выполнение программного кода – это

- + : платформа
- : инфраструктура среды
- : интерфейс пользователя
- : операционная система

2. Инфраструктура среды выполнения программ, нечто, определяющее особенности разработки и выполнения программного кода на данной платформе

- + : Framework
- : платформа
- : интерфейс пользователя
- : операционная система

3. Общая спецификация языков программирования, набор конструкций и ограничений, которые являются руководством для создателей библиотек и компиляторов в среде .NET Framework

- + : CLS (Common Language Specification)
- : CLR (Common Language Runtime)
- : Управляемый код
- : Microsoft .NET

4. Программный код, который при своем выполнении способен использовать службы, предоставляемые CLR

- + : Управляемый код
- : Microsoft .NET
- : CLS (Common Language Specification)
- : CLR (Common Language Runtime)

5. Соответствующая CLS-спецификации объектно-ориентированная библиотека классов, интерфейсов и системы типов (типов-значений), которые включаются в состав платформы Microsoft .NET

- + : FCL (.NET Framework Class Library)
- : CLS (Common Language Specification)
- : CLR (Common Language Runtime)
- : ASP.NET

6. Классы .NET, предназначенные для разработки приложений, работающих с базами данных

- + : ADO .NET
- : .NET FCL
- : ASP.NET
- : CLR
- 7. Промежуточный язык платформы Microsoft .NET
  - + : MSIL (Microsoft Intermediate Language)
  - : FCL (.NET Framework Class Library)
  - : CLS (Common Language Specification)
  - : CLR (Common Language Runtime)
- 8. Базовый строительный блок приложения в .NET Framework
  - + : Сборка (Assembly)
  - : Библиотека
  - : Класс
  - : Объект
- 9. Способ организации системы типов в единую группу
  - + : Пространство имен
  - : Сборка (Assembly)
  - : Библиотека
  - : FCL (.NET Framework Class Library)
- 10. Оболочка, которая содержит множество классов, объединенных, как правило, общей тематикой или группой разработчиков
  - + : Пространство имен
  - : Сборка (Assembly)
  - : Библиотека
  - : FCL (.NET Framework Class Library)

### **Критерии формирования оценок по тестовым заданиям**

Выполнение тестирования оценивается согласно проценту правильных ответов. Максимально возможное количество баллов за тестирование – 5.

### **5.3. Оценочные материалы для промежуточной аттестации**

Целью промежуточных аттестаций по дисциплине является оценка качества освоения дисциплины обучающимися.

Промежуточная аттестация предназначена для объективного подтверждения и оценивания достигнутых результатов обучения после завершения изучения дисциплины. Осуществляется в конце семестра и представляет собой итоговую оценку знаний по дисциплине «Программирование на C#» в виде проведения экзамена.

Промежуточная аттестация может проводиться в устной, письменной форме, и в форме тестирования. На промежуточную аттестацию отводится до 30 баллов.

### **Вопросы итоговой аттестации по дисциплине (контролируемые компетенции ПКС-2)**

1. Обзор .NET. Основные понятия.
2. Принципы объектно-ориентированного программирования. Язык C# и платформа .NET
3. JIT-компиляция.

4. Компиляция в командной строке.
5. Типы данных и переменные в C#.
6. Объявление переменных в C#.
7. Неявная типизация в C#.
8. Область видимости (контекст) переменных в C#.
9. Преобразования базовых типов данных в C#.
10. Явные и неявные преобразования в C#.
11. Операции языка C#.
12. Преобразования базовых типов данных в C#.
13. Явные и неявные преобразования в C#.
14. Массивы в C#.
15. Многомерные массивы в C#.
16. Массив массивов в C#.
17. Методы и свойства массивов в C#.
18. Условные конструкции в C#.
19. Конструкция switch/case Тернарная операция в C#.
20. Циклы в C#.
21. Цикл foreach в C#.
22. Операторы continue и break в C#.
23. Методы в C#.
24. Функции в C#.
25. Параметры методов в C#.
26. Передача параметров по ссылке и модификатор ref в C#.
27. Необязательные параметры методов в C#.
28. Именованные параметры методов в C#.
29. Массив параметров и ключевое слово params в C#.
30. Рекурсивные функции в C#.
31. Перечисления enum в C#.
32. Структуры в C#.
33. Конструкторы в структурах в C#.
34. Обработка исключений в C#.
35. Обработка исключений и условные конструкции в C#.
36. Фильтры исключений в C#.
37. Работа с консолью и класс Console в C#.
38. Типы значений и ссылочные типы в C#.
39. Копирование значений в C#.
40. Ссылочные типы внутри типов значений в C#.
41. Объекты классов как параметры методов в C#.
42. Классы и объекты в C#.
43. Инициализаторы объектов в C#.
44. Частичные классы в C#.
45. Модификаторы доступа в C#.
46. Константы и поля для чтения в C#.
47. Инкапсуляция в C#.
48. Автоматические свойства в C#.
49. Перегрузка методов и операторов в C#.
50. Перегрузка операторов в C#.
51. Статические члены класса в C#.
52. Статический конструктор в C#.

