

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Андрей Драгомирович Хлутков
Должность: директор
Дата подписания: 09.06.2026 20:14:41
Уникальный программный ключ:
880f7c07c583b07b775f6604a630281b13ca9fd2

Приложение 4
к образовательной программе

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.11 Программирование

(индекс, наименование дисциплины в соответствии с учебным планом)

38.03.05 Бизнес-информатика

(код, наименование направления подготовки/специальности)

Бизнес-аналитика

(наименование образовательной программы)

Очная форма обучения

(форма обучения)

Год набора - 2026

Санкт-Петербург

Автор(ы)-составитель(и) РПД:

Рассказов Владимир Александрович, кандидат технических наук, доцент кафедры бизнес-информатики

Заведующий кафедрой:

Наумов Владимир Николаевич, доктор военных наук, профессор, заведующий кафедрой бизнес-информатики

Рабочая программа дисциплины Б1.О.11 Программирование одобрена на заседании кафедры бизнес-информатики факультета экономики и финансов СЗИУ РАНХиГС.

протокол № 6 от «26» марта 2026 г.

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.
2. Объем и место дисциплины в структуре образовательной программы.
3. Содержание и структура дисциплины.
4. Типы оценочных материалов, показатели и критерии их оценивания.
5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам.
6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине.
7. Методические материалы по освоению дисциплины.
8. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет».
9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы.

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Дисциплина Б1.О.11 Программирование обеспечивает формирование у обучающихся следующих универсальных, общепрофессиональных и профессиональных компетенций:

ОТФ/ТФ и реквизиты ПС	Код компетенции	Наименование компетенции	Код индикатора достижения компетенции	Наименование индикатора достижения компетенций	Образовательный результат
	ОПК-3	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3.1	Разрабатывает алгоритмы и программы, проектирует базы данных с целью управления процессами создания продуктов и услуг ИКТ	<p>ОПК-3.1 З-1. Знает основные понятия и методы программирования, классификацию языков программирования, парадигмы программирования;</p> <p>ОПК-3.1 З-2. Знает основные положения теории баз данных и методы проектирования баз данных;</p> <p>ОПК-3.1 У-1. Умеет разрабатывать алгоритмы и программы;</p> <p>ОПК-3.1 У-2. Умеет проектировать базы данных;</p> <p>ОПК-3.1 У-3. Умеет разрабатывать программное обеспечение с использованием современных инструментальных средств программирования</p>

2. Объем и место дисциплины в структуре образовательной программы

Объем дисциплины

Общий объем дисциплины:

7,00 з.е., 252 ак.час

Контактная работа обучающихся с преподавателем по видам учебных

занятий: 111 ак.часов на контактную работу с преподавателем, из них 32 ак.часа на лекции и 60 ак.часа на практические занятия, 2 ак.часа на консультацию, 17 ак.часов на Каттэк. 114 ак.часов на самостоятельную работу обучающихся. 9 ак.часов на самостоятельную работу на подготовку курсовой работы, 18 ак.часов на самостоятельную работу на подготовку к экзамену.

Место дисциплины в структуре ОП ВО

Дисциплина Б1.О.11 «Программирование» входит реализуется в 1-м и 2-м семестрах 1-го курса. Дисциплина опирается на знание общеобразовательных дисциплин, в первую очередь на дисциплину Б1.О.10 «Основы информатики».

Дисциплина Б1.О.11 «Программирование» создаёт необходимые предпосылки для освоения программ таких дисциплин, как Б1.О.12 «Базы данных», Б1.О.14 «Объектно-ориентированный анализ и программирование» и ряда дисциплин по выбору студента.

3. Содержание и структура дисциплины

3.1. Структура дисциплины

Очная форма обучения

№ п/п	Наименование тем и (или) разделов	ВСЕГО	Объем дисциплины, ак.час										Форма текущего контроля успеваемости, промежуточной аттестации	
			Контактная работа обучающихся с преподавателем по видам учебных занятий							Самостоятельная работа				
			Период теоретического обучения				Период промежуточной аттестации (сессия)							
			Занятия лекционного типа		Занятия семинарского типа		ИК	КСР	КЭ	Кат тЭК	К о н т р о л ь	СРкр		СРэк
Л	ВЛ	ЛР	ПЗ											
Тема 1	Введение в теорию алгоритмов.	32	4			8							20	Тестирование
Тема 2	Основные конструкции языка программирования С#.	36	4			8							24	Тестирование
Тема 3	Основы объектно-ориентированного программирования на С#.	52	4			10							38	Контрольная работа

