

**МИНОБРНАУКИ РОССИИ**  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**  
(БГТУ им. В.Г. Шухова)

УТВЕРЖДАЮ  
Директор института ЭИТУС  
А.В. Белоусов  
« 24 » 05 2021 г.



**РАБОЧАЯ ПРОГРАММА**  
дисциплины

**Программная инженерия**

направление подготовки

09.03.03 Прикладная информатика

Направленность программы

Прикладная информатика в бизнесе

Квалификация

бакалавр

Форма обучения

очная


Институт: Энергетики, информационных технологий и управляющих систем

Кафедра: Информационных технологий

Белгород 2021

Рабочая программа составлена на основании требований:

- Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению 09.03.03 Прикладная информатика, утвержденного Приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 922
- учебного плана, утвержденного ученым советом БГТУ им. В.Г. Шухова в 2021 году.

Составитель: ст.преп.  (Е.А.Лазебная)  
(ученая степень и звание, подпись) (инициалы, фамилия)

Рабочая программа обсуждена на заседании кафедры

«30» 04 2021 г., протокол № 6

И.о. зав. кафедрой: канд.техн.наук  (Д.Н. Старченко)  
(ученая степень и звание, подпись) (инициалы, фамилия)

Рабочая программа согласована с выпускающей кафедрой  
информационных технологий

И.о. зав. кафедрой: канд.техн.наук  (Д.Н. Старченко)  
(ученая степень и звание, подпись) (инициалы, фамилия)

«30» 04 2021 г.

Рабочая программа одобрена методической комиссией института

«20» 05 2021 г., протокол № 9

Председатель: канд.техн.наук, доц.  (А.Н. Семернин)  
(ученая степень и звание, подпись) (инициалы, фамилия)

## 1. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Категория (группа) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Наименование показателя оценивания результата обучения по дисциплине
	ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1. Понимает принципы работы современных информационных технологий и программных средств	Знание общих характеристик современных информационных технологий и программных средств; этапов разработки ПС при решении задач профессиональной деятельности; основных процессов программной инженерии и вопросов их выполнения; методов и инструментов программной инженерии; приемов и инструментов тестирования.
		ОПК-2.2. Уметь: выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.	Умение применять современные информационные технологии и программные средства (.NET Framework, Visual Studio C#) при решении задач профессиональной деятельности.
		ОПК-2.3. Иметь навыки: применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.	Владение навыками применения современных информационных технологий и программных средств (.NET Framework, Visual Studio C#), при решении задач профессиональной деятельности; разработки консольных приложений и приложений с визуальным интерфейсом.
	ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Использует методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий	Знание методов алгоритмизации, основ объектно-ориентированного программирования; основных типов данных языка программирования C# и основные приемы организации работы с ними; библиотеки классов BCL платформы .NET Framework; элементов управления среды Visual Studio C#.

		ОПК-7.2. Разрабатывает алгоритмы и выбирает языки и технологии программирования для решения профессиональных задач в области информационных систем и технологий	Умение применять принципы объектно-ориентированного программирования при решении профессиональных задач в области информационных систем и технологий; проводить выбор исходных данных для проектирования; описывать собственные классы и использовать классы библиотеки классов BCL платформы .NET Framework; делать выбор среди элементов управления среды Visual Studio C# для разработки визуального интерфейса приложения.
		ОПК-7.3. Программирует, проводит отладку и тестирование прототипов программно-технических комплексов задач	Владение навыками использования технологии объектно-ориентированного программирования и визуального программирования интерфейса программных средств; приемами и инструментами отладки и тестирования ПС.
	ПК-4. Способен проводить тестирование компонентов программного обеспечения ИС.	ПК-4.1. Определяет подходы, методы и принципы тестирования компонентов программного обеспечения ИС	Знание общих характеристик современных информационных технологий и программных средств используемых при тестировании ПС; подходов, методов и принципов тестирования компонентов ПО ИС.
		ПК 4-2. Применяет подходы, и методы тестирования компонентов программного обеспечения ИС	Умение применять современные информационные технологии и программные средства (.NET Framework, Visual Studio C#) при тестировании компонентов ПО ИС.
		ПК 4.3. Проводит тестирование компонентов ИС по заданным сценариям	Владение навыками применения современных информационных технологий и программных средств (.NET Framework, Visual Studio C#), при решении задачи тестирования компонентов ПО ИС.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

### 1. Компетенция ОПК-2

Данная компетенция формируется следующими дисциплинами.

Стадия	Наименования дисциплины
1	Информационные технологии
2	Базы данных

3	Программная инженерия
4	Большие данные
5	Инструментальные средства информационных систем
6	Интеллектуальные системы и технологии
7	Информационная безопасность и защита персональных данных
8	Учебная технологическая (проектно-технологическая) практика

## **2. Компетенция ОПК-7**

Данная компетенция формируется следующими дисциплинами.

Стадия	Наименования дисциплины
1	Информационные технологии
2	Алгоритмизация и программирование
3	Программная инженерия
4	Информатика
5	Учебная ознакомительная практика
6	Учебная технологическая (проектно-технологическая) практика

## **2. Компетенция ПК-4**

Данная компетенция формируется следующими дисциплинами.

Стадия	Наименования дисциплины
1	Алгоритмизация и программирование
2	Программная инженерия
3	Надежность информационных систем
4	Производственная преддипломная практика

### 3. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 5 зач. единиц, 180 часов.

Форма промежуточной аттестации дифференцированный зачет

Вид учебной работы	Всего часов	Семестр № 4
Общая трудоемкость дисциплины, час	180	180
<b>Контактная работа (аудиторные занятия), в т.ч.:</b>		
лекции	51	51
лабораторные	34	34
практические		
групповые консультации в период теоретического обучения и промежуточной аттестации	5	5
<b>Самостоятельная работа студентов, включая индивидуальные и групповые консультации, в том числе:</b>		
Курсовой проект		
Курсовая работа		
Расчетно-графическое задание		
Индивидуальное домашнее задание		
Самостоятельная работа на подготовку к аудиторным занятиям (лекции, практические занятия, лабораторные занятия)	90	90
Диф.зачет		Диф.зачет

## 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1 Наименование тем, их содержание и объем Курс 2 Семестр 4

№ п/п	Наименование раздела (краткое содержание)	Объем на тематический раздел по видам учебной нагрузки, час			
		Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа на подготовку к аудиторным занятиям
1. Основы объектно-ориентированного программирования					
	Основные понятия и принципы объектно-ориентированного программирования. Объектно-ориентированный подход к проектированию и разработке программ. Событийное программирование.	2			1
2. Интегрированная среда разработки Visual Studio					
	Платформа .NET Framework. CLR-среда. Библиотека классов. Пространства имен, классы, методы классов.	2			1
3. Основы языка программирования C#					
	Структура проекта. Структура программы. Основные конструкции языка программирования C#.	2		2	5
4. Система типов языка программирования C#.					
	Типы данных языка C#. Значимые и ссылочные типы данных. Структуры. Массивы. Строки. Перечисления. Работа с файлами	4		4	10
5. Классы					
	Понятие класса и объекта. Члены класса. Описание класса. Создание объекта. Модификаторы доступа класса и членов класса. Инкапсуляция. Наследование. Полиморфизм. Абстрактные классы. Перегрузка методов. Виртуальные методы. Статические классы.	6		4	10
6. Технология создания Windows Forms					
	Элементы управления среды Visual Studio C#. Свойства формы. Управление свойствами и поведением элементов управления. Элементы управления для отображения и ввода данных. Элемент управления для построения диаграмм и графиков.	2		6	12
7. Методы класса Graphics.					
	Классы графического интерфейса приложений C#. Класс Graphics и его методы. Класс Bitmap. Элемент управления PictureBox.	2			4
8. Делегаты. Безопасность и обработка исключений					
	Понятие делегата. Описание, создание и применение делегата. Исключительные ситуации. Конструкции языка C# для обеспечения безопасности и обработки исключений.	4		4	8
9. Разработка пользовательских элементов управления					