53. Статические классы в C#.
54. Наследование в C#.
55. Доступ к членам базового класса из класса-наследника в C#.
56. Ключевое слово `base` в C#.
57. Конструкторы в производных классах в C#.
58. Полиморфизм и переопределение методов в C#.
59. Абстрактные классы в C#.
60. Класс `System.Object` и его методы в C#.
61. Преобразование типов в C#.
62. Обобщенные типы в C#.
63. Анонимные типы в C#.
64. Методы расширения в C#.
65. Пространства имен в C#.
66. Псевдонимы в C#.
67. Интерфейсы в C#.

#### **Критерии формирования оценок по промежуточной аттестации**

**«Отлично»** получают обучающиеся, которые свободно ориентируются в материале и отвечают без затруднений. Обучающийся способен к выполнению сложных заданий, постановке целей и выборе путей их реализации. Работа выполнена полностью без ошибок, решено 100% задач;

**«Хорошо»** получают обучающиеся, которые относительно полно ориентируются в материале, отвечают без затруднений, допускают незначительное количество ошибок. Обучающийся способен к выполнению сложных заданий. Работа выполнена полностью, но имеются не более одной негрубой ошибки и одного недочета, не более трех недочетов. Допускаются незначительные неточности при решении задач, решено 70% задач;

**«Удовлетворительно»** получают обучающиеся, у которых недостаточно высок уровень владения материалом. В процессе ответа на экзамене допускаются ошибки и затруднения при изложении материала. Обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой. Обучающийся затрудняется с правильной оценкой предложенной задачи, дает неполный ответ, решено 55% задач;

**«Неудовлетворительно»** получают обучающиеся, которые допускают значительные ошибки. Обучающийся имеет лишь начальную степень ориентации в материале. В работе число ошибок и недочетов превысило норму для оценки 3 или правильно выполнено менее 2/3 всей работы. Обучающийся дает неверную оценку ситуации, решено менее 50% задач.

### **6. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ**

Максимальная сумма (100 баллов), набираемая студентом по дисциплине включает две составляющие:

- первая составляющая – оценка регулярности, своевременности и качества выполнения студентом учебной работы по изучению дисциплины в течение периода изучения дисциплины (семестра, или нескольких семестров) (сумма – не более 70 баллов). Баллы, характеризующие успеваемость студента по дисциплине,

набираются им в течение всего периода обучения за изучение отдельных тем и выполнение отдельных видов работ.

- вторая составляющая – оценка знаний студента по результатам промежуточной аттестации (не более 30 –баллов).

Критерием оценки уровня сформированности компетенций в рамках учебной дисциплины «Программирование на С#» в 1 семестре является зачет.

Общий балл текущего и рубежного контроля складывается из следующих составляющих (Приложение).

Целью промежуточных аттестаций по дисциплине является оценка качества освоения дисциплины обучающимися.

Таблица 9

Результаты освоения учебной дисциплины, подлежащие проверке

Результаты обучения (компетенции)	Освоенные показатели оценки результатов обучения	Вид оценочного материала
<p>ПКС-2</p> <p>Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности</p>	<p><b>ИД-1<sub>ПКС-2</sub>. Знать:</b> современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности</p> <p><b>ИД-2<sub>ПКС-2</sub>. Уметь:</b> выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.</p> <p><b>ИД-оп<sub>ПКС-2</sub>. Владеть:</b> навыками: применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.</p>	<p>Типовые оценочные материалы для устного опроса, типовые тестовые задания, лабораторные работы, задания для самостоятельной работы (раздел 5)</p>

## 7. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 7.1. Нормативно-правовая база

1. ГОСТ «Единая система программной документации».
2. ГОСТ Р ИСО/МЭК ТО 15271-2002 «Руководство по применению ГОСТ Р ИСО/МЭК 12270 (Процессы жизненного цикла программных средств).
3. ГОСТ Р ИСО/МЭК ТО 12182-2010 «Классификация программных средств».
4. ISO/IEC 14764:2006 «Разработка программного обеспечения. Процессы жизненного цикла программного обеспечения. Сопровождение».
5. ГОСТ Р ИСО/МЭК 12119-2000 «Информационная технология. Пакеты программ. Требования к качеству и тестирование»
6. ГОСТ 28195-89 «Оценка качества программных средств. Общие положения».
7. ISO/IEC 25000:2005 «Технология программного обеспечения. Требования и оценка качества программного продукта. Руководство».
8. ISO/IEC 25001:2014 «Программирование. Требования к качеству программного продукта и его оценка. Планирование и менеджмент».
9. ISO/IEC 25010:2011 «Проектирование систем и разработка программного обеспечения. Требования к качеству систем и программного обеспечения и их оценка (SQuaRE). Модели качества систем и программного обеспечения».

10. ISO/IEC 25012:2008 «Программная инженерия – Требования к качеству и оценке программного обеспечения. Модель качества данных».
11. ISO/IEC 25020:2007 «Разработка программного обеспечения. Требования к качеству и оценка качества программного продукта. Измерительная эталонная модель и руководство».

## 7.2. Основная литература

1. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс] / Б. Мейер. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 285 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/39552.html>
2. Николаев Е.И. Объектно-ориентированное программирование [Электронный ресурс] : учебное пособие / Е.И. Николаев. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2015. — 225 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/62967.html>
3. Букунов С.В. Основы объектно-ориентированного программирования [Электронный ресурс] : учебное пособие / С.В. Букунов, О.В. Букунова. — Электрон. текстовые данные. — СПб. : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017. — 196 с. — 978-5-9227-0713-8. — Режим доступа: <http://www.iprbookshop.ru/74339.html>
4. Биллиг В.А. Основы объектного программирования на С# (С# 3.0, Visual Studio 2008) [Электронный ресурс] : учебное пособие / В.А. Биллиг. — Электрон. текстовые данные. — Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 583 с. — 978-5-4487-0145-0. — Режим доступа: <http://www.iprbookshop.ru/72339.html>
5. Марченко А.Л. Основы программирования на С# 2.0 [Электронный ресурс] : учебное пособие / А.Л. Марченко. — Электрон. текстовые данные. — Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 552 с. — 978-5-4487-0084-2. — Режим доступа: <http://www.iprbookshop.ru/67382.html>

## 7.3. Дополнительная литература

1. Казанский А.А. Объектно-ориентированное программирование на языке Microsoft Visual C# в среде разработки Microsoft Visual Studio 2008 и .NET Framework. 4.3 [Электронный ресурс] : учебное пособие и практикум / А.А. Казанский. — Электрон. текстовые данные. — М. : Московский государственный строительный университет, ЭБС АСВ, 2011. — 180 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/19258.html>
2. Лисицин Д.В. Объектно-ориентированное программирование [Электронный ресурс] : конспект лекций / Д.В. Лисицин. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2010. — 88 с. — 978-5-7782-1454-5. — Режим доступа: <http://www.iprbookshop.ru/44970.html>
3. Фленов М. Е. Библия С#, 2-е издание. СПб: БХВ-Петербург, 2011, 560с.

4. Шилд Г. С# 4.0: Полное руководство. М: Вильямс, 2011, 1056 с.
5. Адриянова А.А., Исмагилов Л. Н., Мухтарова Т. М. Объектно-ориентированное программирование на С#: Учебное пособие. Казань: Казанский (Приволжский) Федеральный Университет, 2012, 134 с.
6. Эрих Гамма, Ричард Хелм, Ральф Джонсон, Джон Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования. Питер, 2016.
7. Мэтт Вайсфельд. Объектно-ориентированное мышление. Питер, 2014.
8. Эдвард Йордон, Карл Аргила. Объектно-ориентированный анализ и проектирование систем. Лори, 2014.

#### **7.4. Периодические издания (газета, вестник, бюллетень, журнал)**

1. Журнал «Объектно-ориентированное программирование для профессионалов».