	Выбор шаблона для создания пользовательского элемента управления, Событие. Создание событий, основанных на делегатах.	4		2	7
10. Интерфейсы. Коллекции					
	Понятие и применение интерфейсов. Создание новых интерфейсов и реализация существующих. Массивы и коллекции. Обобщенные и необобщенные коллекции. Коллекции ArrayList, List<T>, BindingList<T>. Базовые интерфейсы коллекций.	44		4	9
11. Организация работы с коллекциями объектов в приложении					
	Элементы управления DataGridView, TreeView. Приемы работы с коллекцией объектов в приложении. Технология Drag-and-Drop, Сериализация и Десериализация.	2		4	8
12. Программная инженерия					
	Основные процессы программной инженерии. Общие вопросы выполнения процессов программной инженерии. Методы и инструменты программной инженерии	2			2
13. Качество программного обеспечения.					
	Характеристики качества ПО. Методики повышения качества ПО. Относительная эффективность методик контроля качества ПО	2			1
14. Тестирование, выполняемое разработчиками					
	Тестирование и качество ПО. Приемы тестирования. Инструменты тестирования. Типичные ошибки. Оптимизация процесса тестирования. Протоколы тестирования. Разработка тестов. Среда тестирования JUnit. Самотестирующийся код	3		4	7
15. Отладка					
	Общие вопросы. Поиск и устранение дефекта. Инструменты отладки	2			1
16. Рефакторинг					
	Введение в рефакторинг. Отдельные виды и стратегии рефакторинга. Каталог методов рефакторинга	2			2
17. Методы рефакторинга					
	Выделение методов. Встраивание методов. Встраивание временной переменной. Введение поясняющей переменной. Решение задач обобщения. Упрощение вызовов методов. Крупные рефактоинги. Инструментальные средства проведения рефакторинга	2			2
	ВСЕГО	51		34	90



## 4.2. Содержание практических занятий

Не предусмотрено учебным планом

## 4.3. Содержание лабораторных занятий

№ п/п	Наименование раздела дисциплины	Тема лабораторного занятия	К-во часов	Самостоятельная работа на подготовку к аудиторным занятиям
<b>семестр № 4</b>				
1	Основы языка программирования С#	Лабораторная работа №1. Интегрированная среда разработки Visual Studio. Основы языка программирования С#	2	3
2	Система типов языка программирования С#.	Лабораторная работа №2. Система типов языка программирования С#. Структуры. Массивы. Сложность алгоритмов	2	4
3	Классы	Лабораторная работа №3. Основы объектно-ориентированного программирования	2	4
4	Технология создания Windows Forms	Лабораторная работа №4. Технология Windows Forms. Элементы управления среды Visual Studio С#	4	6
5	Делегаты. Безопасность и обработка исключений	Лабораторная работа №5. Делегаты. Безопасность и обработка исключений. Построение графиков функций	2	4
6	Технология создания Windows Forms	Лабораторная работа №6. Методы класса Graphics. Элемент управления DataGridView	2	4
7	Интерфейсы. Коллекции	Лабораторная работа №7. Массивы и коллекции С#	4	4
8	Технология создания Windows Forms Организация работы с коллекциями объектов в приложении	Лабораторная работа №8,9. Иерархии классов, элемент управления TreeView и технология Drag-and-Drop	6	9
9	Разработка пользовательских элементов управления	Лабораторная работа № 10 Пользовательские элементы управления и компоненты. Разработка простых и составных элементов управления.	2	4
10	Технология создания Windows Forms	Лабораторная работа № 11 Документирование кода с помощью XML-комментариев	4	6
11	Тестирование, выполняемое разработчиками	Лабораторная работа № 12 Тестирование ПС	4	6
<b>ИТОГО:</b>			<b>34</b>	<b>64</b>
<b>ВСЕГО:</b>			<b>34</b>	<b>98</b>

#### 4.4. Содержание курсового проекта/работы

Не предусмотрено учебным планом

#### 4.5. Содержание расчетно-графического задания, индивидуальных домашних заданий

Не предусмотрено учебным планом

### 5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

#### 5.1. Реализация компетенций

**1 Компетенция** ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности

Наименование индикатора достижения компетенции	Используемые средства оценивания
ОПК-2.1. Понимает принципы работы современных информационных технологий и программных средств	Собеседование, защита лабораторной работы, тестовый контроль, устный опрос, зачет
ОПК-2.2. Использует современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	Собеседование, защита лабораторной работы, тестовый контроль, устный опрос, зачет
ОПК-2.3. Осуществляет выбор современных информационных технологий и программных средств, в том числе отечественного производства, для решения задач профессиональной деятельности	Собеседование, защита лабораторной работы, тестовый контроль, устный опрос, зачет

**2 Компетенция** ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий

Наименование индикатора достижения компетенции	Используемые средства оценивания
ОПК-7.1. Использует методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий	Собеседование, защита лабораторной работы, тестовый контроль, устный опрос, зачет
ОПК-7.2. Разрабатывает алгоритмы и выбирает языки и технологии программирования для решения профессиональных задач в области информационных систем и технологий	Собеседование, защита лабораторной работы, тестовый контроль, устный опрос, зачет
ОПК-7.3. Программирует, проводит отладку и тестирование прототипов программно-технических комплексов задач	Собеседование, защита лабораторной работы, тестовый контроль, устный опрос, зачет

**2 Компетенция** ПК-4. Способность проводить тестирование компонентов

программного обеспечения ИС.

Наименование индикатора достижения компетенции	Используемые средства оценивания
ПК-4.1. Определяет подходы, методы и принципы тестирования компонентов программного обеспечения ИС	Собеседование, защита лабораторной работы, тестовый контроль, устный опрос, зачет
ПК 4-2. Применяет подходы, и методы тестирования компонентов программного обеспечения ИС	Собеседование, защита лабораторной работы, тестовый контроль, устный опрос, зачет
ПК 4.3. Проводит тестирование компонентов ИС по заданным сценариям	Собеседование, защита лабораторной работы, тестовый контроль, устный опрос, зачет

## 5.2. Типовые контрольные задания для промежуточной аттестации

### 5.2.1. Перечень контрольных вопросов для зачета

№ п/п	Наименование раздела дисциплины	Содержание вопросов (типовых заданий)
1	Основы объектно-ориентированного программирования	Основные принципы ООП Полиморфизм Наследование Инкапсуляция Абстрактные классы. Классы, закрытые для наследования. Статические классы. Полиморфизм и реализация полиморфизма. Абстрактные и виртуальные методы. Перегрузка методов.
2	Интегрированная среда разработки Visual Studio	Основные компоненты платформы <i>.NET Framework</i> Общезыковая среда выполнения (CLR) Библиотека классов. Этапы создания приложения в <i>.NET Framework</i> . Архитектура платформы <i>.NET Framework</i> . Элементы окна интегрированной среды <i>Visual Studio C#</i> .
3	Основы языка программирования C#	Пространство имен платформы <i>.NET Framework</i> . Ускоренные методы доступа к методам классов. Структура проекта консольного приложения. Структура программы на языке C#. Оператор языка C# <code>foreach</code> . Консольный ввод/вывод.
4	Система типов языка программирования C#.	Система общих типов. Наиболее востребованные встроенные типы. Область действия и время существования переменных. Параметры методов. Преобразования типов данных. Операции упаковки и распаковки.
5	Классы	Понятие класса и объекта Члены класса, их назначение Описание классов в C# Модификаторы доступа классов Модификаторы доступа членов класса Перегрузка методов

		<p>Перегрузка операций</p> <p>Статические классы и поля</p> <p>Автореализуемое свойство</p> <p>Параметры методов</p>
6	Технология создания Windows Forms	<p>Технология <i>Windows Forms</i></p> <p>Класс <i>Control</i> его свойства и события</p> <p>Свойство <i>Controls</i> класса <i>Control</i></p> <p>Общие для всех элементов управления свойства</p> <p>Статические элементы управления</p> <p>Кнопки и переключатели</p> <p>Горизонтальная и вертикальная полосы прокрутки</p> <p>Элементы управления с поддержкой редактирования текста</p> <p>Список и поле со списком</p> <p>Наборные счётчики</p> <p>Древоподобное и списковое представление</p> <p>Разделители</p> <p>Меню и панели инструментов</p> <p>Диалоговые окна</p> <p>Класс <i>MessageBox</i></p> <p>Модальные и немодальные окна и формы</p> <p>Организация взаимодействия одной формы с другой</p>
7	Методы класса Graphics	<p>Назначение элемента управления <i>Chart</i> и его наиболее важные свойства</p> <p>Типы диаграмм для представления данных класса <i>Chart</i></p> <p>Интерфейс графических устройств GDI</p> <p>Методы класса <i>Graphics</i></p> <p>Система координат окна, для которого получен объект <i>Graphics</i></p> <p>Варианты метода <i>DrawLine</i> для рисования одной и нескольких линий</p> <p>Варианты метода <i>DrawRectangles</i> для рисования одного и нескольких прямоугольников</p> <p>События <i>MouseUp</i>, <i>MouseDown</i>, <i>MouseMove</i></p> <p>Использование объекта <i>Bitmap</i></p> <p>Использование объекта <i>PictureBox</i></p>
8	Делегаты. Безопасность и обработка исключений	<p>Понятие делегата</p> <p>Описание и создание делегата</p> <p>Порядок работы с делегатом</p> <p>В каких случаях используют делегаты</p> <p>Безопасность и обработка исключений</p> <p>Генерация исключений</p> <p>Синтаксис операторов <i>try</i>, <i>catch</i> и <i>finally</i></p>
9	Разработка пользовательских элементов управления	<p>Элементы управления для создания справочной поддержки приложения</p> <p>Добавление и настройка всплывающих подсказок к элементам управления приложения</p> <p>Добавление и настройка строки состояния для формы</p> <p>Добавление и настройка окна справки «О программе»</p> <p>Свойства формы для настройки строки-заголовка</p>
10	Интерфейсы. Коллекции	<p>Назначение и свойства элемента управления <i>DataGridView</i></p> <p>Настройка несвязанного <i>DataGridView</i></p> <p>Настройка <i>DataGridView</i>, связанного с источником данных</p> <p>Допустимые источники данных для <i>DataGridView</i></p> <p>Обеспечение двустороннего механизма связи между</p>

		<p><i>DataGridView</i> и коллекцией</p> <p>Понятие и назначение интерфейса</p> <p>Объявление интерфейса в С#</p> <p>Особенности реализации интерфейса в классах</p> <p>Понятие и назначение коллекций в С#</p> <p>Типы коллекций в С#</p> <p>Сравнение обобщенных и необобщенных коллекций</p>
11	Организация работы с коллекциями объектов в приложении	<p>Классы обобщенных коллекций</p> <p>Интерфейсы, представляющие базовую функциональность коллекций в пространстве имен <i>System.Collections</i></p> <p>Понятие и назначение коллекции <i>ArrayList</i></p> <p>Интерфейсы, реализованные в классе <i>System.Array</i></p> <p>Примеры методов класса <i>System.Array</i></p> <p>Порядок работы с коллекцией <i>ArrayList</i></p> <p>Особенности сортировки и поиска элементов в коллекции <i>ArrayList</i></p> <p>Порядок создания главного меню в приложении</p> <p>Порядок создания контекстного меню в приложении</p> <p>Наследование и реализация наследования в С#.</p> <p>Управление доступностью классов и его членов при наследовании.</p> <p>Построение схемы классов в среде <i>Visual Studio С#</i>.</p> <p>Элементы управления для древовидного и спискового представления данных</p> <p>Создание объекта класса <i>TreeNode</i> и получение доступа к узлам класса <i>TreeNode</i></p> <p>Методы развёртывания/свёртывания узлов.</p> <p>Управление видом элемента управления <i>TreeView</i></p> <p>Назначение и порядок реализации механизма <i>Drag-and-Drop</i></p> <p>Назначение и порядок реализации сериализации/десериализации.</p>
12	Программная инженерия	<p>Понятие программной инженерии</p> <p>Основные процессы программной инженерии.</p> <p>Общие вопросы выполнения процессов программной инженерии.</p> <p>Методы программной инженерии</p> <p>Инструменты программной инженерии</p>
13	Качество программного обеспечения	<p>Характеристики качества ПО.</p> <p>Методики повышения качества ПО.</p> <p>Относительная эффективность методик контроля качества ПО</p>
14	Тестирование, выполняемое разработчиками	<p>Тестирование и качество ПО.</p> <p>Приемы тестирования.</p> <p>Инструменты тестирования.</p> <p>Типичные ошибки.</p> <p>Оптимизация процесса тестирования.</p> <p>Протоколы тестирования.</p> <p>Разработка тестов.</p> <p>Среда тестирования JUnit.</p> <p>Самотестирующийся код</p>
15	Отладка	<p>Общие вопросы отладки.</p> <p>Устранение дефекта.</p> <p>Поиск дефекта.</p>

		Инструменты отладки
16	Рефакторинг	Понятие рефакторинга Отдельные виды рефакторинга Отдельные стратегии рефакторинга Методы рефакторинга Каталог методов рефакторинга
17	Методы рефакторинга	Выделение методов. Встраивание методов. Встраивание временной переменной. Введение поясняющей переменной. Решение задач обобщения. Упрощение вызовов методов. Крупные рефактинги. Инструментальные средства проведения рефакторинга

### 5.2.2. Перечень контрольных материалов для защиты курсового проекта/ курсовой работы

Не предусмотрено учебным планом

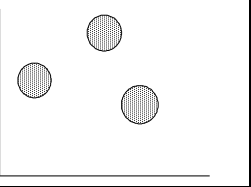
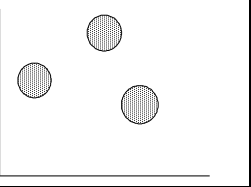
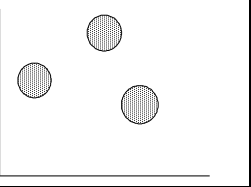
### 5.3. Типовые контрольные задания (материалы) для текущего контроля в семестре

Контроль знаний студентов осуществляется в процессе выполнения и защиты лабораторных работ.

"Выполнение" лабораторной работы предполагает демонстрацию студентом результатов выполнения заданий, а именно отчета и необходимых файлов (документов или программ). Полные перечни заданий с примерами выполнения приведены в методических указаниях. Примерные варианты заданий приведены в следующей таблице.

№	Тема лабораторной работы	Задание				
1	Лабораторная работа №1. Интегрированная среда разработки Visual Studio. Основы языка программирования C#	<p>Ознакомиться с разделами справочной службы MSDN для получения сведений об использовании интегрированной среды разработки <i>Visual C#</i>. Ознакомиться со встроенными типами данных в C#, массивами и строками. Разработать алгоритм решения задачи в соответствии с вариантом задания, написать программу на языке <i>Visual C#</i> и создать консольное приложение в среде разработки <i>Visual C#</i>. Решение задачи разделить на подзадачи: формирование исходного массива, вывод на экран элементов исходного массива, выполнение задания и вывод результатов на экран. Решение каждой подзадачи оформить в виде отдельных функций.</p> <p>Вариант задания:</p> <table border="1"> <thead> <tr> <th>Задание</th> <th>Элементы матрицы</th> </tr> </thead> <tbody> <tr> <td>Вычислить сумму и число положительных элементов каждого столбца матрицы. Результаты вывести в виде двух строк</td> <td> <math display="block">a_{ij} = \frac{\sqrt{i}}{2 + \sin j} \cos j</math> <math display="block">i = 0, \dots, 9</math> <math display="block">j = 0, \dots, 14</math> </td> </tr> </tbody> </table>	Задание	Элементы матрицы	Вычислить сумму и число положительных элементов каждого столбца матрицы. Результаты вывести в виде двух строк	$a_{ij} = \frac{\sqrt{i}}{2 + \sin j} \cos j$ $i = 0, \dots, 9$ $j = 0, \dots, 14$
Задание	Элементы матрицы					
Вычислить сумму и число положительных элементов каждого столбца матрицы. Результаты вывести в виде двух строк	$a_{ij} = \frac{\sqrt{i}}{2 + \sin j} \cos j$ $i = 0, \dots, 9$ $j = 0, \dots, 14$					
2	Лабораторная работа №2. Система типов языка программирования C#. Структуры. Массивы. Сложность алгоритмов	<ol style="list-style-type: none"> <li>Ознакомиться со встроенными типами данных в C#, структурами, массивами и строками.</li> <li>Ознакомиться с особенностями консольного ввода/вывода и файлового ввода/вывода в консольных приложениях C#.</li> <li>Задание, приведенное в варианте реализовать двумя способами. Для каждого из способов разработать алгоритм решения задачи в соответствии с требованиями, сформулированными ниже. Написать для каждого алгоритма текст программы на языке <i>Visual C#</i> и создать консольные приложения в среде разработки <i>Visual C#</i>.</li> </ol>				