#### **7.5. Интернет-ресурсы**

1. С.В. Зыков. Технологии и средства разработки корпоративных систем. Интуит.ру, <http://www.intuit.ru/department/itmngt/techcorpsys/>
2. А.В. Марчуков, А.О. Савельев. Работа в Microsoft Visual Studio. Интуит.ру, <http://www.intuit.ru/department/itmngt/workinmsvistudio/>
3. В.А. Биллиг. Основы программирования на С# 3.0: ядро языка. Интуит.ру, <http://www.intuit.ru/department/pl/tincsharp3/>
4. [habr.com](http://habr.com)
5. <http://www.intuit.ru>
6. <http://citforum.ru>

#### **7.6. Современные профессиональные базы данных**

1. База данных Science Index (РИНЦ) <http://elibrary.ru>
2. Национальная электронная библиотека РГБ <https://нэб.рф>
3. Крупнейшая единая база данных, содержащая аннотации и информацию о цитируемости рецензируемой научной литературы, со встроенными инструментами отслеживания, анализа и визуализации данных. [www.scopus.com](http://www.scopus.com)
4. Самая полная математическая база данных, охватывающая материалы с конца 19 века. zbMath содержит документы, журналы и книги по математике, статистике, информатике, а также машиностроению, физике, естественным наукам и др. [www.zbmath.org](http://www.zbmath.org) (доступ открытый).

#### **7.7. Методические указания по проведению различных учебных занятий и самостоятельной работы**

##### **Методические указания к лабораторным занятиям**

Выполнение каждой лабораторной работы складывается из следующих этапов.



Самостоятельная подготовка студентов к работе. Перед началом работы студенты должны четко представлять себе цель работы, знать схему, метод измерения, физическую сущность ожидаемых результатов. Должен быть подготовлен отчет, содержащий о порядке выполнения лабораторной работы. Студенты, не подготовившиеся к работе в соответствии с этими требованиями, к выполнению работы не допускаются.

Проведение эксперимента. Этот этап осуществляется в соответствии с методическими указаниями, которые содержатся в описании к каждой работе. Приступать к работам на стенде студент может начать только после ознакомления с теоретической частью и описания хода выполнения работы. Любые изменения в схеме проводятся при тщательной проверке схемы, для исключения короткого замыкания. Результаты выполнения проверяются преподавателем.

Составление отчета о проделанной работе. Отчёт должен содержать исчерпывающие данные, как о цели работы, так и о результатах в следующей последовательности: задание; схема установки и описание хода выполнения; результаты выполнения работы, включая рисунки, схемы, таблицы; общие выводы и заключение.

Текст отчета должен быть написан аккуратно и разборчиво от руки или представлен в виде распечатки, после компьютерной верстки. В обоих случаях текст должен представлять собой логическое изложение существа вопроса. Недопустимо приведение формул, таблиц без разъяснений всех обозначений и сокращений. Полученные зависимости должны сопровождаться теоретическим обоснованным объяснением причин, влияющих на их ход, для чего в процессе составления отчета студент обязан по литературным источникам ознакомиться с материалом, который был объектом его исследования в лаборатории. Без такого ознакомления с испытуемым методом студент не будет в состоянии дать правильный анализ процессов, происходящих в материале при эксперименте.

Защита лабораторной работы с представлением отчета. При сдаче отчета студенты должны показать понимание сущности проведенных исследований, объяснить полученные результаты и сделать выводы. При работе в лаборатории необходимо строго выполнять все правила техники безопасности и указания преподавателя.

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

Самостоятельная работа обучающихся – способ активного, целенаправленного приобретения обучающимися новых для него знаний и умений без непосредственного участия в этом процесса преподавателей. Повышение роли самостоятельной работы обучающихся при проведении различных видов учебных занятий предполагает:

- оптимизацию методов обучения, внедрение в учебный процесс новых технологий обучения, повышающих производительность труда преподавателя, активное использование информационных технологий, позволяющих обучающемуся в удобное для него время осваивать учебный материал;
- широкое внедрение компьютеризированного тестирования;
- совершенствование методики проведения практик и научно-исследовательской работы обучающихся, поскольку именно эти виды учебной работы в первую очередь готовят обучающихся к самостоятельному выполнению профессиональных задач;
- модернизацию системы курсового и дипломного проектирования, которая должна повышать роль обучающихся в подборе материала, поиске путей решения задач.

Самостоятельная работа приводит обучающихся к получению нового знания, упорядочению и углублению имеющихся знаний, формированию у него профессиональных навыков и умений. Самостоятельная работа выполняет ряд функций:

- развивающую;
- информационно-обучающую;
- ориентирующую и стимулирующую;
- воспитывающую;
- исследовательскую.

В рамках курса выполняются следующие виды самостоятельной работы:

- Проработка учебного материала (по конспектам, учебной и научной литературе);
- Выполнение разноуровневых заданий;
- Работа с тестами и вопросами для самопроверки;
- Выполнение итоговой контрольной работы.

Обучающимся рекомендуется с самого начала освоения курса работать с литературой и предлагаемыми заданиями в форме подготовки к очередному аудиторному занятию. При этом актуализируются имеющиеся знания, а также создается база для усвоения нового материала, возникают вопросы, ответы на которые обучающийся получает в аудитории.

Необходимо отметить, что некоторые задания для самостоятельной работы по курсу имеют определенную специфику. При освоении курса обучающийся может пользоваться библиотекой вуза, которая в полной мере обеспечена соответствующей литературой. Значительную помощь в подготовке к очередному занятию может оказать имеющийся в учебно-методическом комплексе краткий конспект лекций. Он же может использоваться и для закрепления полученного в аудитории материала. Самостоятельная работа обучающихся предусмотрена учебным планом и выполняется в обязательном порядке. Задания предложены по каждой изучаемой теме и могут готовиться индивидуально или в группе. По необходимости обучающийся может обращаться за консультацией к преподавателю. Выполнение заданий контролируется и оценивается преподавателем.

Для успешной организации самостоятельной работы все активнее применяются разнообразные образовательные ресурсы в сети Интернет: системы тестирования по различным областям, виртуальные лекции, лаборатории, при этом пользователю достаточно иметь компьютер и подключение к Интернету для того, чтобы связаться с преподавателем, решать вычислительные задачи и получать знания. Использование сетей усиливает роль самостоятельной работы обучающихся и позволяет кардинальным образом изменить методику преподавания.

Обучающийся может получать все задания и методические указания через сервер, что дает ему возможность привести в соответствие личные возможности с необходимыми для выполнения работ трудозатратами. Обучающийся имеет возможность выполнять работу дома или в аудитории. Большое воспитательное и образовательное значение в самостоятельном учебном труде обучающийся имеет самоконтроль. Самоконтроль возбуждает и поддерживает внимание и интерес, повышает активность памяти и мышления, позволяет обучающемуся своевременно обнаружить и устранить допущенные ошибки и недостатки, объективно определить уровень своих знаний, практических умений. Самое доступное и простое средство самоконтроля с применением информационно-коммуникационных технологий - это ряд тестов «on-line», которые позволяют в режиме реального времени определить свой уровень владения предметным материалом, выявить свои ошибки и получить рекомендации по самосовершенствованию.

### **Методические рекомендации по работе с литературой**

Всю литературу можно разделить на учебники и учебные пособия, оригинальные научные монографические источники, научные публикации в периодической печати. Из них можно выделить литературу основную (рекомендуемую), дополнительную и литературу для углубленного изучения дисциплины.

Изучение дисциплины следует начинать с учебника, поскольку учебник – это книга, в которой изложены основы научных знаний по определенному предмету в соответствии с целями и задачами обучения, установленными программой.

При работе с литературой необходимо учитывать, что имеются различные виды чтения, и каждый из них используется на определенных этапах освоения материала.

Предварительное чтение направлено на выявление в тексте незнакомых терминов и поиск их значения в справочной литературе. В частности, при чтении указанной литературы необходимо подробнейшим образом анализировать понятия.

Сквозное чтение предполагает прочтение материала от начала до конца. Сквозное чтение литературы из приведенного списка дает возможность студенту сформировать свод основных понятий из изучаемой области и свободно владеть ими.

Выборочное – наоборот, имеет целью поиск и отбор материала. В рамках данного курса выборочное чтение, как способ освоения содержания курса, должно использоваться при подготовке к практическим занятиям по соответствующим разделам.

Аналитическое чтение – это критический разбор текста с последующим его конспектированием. Освоение указанных понятий будет наиболее эффективным в том случае, если при чтении текстов студент будет задавать к этим текстам вопросы. Часть из этих вопросов сформулирована в ФОС в перечне вопросов для собеседования. Перечень этих вопросов ограничен, поэтому важно не только содержание вопросов, но сам принцип освоения литературы с помощью вопросов к текстам.