		<p><u>Способ А.</u> Создать структуру, содержащую поля в соответствии с приведенным в варианте задания форматом строк (вся исходная информация считывается из файла). Решение задачи разделить на подзадачи: чтение данных из файла в массив, состоящий из N объектов структуры; вывод на экран элементов исходного массива; выполнение задания в соответствии с вариантом и вывод результатов на экран. Решение каждой подзадачи оформить в виде отдельного метода. Предусмотреть передачу значений между методами через параметры.</p> <p><u>Способ Б.</u> Для решения задачи разработать эффективный по времени выполнения и по используемой памяти алгоритм (без создания массива, состоящего из N объектов структуры). В случае необходимости можно использовать структуру и вспомогательный массив, размерность которого не зависит от N. Вся исходная информация должна считываться из файла.</p> <p>4. Оценить и сравнить сложность алгоритмов, реализованных при выполнении задания способами А и Б. Оценить и сравнить размер памяти, занимаемых объектами программ, реализующих задания способами А и Б. Оценить и сравнить время работы программ.</p> <p>Вариант задания:</p> <table border="1" data-bbox="635 696 1503 882"> <thead> <tr> <th data-bbox="635 696 1158 730">Формат строк, хранящихся в файле</th> <th data-bbox="1158 696 1503 730">Задание</th> </tr> </thead> <tbody> <tr> <td data-bbox="635 730 1158 882">Сведения о результатах участия в олимпиаде учащихся нескольких школ: &lt;Фамилия&gt; &lt;Инициалы&gt; &lt;Номер школы&gt; Где &lt;Номер школы&gt; – не более чем двузначный номер</td> <td data-bbox="1158 730 1503 882">Вывести на экран информацию из каких двух школ было меньше всего участников (таких школ может быть несколько)</td> </tr> </tbody> </table>	Формат строк, хранящихся в файле	Задание	Сведения о результатах участия в олимпиаде учащихся нескольких школ: <Фамилия> <Инициалы> <Номер школы> Где <Номер школы> – не более чем двузначный номер	Вывести на экран информацию из каких двух школ было меньше всего участников (таких школ может быть несколько)
Формат строк, хранящихся в файле	Задание					
Сведения о результатах участия в олимпиаде учащихся нескольких школ: <Фамилия> <Инициалы> <Номер школы> Где <Номер школы> – не более чем двузначный номер	Вывести на экран информацию из каких двух школ было меньше всего участников (таких школ может быть несколько)					
3	Лабораторная работа №3. Основы объектно-ориентированного программирования	<p>По заданному описанию класса в соответствии с вариантом создать класс на языке C# (в отдельном файле проекта).</p> <p>Разработать следующие элементы класса:</p> <ul style="list-style-type: none"> <li>- поля;</li> <li>- конструкторы с разными наборами параметров (2-3 конструктора);</li> <li>- свойства, позволяющие установить и получить значения полей класса с проверкой корректности ввода исходных значений;</li> <li>- методы;</li> <li>- перегрузки унарной операции, бинарной операции и констант true и false.</li> </ul> <p>Создать программу, использующую созданную структуру классов.</p> <p>Вариант задания:</p> <p>Класс <i>Rectangle</i>.</p> <p>Поля:</p> <ul style="list-style-type: none"> <li>- <i>int a, b</i> // длины сторон прямоугольника</li> </ul> <p>Методы:</p> <ul style="list-style-type: none"> <li>- вывод длин сторон прямоугольника на экран;</li> <li>- вычисление периметра прямоугольника;</li> <li>- вычисление площади прямоугольника.</li> </ul> <p>Перегрузки:</p> <ul style="list-style-type: none"> <li>- операции ++ (—): одновременно увеличивает (уменьшает) значение полей a и b;</li> <li>- констант <i>true</i> и <i>false</i>: обращение к экземпляру класса даёт значение <i>true</i>, если прямоугольник с заданными длинами сторон является квадратом, иначе <i>false</i>;</li> <li>- операции *: перемножает длины сторон двух прямоугольников.</li> </ul>				
4	Лабораторная работа №4. Технология Windows Forms. Элементы управления среды Visual Studio C#	<p>Познакомиться с видами элементов управления, свойствами форм их свойствами и способами работы с ними. Получить навыки в использовании различных видов элементов управления при разработке <i>Windows Forms</i> приложений, организации взаимодействия между формами. Создать эмуляцию формы согласно варианту.</p> <p>Варианты задания:</p> <p><u>Задание А.</u></p> <ol style="list-style-type: none"> <li>1. Эмуляция экранной формы Экран – Разрешение экрана.</li> <li>2. Эмуляция экранной формы пункта меню <i>MS Word</i>: Настройка– Параметры.</li> <li>3. Эмуляция экранной формы пункта меню <i>MS Word</i>: Разметка страницы – Параметры страницы.</li> </ol> <p><u>Задание Б.</u></p>				

		<p>Для класса, разработанного при выполнении лабораторной работы №2, создать <i>Windows Forms</i> приложение. Приложение должно предоставлять возможность создания и редактирования экземпляра класса, выполнения описанных в классе методов и перегруженных операторов. Приложение должно включать две формы и осуществлять взаимодействие одной формы с другой путем передачи между ними значений. Продемонстрировать работу с окном сообщений <i>MessageBox</i>. Используя свойство <i>Controls</i>, определяемое классом <i>Control</i>, реализовать изменение свойств однотипных элементов управления формы в цикл <i>foreach</i>.</p>															
5	<p>Лабораторная работа №5. Делегаты. Безопасность и обработка исключений. Построение графиков функций</p>	<p>Создать приложение позволяющее решать методом Ньютона и методом половинного деления нелинейные уравнения (согласно варианту), а также выполнять графическое построение графиков этих уравнений с помощью элемента управления <i>Chart</i>. При выполнении задания обязательно пользоваться делегатами. Предусмотреть проверку корректности ввода исходных данных в приложение и обработку исключительных ситуаций. Создать в приложении справочную систему</p> <p>Вариант задания:</p> <table border="1" data-bbox="651 667 1465 763"> <tr> <th>Уравнения</th> </tr> <tr> <td>1) <math>x \cdot \lg(x + 3) = 1</math></td> </tr> <tr> <td>2) <math>x^3 + x^2 + 2x + 3 = 0</math></td> </tr> </table>	Уравнения	1) $x \cdot \lg(x + 3) = 1$	2) $x^3 + x^2 + 2x + 3 = 0$												
Уравнения																	
1) $x \cdot \lg(x + 3) = 1$																	
2) $x^3 + x^2 + 2x + 3 = 0$																	
6	<p>Лабораторная работа №6. Методы класса <i>Graphics</i>. Элемент управления <i>DataGridView</i></p>	<p>Разработать приложение для построения диаграммы по введенным данным. Данные должны вводиться в <i>DataGridView</i>, диаграмма должна рисоваться без применения элемента управления <i>Chart</i>. Вид диаграммы определяется в соответствии с вариантом.</p> <p>Вариант задания:</p> <table border="1" data-bbox="624 920 1481 1173"> <thead> <tr> <th>Размерность таблицы</th> <th>Наименование диаграммы</th> <th>Вид диаграммы</th> </tr> </thead> <tbody> <tr> <td>3x3</td> <td>Пузырьковая диаграмма. Отображает на плоскости наборы из трех значений. Подобна точечной диаграмме, но третья величина отображает размер пузырька.</td> <td></td> </tr> </tbody> </table>	Размерность таблицы	Наименование диаграммы	Вид диаграммы	3x3	Пузырьковая диаграмма. Отображает на плоскости наборы из трех значений. Подобна точечной диаграмме, но третья величина отображает размер пузырька.										
Размерность таблицы	Наименование диаграммы	Вид диаграммы															
3x3	Пузырьковая диаграмма. Отображает на плоскости наборы из трех значений. Подобна точечной диаграмме, но третья величина отображает размер пузырька.																
7	<p>Лабораторная работа №7. Массивы и коллекции <i>C#</i></p>	<p>Создать коллекцию <i>ArrayList</i> для хранения экземпляров класса, созданного при выполнении лабораторной работы №3. Разработать приложение с элементом управления <i>MenuStrip</i>. Главное меню должно включать пункты и подпункты, представленные на рис.</p> <div data-bbox="667 1308 1449 1621" style="text-align: center;"> <table border="1"> <tr> <td>МАССИВ</td> <td>РАБОТА</td> <td>О ПРОГРАММЕ</td> </tr> </table>   <table border="1"> <tr> <td>Ввод Автомат</td> <td>Создать Просмотреть Редактировать  Сохранить Загрузить Выход</td> <td>Сортировка Поиск</td> <td>Линейный Бинарный</td> </tr> </table> </div> <p>Реализовать все указанные в меню действия с массивами. Для сортировки массива и поиска элемента использовать методы класса <i>ArrayList</i>. Для манипуляции содержимым массива обязательно использовать двустороннюю связь с <i>DataGridView</i>. Для реализации выполнения команд: Создать, Просмотреть и Редактировать использовать одну форму, динамически изменяя свойства элементов управления.</p> <p>Команды пункта меню Работа должны реализовывать способ сортировки коллекции, поиск элемента и выполнение индивидуального задания в соответствии с вариантом задания</p> <p>Вариант задания:</p> <table border="1" data-bbox="624 1935 1433 2116"> <thead> <tr> <th>Имя класса</th> <th>Способ сортировки коллекции</th> <th>Элемент для поиска</th> <th>Задание</th> </tr> </thead> <tbody> <tr> <td>Класс <i>Rectangle</i></td> <td>По возрастанию периметров</td> <td>Поиск трех прямоугольников с</td> <td>Подсчитать количество квадратов и</td> </tr> </tbody> </table>	МАССИВ	РАБОТА	О ПРОГРАММЕ	Ввод Автомат	Создать Просмотреть Редактировать  Сохранить Загрузить Выход	Сортировка Поиск	Линейный Бинарный	Имя класса	Способ сортировки коллекции	Элемент для поиска	Задание	Класс <i>Rectangle</i>	По возрастанию периметров	Поиск трех прямоугольников с	Подсчитать количество квадратов и
МАССИВ	РАБОТА	О ПРОГРАММЕ															
Ввод Автомат	Создать Просмотреть Редактировать  Сохранить Загрузить Выход	Сортировка Поиск	Линейный Бинарный														
Имя класса	Способ сортировки коллекции	Элемент для поиска	Задание														
Класс <i>Rectangle</i>	По возрастанию периметров	Поиск трех прямоугольников с	Подсчитать количество квадратов и														



			прямоугольником	максимальным периметром	вычислить для них средние значения площади, периметра и длины стороны		
8	Лабораторная работа №8,9. Иерархии классов, элемент управления TreeView и технология Drag-and-Drop	<p>А) Требования к разработке классов (Лабораторная работа № 8)</p> <ol style="list-style-type: none"> <li>Иерархия классов должна состоять из двух-трех уровней.</li> <li>В классах описать: поля (обязательно должны быть поля с числовыми значениями), свойства, конструкторы, минимум 2 метода (вычисляющие значение на основе значений числовых полей) (<i>методы придумать самим, исходя из особенностей предметной области</i>), перегрузку интерфейса IComparable&lt;T&gt;, минимум 2 перегрузки операций, перегрузку констант true и false, перегрузки implicit и explicit. Один из методов в классе верхнего уровня должен быть абстрактным (или виртуальным), и все классы нижнего уровня должны переопределять этот метод, предоставляя разную его реализацию.</li> <li>Для хранения в памяти ПК экземпляров созданных классов использовать обобщенные коллекции LIST&lt;T&gt;.</li> <li>Для коллекций экземпляров созданных классов реализовать методы, например: <ul style="list-style-type: none"> <li>поиск объекта с заданными характеристиками;</li> <li>сортировка объектов по некоторому признаку;</li> <li>подсчет статистики встречаемости объектов с определенными характеристиками;</li> <li>сортировка объектов;</li> <li>«фильтрация» объектов</li> </ul> и т.д.</li> </ol> <p>Б) Требования к разработке приложения (Лабораторная работа № 9)</p> <ol style="list-style-type: none"> <li>Приложение должно реализовывать возможности: <ul style="list-style-type: none"> <li>создания/удаления объектов построенной иерархии классов;</li> <li>изменения характеристик объектов;</li> <li>визуализации объектов (для нее разработать интерфейс с применением TreeView (например, как в примере с TreeView выше));</li> <li>манипулирования объектами на форме приложения с помощью «мышки»;</li> <li>сохранения/считывания текущего состояния иерархии объектов в формате XML (сериализация).</li> <li>демонстрации работы реализованных методов</li> </ul> </li> <li>Приложение должно отвечать требованиям Windows-приложений: <ul style="list-style-type: none"> <li>система меню (главное, контекстное), панель инструментов, горячие клавиши</li> <li>система подсказок (всплывающие, F1, пункт меню Справка, строка состояний)</li> <li>продуманная последовательность переключения фокуса управляющих элементов формы</li> <li>обработка событий клавиатуры и мыши (реализовать <i>Drag-and-Drop</i> (например, создать картинку корзины и при перетаскивании туда узла он бы удалялся))</li> <li>обработка исключительных ситуаций</li> </ul> </li> </ol> <p>Вариант задания:</p> <ol style="list-style-type: none"> <li>Иерархия объектов, содержащих информацию об основных изделиях промышленности строительных материалов</li> <li>Иерархия объектов вооружений</li> </ol>					
9	Лабораторная работа № 10 Пользовательские элементы управления и компоненты. Разработка простых и составных элементов управления.	<p>В соответствии с вариантом задания разработать простой и составной элементы управления (user control). Разместить значки разработанных элементов у правления в Панели элементов. Создать приложение, добавить в него разработанные пользовательские элементы управления и продемонстрировать их работу.</p> <table border="1" data-bbox="635 2042 1476 2107"> <tr> <td>Простой элемент управления</td> <td>Составной элемент управления</td> </tr> </table>				Простой элемент управления	Составной элемент управления
Простой элемент управления	Составной элемент управления						

		«Убегающий» ЭУ при наступлении событий	ЭУ, реализующий 4 действия с целыми числами с обработкой искл. ситуации
10	Лабораторная работа № 11 Документирование кода с помощью XML-комментариев	Для класса, разработанного в лабораторной работе №3 или для иерархии классов, разработанной в лабораторной работе №8 (на ваш выбор) добавить XML- комментарии и сгенерировать XML- документацию.	
11	Лабораторная работа № 12 Разработка тестов. Среда тестирования JUnit. Самотестирующийся код	Для класса, разработанного в лабораторной работе №3 создать проект C#, создать тесты для проверки его кода, выполнить запуск тестов и изучить результаты. После этого выполнить изменение кода проекта и повторный запуск тестов.	

В процессе демонстрации результатов студенту может быть предложено ответить на несколько вопросов, связанных с тематикой работы. Полные перечни контрольных вопросов приведены в методических указаниях. Примерный перечень вопросов приведен в следующей таблице.

№	Тема лабораторной работы	Контрольные вопросы
1	Лабораторная работа №1. Интегрированная среда разработки Visual Studio. Основы языка программирования C#	<ul style="list-style-type: none"> <li>- Основные компоненты платформы <i>.NET Framework</i>: общезыковая среда выполнения (CLR) и библиотека классов.</li> <li>- Этапы создания приложения в <i>.NET Framework</i>.</li> <li>- Архитектура платформы <i>.NET Framework</i>.</li> <li>- Элементы окна интегрированной среды <i>Visual Studio C#</i>.</li> <li>- Пространство имен платформы <i>.NET Framework</i>. Ускоренные методы доступа к методам классов.</li> <li>- Структура проекта консольного приложения.</li> <li>- Структура проекта <i>Windows Forms</i> приложения.</li> <li>- Структура программы на языке C#.</li> <li>- Система общих типов.</li> <li>- Наиболее востребованные встроенные типы.</li> <li>- Оператор языка C# <code>foreach</code>.</li> <li>- Консольный ввод/вывод.</li> <li>- Область действия и время существования переменных. Параметры методов.</li> <li>- Преобразования типов данных.</li> <li>- Операции упаковки и распаковки.</li> </ul>
2	Лабораторная работа №2. Система типов языка программирования C#. Структуры. Массивы. Сложность алгоритмов	<ul style="list-style-type: none"> <li>- Система общих типов, значимые и ссылочные типы.</li> <li>- Массивы в C#.</li> <li>- Структуры в C#.</li> <li>- Наиболее востребованные встроенные типы.</li> <li>- Оператор языка C# <code>foreach</code>.</li> <li>- Консольный ввод/вывод.</li> <li>- Файловый ввод/вывод.</li> <li>- Область действия и время существования переменных.</li> <li>- Параметры методов.</li> <li>- Особенности передачи значений через параметры при вызове методов.</li> <li>- Преобразования типов данных.</li> <li>- Операции упаковки и распаковки.</li> <li>- Сложность алгоритма.</li> </ul>
3	Лабораторная работа №3. Основы объектно-ориентированного программирования	<ul style="list-style-type: none"> <li>- Понятие ООП и основные принципы ООП</li> <li>- Понятие класса и объекта</li> <li>- Члены класса, их назначение</li> <li>- Полиморфизм</li> <li>- Наследование</li> <li>- Инкапсуляция</li> <li>- Описание классов в C#</li> <li>- Модификаторы доступа классов</li> <li>- Модификаторы доступа членов класса</li> <li>- Перегрузка методов</li> <li>- Перегрузка операций</li> <li>- Основные принципы ООП</li> <li>- Статические классы и поля</li> </ul>