Целью изучающего чтения является глубокое и всестороннее понимание учебной информации. Есть несколько приемов изучающего чтения:

1. Чтение по алгоритму предполагает разбиение информации на блоки: название; автор; источник; основная идея текста; фактический материал; анализ текста путем сопоставления имеющихся точек зрения по рассматриваемым вопросам; новизна.
2. Прием постановки вопросов к тексту имеет следующий алгоритм:
  - медленно прочитать текст, стараясь понять смысл изложенного;
  - выделить ключевые слова в тексте;
  - постараться понять основные идеи, подтекст и общий замысел автора.
3. Прием тезирования заключается в формулировании тезисов в виде положений, утверждений, выводов.

К этому можно добавить и иные приемы: прием реферирования, прием комментирования.

Важной составляющей любого солидного научного издания является список литературы, на которую ссылается автор. При возникновении интереса к какой-то обсуждаемой в тексте проблеме всегда есть возможность обратиться к списку относящейся к ней литературы. В этом случае вся проблема как бы разбивается на составляющие части, каждая из которых может изучаться отдельно от других. При этом важно не терять из вида общий контекст и не погружаться чрезмерно в детали, потому что таким образом можно не увидеть главного.

Подготовка к экзамену должна проводиться на основе лекционного материала, материала практических занятий с обязательным обращением к основным учебникам по курсу. Это позволит исключить ошибки в понимании материала, облегчит его осмысление, прокомментирует материал многочисленными примерами.

### **Методические рекомендации для подготовки к зачету с оценкой**

Дифференцированный зачет является формой итогового контроля знаний и умений обучающихся по данной дисциплине, полученных на лекциях, лабораторных занятиях и в процессе самостоятельной работы. Основой для определения оценки служит уровень усвоения обучающимися материала, предусмотренного данной рабочей программой. К зачету допускаются студенты, набравшие 36 и более баллов по итогам текущего и промежуточного контроля. На зачете студент может набрать от 15 до 30 баллов.

В период подготовки к зачету обучающиеся вновь обращаются к учебно-методическому материалу и закрепляют промежуточные знания.

Подготовка обучающегося к зачету включает три этапа:

- самостоятельная работа в течение семестра;
- непосредственная подготовка в дни, предшествующие зачету по темам курса;
- подготовка к ответу на вопросы к зачету.

При подготовке к зачету обучающимся целесообразно использовать материалы лекций, учебно-методические комплексы, нормативные документы, основную и дополнительную литературу.

На зачет выносится материал в объеме, предусмотренном рабочей программой учебной дисциплины за семестр. Зачет проводится в письменной / устной форме.

При проведении зачета в письменной (устной) форме, ведущий преподаватель составляет билеты, которые включают в себя: тестовые задания; теоретические задания; задачи или ситуации. Формулировка теоретических задания совпадает с формулировкой перечня вопросов, доведенных до сведения обучающихся накануне экзаменационной сессии. Содержание вопросов одного билета относится к различным разделам программы с тем, чтобы более полно охватить материал учебной дисциплины.

В аудитории, где проводится устный зачет, должно одновременно находиться не более шести студентов на одного преподавателя, принимающего экзамен. На подготовку ответа на билет на зачете отводится 40 минут.

При проведении письменного зачета на работу отводится 60 минут.

Результат устного (письменного) зачета выражается оценками:

Оценка «отлично» – от 91 до 100 баллов – теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы. Все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному. На зачете студент демонстрирует глубокие знания предусмотренного программой материала, умеет четко, лаконично и логически последовательно отвечать на поставленные вопросы.

Оценка «хорошо» – от 81 до 90 баллов – теоретическое содержание курса освоено, необходимые практические навыки работы сформированы, выполненные учебные задания содержат незначительные ошибки. На зачете студент демонстрирует твердые знания основного (программного) материала, умеет четко, грамотно, без существенных неточностей отвечать на поставленные вопросы.

Оценка «удовлетворительно» – от 61 до 80 баллов – теоретическое содержание курса освоено не полностью, необходимые практические навыки работы сформированы частично, выполненные учебные задания содержат грубые ошибки. На зачете студент демонстрирует знание только основного материала, ответы содержат неточности, слабо аргументированы, нарушена последовательность изложения материала.

Оценка «неудовлетворительно» – от 36 до 60 баллов – теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к существенному повышению качества выполнения учебных заданий. На зачете студент демонстрирует незнание значительной части программного материала, существенные ошибки в ответах на вопросы, неумение ориентироваться в материале, незнание основных понятий дисциплины.

## **8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **8.1. Требования к материально-техническому обеспечению**

Минимально необходимый для реализации ОПОП перечень материально-технического обеспечения включает в себя: лекционные аудитории (оборудованные видеопроекционным оборудованием для презентаций, средствами звуковоспроизведения, экраном и имеющие выход в сеть Интернет), помещения для проведения семинарских и практических занятий (оборудованные учебной мебелью), компьютерные классы и др. (в соответствии с ФГОС и учебным планом).

Во время самостоятельной работы студенты используют компьютерные классы института информатики, электроники и компьютерных технологий, электронные читальные залы КБГУ и домашние компьютеры.

**Лицензионное программное обеспечение, используемое для проведения лекционных и лабораторных занятий**

1. Microsoft Windows 10.
2. Microsoft Office 2016.
3. Visual Studio 2019.

## **8.2. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья**

Для студентов с ограниченными возможностями здоровья созданы специальные условия для получения образования. В целях доступности получения высшего образования по образовательным программам инвалидами и лицами с ограниченными возможностями здоровья университетом обеспечивается (аудитория для самостоятельной работы и коллективного пользования специальными техническими средствами для обучения инвалидов и лиц с ОВЗ в КБГУ, аудитория № 145 Главный корпус КБГУ):

1. Альтернативная версия официального сайта в сети «Интернет» для слабовидящих;
2. Для инвалидов с нарушениями зрения (слабовидящие, слепые):
  - присутствие ассистента, оказывающего обучающемуся необходимую помощь, дублирование вслух справочной информации о расписании учебных занятий; наличие средств для усиления остаточного зрения, брайлевской компьютерной техники, видеоувеличителей, программ невидимого доступа к информации, программ-синтезаторов речи и других технических средств приема-передачи учебной информации в доступных формах для студентов с нарушениями зрения;
  - задания для выполнения на экзамене зачитываются ассистентом;
  - письменные задания выполняются на бумаге, надиктовываются ассистенту обучающимся;
3. Для инвалидов и лиц с ограниченными возможностями здоровья по слуху (слабослышащие, глухие):
  - на зачете/экзамене присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочесть и оформить задание, в том числе записывая под диктовку);
  - зачет/экзамен проводится в письменной форме;
4. Для инвалидов и лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, созданы материально-технические условия, обеспечивающие возможность беспрепятственного доступа обучающихся в учебные помещения, объекты питания, туалетные и другие помещения университета, а также пребывания в указанных помещениях (наличие расширенных дверных проемов, поручней и других приспособлений).
  - письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;
  - по желанию студента экзамен проводится в устной форме.

Обучающиеся из числа лиц с ограниченными возможностями здоровья обеспечены электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

**9. ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ**

Рабочая программа:

одобрена на 2022/2023 учебный год. Протокол № \_\_\_\_\_ заседания кафедры от  
«\_\_\_» \_\_\_\_\_ 20\_\_ г.

В рабочую программу внесены следующие изменения:

---

---

---

---

---

---

Разработчик программы \_\_\_\_\_

Зав. кафедрой \_\_\_\_\_

## ПРИЛОЖЕНИЕ

## Распределение баллов текущего и рубежного контроля

№п/п	Вид контроля	Сумма баллов			
		Общая сумма	1-я точка	2-я точка	3-я точка
1	Посещение занятий	до 10 б.	до 3 б.	до 3б.	до 4б.
2	Текущий контроль:	до 28 б.	до 9 б.	до 9 б.	до 10 б.
	Выполнение лабораторных работ	до 18 б.	до 6 б.	до 6 б.	до 6 б.
	Выполнение самостоятельных заданий	от 0 до 10 б.	от 0 до 3 б.	от 0 до 3 б.	от 0 до 4 б.
3	Рубежный контроль	до 27 баллов	до 9 б.	до 9 б.	до 9 б.
	тестирование	от 0 до 15 б.	от 0 до 5 б.	от 0 до 5 б.	от 0 до 5 б.
	коллоквиум	от 0 до 12 б.	от 0 до 4 б.	от 0 до 4 б.	от 0 до 4 б.
4	Итого сумма текущего и рубежного контроля	до 70 б.	до 23 б.	до 23 б.	до 24 б.