№	Тема лабораторной работы	Контрольные вопросы
		<ul style="list-style-type: none"> <li>- Автореализуемое свойство</li> <li>- Параметры методов</li> </ul>
4	Лабораторная работа №4. Технология Windows Forms. Элементы управления среды Visual Studio C#	<ul style="list-style-type: none"> <li>- Технология <i>Windows Forms</i></li> <li>- Класс <i>Control</i> его свойства и события</li> <li>- Свойство <i>Controls</i> класса <i>Control</i></li> <li>- Общие для всех элементов управления свойства</li> <li>- Статические элементы управления</li> <li>- Кнопки и переключатели</li> <li>- Горизонтальная и вертикальная полосы прокрутки</li> <li>- Элементы управления с поддержкой редактирования текста</li> <li>- Список и поле со списком</li> <li>- Наборные счётчики</li> <li>- Древовидное и списковое представление</li> <li>- Разделители</li> <li>- Меню и панели инструментов</li> <li>- Диалоговые окна</li> <li>- Класс <i>MessageBox</i></li> <li>- Модальные и немодальные окна и формы</li> <li>- Организация взаимодействия одной формы с другой</li> </ul>
5	Лабораторная работа №5. Делегаты. Безопасность и обработка исключений. Построение графиков функций	<ul style="list-style-type: none"> <li>- Понятие делегата</li> <li>- Описание и создание делегата</li> <li>- Порядок работы с делегатом</li> <li>- В каких случаях используют делегаты</li> <li>- Назначение элемента управления <i>Chart</i> и его наиболее важные свойства</li> <li>- Типы диаграмм для представления данных класса <i>Chart</i></li> <li>- Элементы управления для создания справочной поддержки приложения</li> <li>- Добавление и настройка всплывающих подсказок к элементам управления приложения</li> <li>- Добавление и настройка строки состояния для формы</li> <li>- Добавление и настройка окна справки «О программе»</li> <li>- Свойства формы для настройки строки-заголовка</li> <li>- Безопасность и обработка исключений</li> <li>- Генерация исключений</li> <li>- Синтаксис операторов <i>try</i>, <i>catch</i> и <i>finally</i></li> </ul>
6	Лабораторная работа №6. Методы класса <i>Graphics</i> . Элемент управления <i>DataGridView</i>	<ul style="list-style-type: none"> <li>- Интерфейс графических устройств <i>GDI</i></li> <li>- Методы класса <i>Graphics</i></li> <li>- Система координат окна, для которого получен объект <i>Graphics</i></li> <li>- Варианты метода <i>DrawLine</i> для рисования одной и нескольких линий</li> <li>- Варианты метода <i>DrawRectangles</i> для рисования одного и нескольких прямоугольников</li> <li>- События <i>MouseUp</i>, <i>MouseDown</i>, <i>MouseMove</i></li> <li>- Использование объекта <i>Bitmap</i></li> <li>- Использование объекта <i>PictureBox</i></li> <li>- Назначение и свойства элемента управления <i>DataGridView</i></li> <li>- Настройка несвязанного <i>DataGridView</i></li> <li>- Настройка <i>DataGridView</i>, связанного с источником данных</li> <li>- Допустимые источники данных для <i>DataGridView</i></li> <li>- Обеспечение двустороннего механизма связи между <i>DataGridView</i> и коллекцией</li> </ul>
7	Лабораторная работа №7. Массивы и коллекции C#	<ul style="list-style-type: none"> <li>- Понятие и назначение интерфейса</li> <li>- Объявление интерфейса в C#</li> <li>- Особенности реализации интерфейса в классах</li> <li>- Понятие и назначение коллекций в C#</li> <li>- Типы коллекций в C#</li> <li>- Сравнение обобщенных и необобщенных коллекций</li> <li>- Классы обобщенных коллекций</li> <li>- Интерфейсы, представляющие базовую функциональность коллекций в пространстве имен <i>System.Collections</i></li> </ul>

№	Тема лабораторной работы	Контрольные вопросы
		<ul style="list-style-type: none"> <li>- Понятие и назначение коллекции <i>ArrayList</i></li> <li>- Интерфейсы, реализованные в классе <i>System.Array</i></li> <li>- Примеры методов класса <i>System.Array</i></li> <li>- Порядок работы с коллекцией <i>ArrayList</i></li> <li>- Особенности сортировки и поиска элементов в коллекции <i>ArrayList</i></li> <li>- Порядок создания главного меню в приложении</li> <li>- Порядок создания контекстного меню в приложении</li> </ul>
8	Лабораторная работа №8,9. Иерархии классов, элемент управления <i>TreeView</i> и технология <i>Drag-and-Drop</i>	<ul style="list-style-type: none"> <li>- Наследование и реализация наследования в C#.</li> <li>- Управление доступностью классов и его членов при наследовании.</li> <li>- Абстрактные классы.</li> <li>- Классы, закрытые для наследования.</li> <li>- Статические классы.</li> <li>- Полиморфизм и реализация полиморфизма.</li> <li>- Абстрактные и виртуальные методы.</li> <li>- Перегрузка методов.</li> <li>- Построение схемы классов в среде <i>Visual Studio C#</i>.</li> <li>- Элементы управления для древовидного и спискового представления данных</li> <li>- Создание объекта класса <i>TreeNode</i> и получение доступа к узлам класса <i>TreeNode</i></li> <li>- Методы развёртывания/свёртывания узлов.</li> <li>- Управление видом элемента управления <i>TreeView</i></li> <li>- Назначение и порядок реализации механизма <i>Drag-and-Drop</i></li> <li>- Назначение и порядок реализации сериализации/ десериализации.</li> </ul>
9	Лабораторная работа № 10 Пользовательские элементы управления и компоненты. Разработка простых и составных элементов управления.	<ul style="list-style-type: none"> <li>- типами создаваемого программного продукта</li> <li>- Компонент</li> <li>- Пользовательский компонент</li> <li>- <u>Этапы разработки компонентов</u></li> <li>- <u>виды компонент</u></li> <li>- невидимые компоненты</li> <li>- специальные элементы управления</li> <li>- <u>Список базовых классов среды .NET Framework</u></li> </ul>
10	Лабораторная работа № 11 Документирование кода с помощью XML-комментариев	<ul style="list-style-type: none"> <li>- Пользовательская документация</li> <li>- Техническая документация</li> <li>- использованием XML документации</li> <li>- теги для комментариев документации (&lt;summary&gt;, &lt;remarks&gt;, &lt;returns&gt;, &lt;value&gt;, &lt;example&gt;, и другие</li> </ul>
11	Лабораторная работа № 12 Разработка тестов. Среда тестирования JUnit. Самотестирующийся код	<ul style="list-style-type: none"> <li>- Понятие модульного теста</li> <li>- Платформа модульных тестов Майкрософт</li> <li>- Управляемый код</li> <li>- Создание тестов для проверки кода</li> <li>- Создание метода теста</li> <li>- Требования к методу теста</li> <li>- Запуск тестов</li> <li>- Анализ результатов тестирования</li> </ul>

Процедура "выполнения" лабораторных работ представляет собой качественную оценку знаний, умений и навыков студентов.

Количественная оценка предусматривается в процессе "защиты" лабораторных работ. Такая оценка производится на основании результатов, полученных в ходе выполнения заданий.

Для защиты лабораторных работ предусмотрены задания, в каждом из которых по 5 вопросов соответствующей тематики. Каждое задание необходимо выполнить на положительную оценку, ответив минимум на 3 вопроса из пяти предложенных. Студент, выполнивший и защитивший все лабораторные работы в семестре, получает "зачет".

## Ниже приведены образцы заданий для «защиты» лабораторных работ

### Вариант 1

1. Как называются методы, предназначенные для создания объектов?
2. Какие файлы создаются при создании проекта?
3. Основные отличия ООП и процедурного программирования?
4. Что такое класс?
5. Что такое поле

### Вариант 2

1. Опишите принцип работы конструктора объектов.
2. Какой метод используется для уничтожения объектов?
3. Какие виды доступа к полям, методам и свойствам вам известны
4. Дайте определение свойству.
5. Для чего предназначено окно Object Tree View?

### Вариант 3

1. Что такое метод
2. Для чего предназначена вкладка **Properties** окна **Object Inspector**?
3. Что является основой программной среды Visual Studio?
4. Назовите основные окна, которые включает в себя Visual Studio.
5. Из чего состоит Главное окно? Для чего оно предназначено?

### Вариант 4

1. Дайте определение Событию. Приведите пример события на языке C#
2. Что понимается под объектно-ориентированным программированием?
3. На взаимодействии каких процессов строится программа в Delphi?
4. В каком разделе описываются объекты(экземпляры) класса?
5. Приведите пример описания класса на Delphi.

### Вариант 5

1. Что такое инкапсуляция?
2. Основные отличия ООП и процедурного программирования?
3. Опишите принцип работы конструктора объектов.
4. Какой класс называется классом-потомком или дочерним классом?
5. Что такое виртуальные и динамические методы?

### Вариант 6

1. Расскажите об использовании Элемента управления Time. Приведите пример.
2. Какие классы общего назначения позволяют выводить графические изображения на поверхность компонентов?
3. Область видимости элементов какой секции повысить нельзя?
4. В чём состоит отличие между областями видимости public и published?
5. Приведите пример использования полиморфизма в самой среде Delphi.

### Вариант 7

1. На что указывает директива override?
2. Приведите пример описания динамического массива.
3. Что происходит при нажатии произвольной клавиши?
4. Какие вещественные типы данных вы знаете. Укажите диапазон и кол-во байт.
5. Какие события входят в набор событий клавиатуры?

### Вариант 8

1. Что называется делегированием?

2. Опишите виртуальные методы связи.
3. Что называется событием в Object Pascal?
4. Какой параметр является обязательным для всех событий?
5. Какой метод даёт возможность сделать форму невидимой?

### Вариант 9

1. Какие целочисленные типы данных вы знаете. Укажите диапазон и кол-во байт.
2. Какие типы используются для обработки текстов в Object Pascal?
3. Расскажите о типе String.
4. Какие логические типы данных вы знаете.
5. С помощью какого оператора происходит инициализация массива?

Предусмотрены текущие тесты для защиты лабораторных работ. Для защиты каждой из лабораторных работ необходимо пройти на положительную оценку "текущий" тест соответствующей тематики. Тестирование проводится в электронной форме (на базе сервера VeralTest или Online Test Pad). Тесты представляют собой наборы заданий (вопросов) следующих типов: «Единичный выбор ответа», «Множественный выбор ответа». Ввод или выбор правильного ответа в каждом задании оценивается 1 (одним) баллом. Ввод или выбор неправильного ответа в каждом задании оценивается 0 (нулем) баллов. Каждый верный вариант ответа в вопросе с «Множественным выбором ответа» оценивается 1 (одним) баллом. Таким образом, в каждом из заданий типа «Множественный выбор ответа» можно набрать более 1 (одного) балла. Для вопросов с «Множественным выбором ответа» выбор хотя бы одного ошибочного ответа обнуляет количество баллов, набранных в задании. Процент набранных баллов определяется как доля набранных баллов от максимального количества баллов, содержащихся в тесте. Вопросы теста при каждом прохождении выбираются случайным образом из общего банка заданий.

### 5.4. Описание критериев оценивания компетенций и шкалы оценивания

При промежуточной аттестации в форме дифференцированного зачета используется следующая шкала оценивания: 2 – неудовлетворительно, 3 – удовлетворительно, 4 – хорошо, 5 – отлично.

Критериями оценивания достижений показателей являются:

Наименование показателя оценивания результата обучения по дисциплине	Критерий оценивания
Знание общих характеристик современных информационных технологий и программных средств; этапов разработки ПС при решении задач профессиональной деятельности; основных процессов программной инженерии и вопросов их выполнения; методов и инструментов	Знание терминов, определений, понятий: исчерпывающе, последовательно, четко и логически стройно излагает материал; самостоятельно формулирует, анализирует и сравнивает условия применимости, преимущества и недостатки описываемых технологий; самостоятельно может изложить методы решения задач по изученным разделам; знает: Понятие делегата. Описание и создание делегата. Порядок работы с делегатом. Безопасность и обработка исключений. Генерация исключений. Понятие и назначение интерфейса. Абстрактные и виртуальные методы. Понятие и назначение коллекций в C#. Типы коллекций в C#. Сравнение обобщенных и необобщенных коллекций. Классы обобщенных коллекций. Наследование и реализация наследования в C#. Управление доступностью классов и его членов при наследовании. Абстрактные классы. Классы, закрытые для наследования. Статические классы.
	Знание основных закономерностей, соотношений, принципов
	Объем освоенного материала
	Полнота ответов на вопросы
	Четкость изложения и интерпретации знаний

программной инженерии; приемов и инструментов тестирования.	
Умение применять современные информационные технологии и программные средства (.NET Framework, Visual Studio C#) при решении задач профессиональной деятельности.	Умение использовать теоретические знания для выбора методики решения задач, выполнения заданий: использует инструментарий; самостоятельно может предложить и разработать предложения по обоснованному выбору варианта решения задачи; умеет создавать и использовать делегаты; организовывать в приложении проверку корректности ввода исходных данных; грамотно обрабатывать стандартные исключительные ситуации и генерировать собственные; создавать собственные интерфейсы
	Умение проверять решение и анализировать результаты
	Умение качественно оформлять (презентовать) решение задач и выполнения заданий
Владение навыками применения современных информационных технологий и программных средств (.NET Framework, Visual Studio C#), при решении задач профессиональной деятельности; разработки консольных приложений и приложений с визуальным интерфейсом.	Навыки решения стандартных/нестандартных задач: может сформулировать модель для решения задач по изученным разделам и предложить метод ее решения; владеет навыками разработки приложений, поддерживающих безопасность и обработку исключений. Навыками генерации исключений. Навыками реализации в классах интерфейсов. Навыками разработки приложений, позволяющих работать с элементами коллекций. Навыками управления доступностью классов и его членов при наследовании. Владеет навыками разработки интерфейса приложения на основе анализа поставленной задачи
	Объем выполненных заданий
	Качество выполнения трудовых действий
	Самостоятельность планирования выполнения трудовых действий
Знание методов алгоритмизации, основ объектно-ориентированного программирования; основных типов данных языка программирования C# и основные приемы организации работ с ними; библиотеки классов BCL платформы .NET Framework; элементов управления среды Visual Studio C#.	Знание терминов, определений, понятий: Перегрузка методов. Добавление и настройка всплывающих подсказок к элементам управления приложения. Добавление и настройка строки состояния для формы. Добавление и настройка окна справки «О программе». Свойства формы для настройки строки-заголовка. Технология Windows Forms. Общие для всех элементов управления свойства. Статические элементы управления. Понятие ООП. Понятие класса и объекта. Члены класса, их назначение. Структура проекта консольного приложения. Структура проекта Windows Forms приложения. Структура программы на языке C#. Система общих типов
	Знание основных закономерностей, соотношений, принципов
	Объем освоенного материала
	Полнота ответов на вопросы
	Четкость изложения и интерпретации знаний
Умение применять принципы объектно-ориентированного программирования при решении профессиональных задач в области информационных систем и технологий; проводить выбор исходных данных для проектирования; описывать собственные классы и использовать классы библиотеки классов BCL платформы .NET Framework; делать	Освоение методик - умение решать практические задачи, выполнять типовые задания: Использовать при решении задач обобщенные и необобщенные коллекции Создавать классы разных видов (абстрактные, закрытые, статические) Сделать правильный выбор типов данных на основе анализа поставленной задачи
	Умение использовать теоретические знания для выбора методики решения задач, выполнения заданий
	Умение проверять решение и анализировать результаты
	Умение качественно оформлять (презентовать) решение задач и выполнения заданий

выбор среди элементов управления среды Visual Studio C# для разработки визуального интерфейса приложения.	
Владение навыками использования технологии объектно-ориентированного программирования и визуального программирования интерфейса программных средств; приемами и инструментами отладки и тестирования ПС.	Навыки решения стандартных/нестандартных задач: владеет Навыками разработки приложений, позволяющих работать с элементами коллекций. Навыками управления доступностью классов и его членов при наследовании. Владеет навыками разработки интерфейса приложения на основе анализа поставленной задачи. Приемами работы со строкой состояния, статическими элементами управления, всплывающими подсказками, главным и контекстным меню. Навыками работы в среде ООП. Навыками создания и описания классов
	Объём выполненных заданий
	Качество выполнения трудовых действий
	Самостоятельность планирования выполнения трудовых действий

Оценка преподавателем выставляется интегрально с учётом всех показателей и критериев оценивания.

Оценка сформированности компетенций по показателю Знания.

Критерий	Уровень освоения и оценка			
	2	3	4	5
Знание терминов, определений, понятий	Не знает терминов и определений	Знает термины и определения, но допускает неточности формулировок	Знает термины и определения	Знает термины и определения, может корректно сформулировать их самостоятельно
Знание основных закономерностей, соотношений, принципов	Не знает основные закономерности и соотношения, принципы построения знаний	Знает основные закономерности, соотношения, принципы построения знаний	Знает основные закономерности, соотношения, принципы построения знаний, их интерпретирует и использует	Знает основные закономерности, соотношения, принципы построения знаний, может самостоятельно их получить и использовать
Объём освоенного материала	Не знает значительной части материала дисциплины	Знает только основной материал дисциплины, не усвоил его деталей	Знает материал дисциплины в достаточном объеме	Обладает твердым и полным знанием материала дисциплины, владеет дополнительными знаниями
Полнота ответов на вопросы	Не дает ответы на большинство вопросов	Дает неполные ответы на все вопросы	Дает ответы на вопросы, но не все - полные	Дает полные, развернутые ответы на поставленные вопросы
Четкость изложения и интерпретации знаний	Излагает знания без логической последовательности	Излагает знания с нарушениями в логической последовательности	Излагает знания без нарушений в логической последовательности	Излагает знания в логической последовательности, самостоятельно их интерпретируя и анализируя
	Не иллюстрирует изложение поясняющими схемами,	Выполняет поясняющие схемы и рисунки небрежно и с	Выполняет поясняющие рисунки и схемы корректно и	Выполняет поясняющие рисунки и схемы точно и аккуратно, раскрывая



	рисунками и примерами	ошибками	понятно	полноту усвоенных знаний
	Неверно излагает и интерпретирует знания	Допускает неточности в изложении и интерпретации знаний	Грамотно и по существу излагает знания	Грамотно и точно излагает знания, делает самостоятельные выводы

### Оценка сформированности компетенций по показателю Умения.

Критерий	Уровень освоения и оценка			
	2	3	4	5
Освоение методик - умение решать практические задачи, выполнять типовые задания	Не умеет решать практические задачи, выполнять типовые задания	С дополнительной помощью может решать практические задачи, выполнять типовые задания, допускает ошибки	Допускает неточности при решении практических задач и выполнении типовых заданий	Грамотно использует методики, умеет решать все практические задачи, выполнять все типовые задания
Умение использовать теоретические знания для выбора методики решения задач, выполнения заданий	Не умеет использовать теоретические знания для выбора методики решения задач, выполнения заданий	С дополнительной помощью может выполнить выбор методики решения задач. При выполнении заданий допускает ошибки	Умеет использовать теоретические знания для выбора методики решения задач, допускает неточности при выполнении заданий	Самостоятельно может сделать выбора методики решения задач, выполняет все задания без ошибок
Умение проверять решение и анализировать результаты	Не умеет проверять решение и анализировать результаты	Проверяет решение, с дополнительной помощью может анализировать результаты	Проверяет решение в достаточном объеме, при анализе результатов допускает неточности	Обладает твердыми умениями проверки решения и анализа результатов
Умение качественно оформлять (презентовать) решение задач и выполнения заданий	Не умеет качественно оформлять (презентовать) решение задач и выполнения заданий	Выполняет поясняющие схемы и рисунки небрежно и с ошибками	Выполняет оформление решения задач и выполнения заданий корректно и понятно	Качественно и на высоком уровне оформляет решение задач и выполнения заданий

### Оценка сформированности компетенций по показателю Иметь навыки.

Критерий	Уровень освоения и оценка			
	2	3	4	5
Навыки решения стандартных/нестандартных задач	Не может выполнять решения стандартных задач	С дополнительной помощью может выполнить решения стандартных/нестандартных задач, допускает ошибки	Может выполнить решение стандартных/нестандартных задач, но допускает неточности	Самостоятельно может выполнить решение стандартных/нестандартных задач
Объем выполненных заданий	Не выполняет значительную часть заданий по дисциплине	Выполняет задания только по основному материалу дисциплины, не усвоил его деталей	Выполняет задания в достаточном объеме	Выполняет весь объем заданий. Обладает твердым и полным знанием материала дисциплины, владеет

				дополнительными знаниями
Качество выполнения трудовых действий	Не выполняет трудовые действия	Имеет навыки выполнения трудовых действий только по основному материалу дисциплины, не усвоил его деталей	Имеет навыки выполнения трудовых действий в достаточном объеме	Обладает твердыми навыками выполнения трудовых действий по всему материалу дисциплины, владеет дополнительными навыками
Самостоятельность планирования выполнения трудовых действий	Не выполняет планирования выполнения трудовых действий	Допускает неточности при планировании выполнения трудовых действий	Самостоятельно и грамотно выполняет планирование выполнения большинства трудовых действий	Самостоятельно и грамотно выполняет планирование выполнения всех трудовых действий

## 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

### 6.1. Материально-техническое обеспечение

№	Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы
1	Аудитория для лекционных занятий	оборудованы специализированной мебелью, мобильным или стационарным мультимедийным проектором, переносным экраном, ноутбуком, или компьютером на базе одно или двухъядерных процессоров с тактовой частотой не менее 2 ГГц, объемом оперативной памяти не менее 2 Гб и жесткого диска до 500 Гб; локальная сеть с пропускной способностью 100 Мбит/с
2	Компьютерные классы для проведения лабораторных занятий	оборудованы специализированной мебелью, компьютерами с установленными программными продуктами на базе одно или двухъядерных процессоров с тактовой частотой не менее 2 ГГц, объемом оперативной памяти не менее 2 Гб и жесткого диска до 500 Гб; локальная сеть с пропускной способностью 100 Мбит/с, принтеры или многофункциональные устройства форматов А4, А3.
3	Помещения для самостоятельной работы обучающихся	оборудованы специализированной мебелью, компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации

### 6.2. Лицензионное и свободно распространяемое программное обеспечение

№	Перечень лицензионного программного обеспечения.	Реквизиты подтверждающего документа
1	Microsoft Office Professional Plus 2016	Соглашение Microsoft Open Value Subscription V6328633. Соглашение действительно с 02.10.2017 по 31.10.2023
2	Microsoft Windows 10 Корпоративная	Соглашение Microsoft Open Value Subscription V6328633. Соглашение действительно с 02.10.2017 по 31.10.2023). Договор поставки ПО 0326100004117000038-0003147-01 от 06.10.2017
3	Kaspersky Endpoint Security «Стандартный Russian Edition»	Сублицензионный договор № 102 от 24.05.2018. Срок действия лицензии до 19.08.2020 Гражданско-правовой Договор (Контракт) № 27782 «Поставка продления права пользования (лицензии) Kaspersky Endpoint Security от 03.06.2020. Срок действия лицензии 19.08.2022г.
4	Google Chrome	Свободно распространяемое ПО согласно условиям лицензионного соглашения
5	Mozilla Firefox	Свободно распространяемое ПО согласно условиям лицензионного соглашения
6	Microsoft Visual Studio 2013	договор №63-14к от 02.07.2014
7	Система компьютерного тестирования	Свободно распространяемая

### 6.3. Перечень учебных изданий и учебно-методических материалов

1. Биллиг, В. А. Основы объектного программирования на C# (C# 3.0, Visual Studio 2008) [Электронный ресурс] : учебное пособие / Биллиг В. А. - Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. - 583 с. <http://www.iprbookshop.ru/72339.html?replacement=1>
2. Лазебная Е. А. Методические указания к выполнению курсовых работ по дисциплинам учебного плана направления бакалавриата 09.03.02 – Информационные системы и технологии для студентов I-IV курсов очной и заочной форм обучения и правила оформления расчетно-пояснительных записок [Электронный ресурс] / сост. Е. А. Лазебная. - Электрон. текстовые дан. - Белгород : Издательство БГТУ им. В. Г. Шухова, 2017. Режим доступа : <https://elib.bstu.ru/Reader/Book/2017061914021833000000656486>
3. Агапов, В. П. Основы программирования на языке C# [Электронный ресурс] : учебное пособие / Агапов В. П. - Москва : Московский государственный строительный университет, ЭБС АСВ, 2012. - 128 с. <http://www.iprbookshop.ru/16366>
4. Хорев, П. Б. Технологии объектно-ориентированного программирования : учеб. пособие / П. Б. Хорев. - 2-е изд., стер. - Москва : Академия, 2008. - 448 с.
5. Лазебная Е. А. Визуальное программирование : методические указания к выполнению лабораторных работ для студентов 2-го курса направления бакалавриата 09.03.02 - Информационные системы и технологии / БГТУ им. В. Г. Шухова, каф. информ. технологий ; сост.: Е. А. Лазебная, Р. У. Стативко. - Белгород : Издательство БГТУ им. В. Г. Шухова, 2015. - 72 с.
6. Кариев, Ч. А. Разработка Windows-приложений на основе Visual C# [Электронный ресурс] : учебное пособие / Кариев Ч. А. - Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. - 768 с. <http://www.iprbookshop.ru/16097>
7. 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 / Казанский А. А. - Москва : Московский государственный строительный университет, ЭБС АСВ, 2011. - 180 с. <http://www.iprbookshop.ru/19258>
8. Мейер, Б. Объектно-ориентированное конструирование программных систем / Б. Мейер. - Москва : Русская редакция, 2005. - 1198 с.
9. Иванова, Г. С. Объектно-ориентированное программирование : учебник / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев. - Изд. 2-е, перераб. и доп. - Москва : Издательство МГТУ им. Н. Э. Баумана, 2003. - 367 с.

### 6.4. Перечень интернет ресурсов, профессиональных баз данных, информационно-справочных систем

1. <http://it.bstu.ru> – Сайт кафедры информационных технологий БГТУ им. В.Г. Шухова
2. <http://ntb.bstu.ru>. - Официальный сайт научно-технической библиотеки БГТУ им. В.Г. Шухова
3. [www.n-t.ru](http://www.n-t.ru) – "Наука и техника" - электронная библиотека
4. [www.nature.ru](http://www.nature.ru) - "Научная сеть" - научно-образовательные ресурсы
5. [www.intuit.ru](http://www.intuit.ru) - "Интернет-университет информационных технологий"