Тема 4	Концепция абстрактных типов данных. Структуры данных C#.	20	4			4							12	Тестирование
Промежуточная аттестация (1 семестр)		4							4					Зачет
<i>Итого 1 семестр</i>		<i>144</i>	<i>16</i>			<i>30</i>			<i>4</i>				<i>94</i>	
Тема 5	Основные принципы WPF. Общие сведения о элементах управления.	16	4			4							4	Тестирование
Тема 6	Создание приложений с графическим пользовательским интерфейсом на C#.	34	8			16							8	ПКЗ
Тема 7	Технологии ввода-вывода и хранения данных.	24	4			10							8	ПКЗ
Промежуточная аттестация (2 семестр)		15						2	13					Курсовой проект, Экзамен
<i>Итого 2 семестр</i>		<i>108</i>	<i>16</i>			<i>30</i>		<i>2</i>	<i>13</i>	<i>9</i>	<i>18</i>		<i>20</i>	
Итого		252	32			60		2	17	9	18		114	

Используемые сокращения:

Л – лекции - занятия, предусматривающие преимущественную передачу учебной информации обучающимся педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях).

ВЛ – видео лекции.

ЛР – лабораторные работы.

ПЗ – практические занятия (за исключением лабораторных работ).

ИК – индивидуальные консультации.

КСР – контроль самостоятельной работы

КЭ – консультации перед экзаменом

Каттэк – контактная работа на аттестацию в период экзаменационных сессий

Контроль - контактная работа на аттестацию в период экзаменационных сессий для заочной формы обучения

СРкр – самостоятельная работа на подготовку курсовой работы/ курсового проекта.

СРэк – самостоятельная работа на подготовку к экзамену.

СР – самостоятельная работа в семестре на подготовку к учебным занятиям.

Т – тестирование.

ПКЗ - практическое контрольное задание

3.2 Содержание дисциплины

Тема 1. Введение в теорию алгоритмов. ОПК-3.1

Введение в теорию алгоритмов. Алгоритмизация. Информационные процессы. Определение и свойства алгоритма. Вычислимые функции. Частично-рекурсивные функции. Способы записи и типы алгоритмов. Технические средства реализации информационных систем. Тенденции развития ЭВМ. Классификация ЭВМ. Программные средства реализации информационных процессов. Классификация программного обеспечения. Парадигмы программирования. Способы выполнения программы.

Вычислительная сложности алгоритмов. Асимптотический анализ трудоемкости алгоритмов. Оценка сложности алгоритмических конструкций. Типовые функции временной оценки сложности алгоритмов. Составные классы сложности.

Тема 2. Основные конструкции языка программирования C#. ОПК-3.1

Архитектура .NET. Обзор версий C# и .NET. Инструментальные средства разработки приложений на языке C#. Основы синтаксиса языка C#. Идентификаторы, ключевые слова и комментарии. Переменные и типы данных. Неявно типизированные переменные, значения типов по умолчанию. Преобразование типов. Массивы. Перечисления. Операторы, операции и выражения C#. Арифметические операции и операторы. Побитовые операции и операторы. Логические операции и операторы. Простейшие строковые операции и операторы. Null операции и операторы. Управление потоком выполнения программы. Операторы выбора. Операторы циклов и итераций. Операторы перехода.

Тема 3. - Основы объектно-ориентированного программирования на C#. ОПК-3.1

Основы объектно-ориентированного программирования на C#. Введение в классы, объекты. Присваивание переменных ссылочного типа.

Методы. Ключевое слово *this*. Перегрузка методов и операторов. Частичные классы и методы. Конструкторы. Перегрузка конструкторов. Инициализаторы объектов. Свойства. Индексаторы.

Наследование и полиморфизм. Инициализация производных классов. Виртуальные функции-члены. Абстрактные классы. Интерфейсы.

Класс *System.Object* и его методы. Метод *ToString()*. Метод *GetHashCode()*. Получение типа объекта и метод *GetType()*. Метод *Equals()*.

Обобщения. Обобщенные классы. Обобщенные методы

Тема 4. Концепция абстрактных типов данных. Структуры данных C#. ОПК-3.1

Концепция абстрактных типов данных. Абстрактный тип данных «Линейный список». Реализация на основе вектора. Реализация на основе связанного списка. Абстрактный тип данных «стек». Абстрактный тип данных «Очередь». Абстрактный тип данных «Дек».

Структуры данных C#. Интерфейсы перечислителя *IEnumerable* и *IEnumerator*. Интерфейсы *ICollection*, *IList* и *IDictionary*. Класс *Array*. Поиск, сортировка, копирование, преобразование. Списки, очереди, стеки и наборы. *List<T>* и *ArrayList*. *LinkedList<T>*. *Queue<T>* и *Queue*. *Stack<T>* и *Stack*. *BitArray*. *HashSet<T>* и *SortedSet<T>*. Словари.

LINQ to Objects. Использование лямбда-выражений, делегаты *Func*. Операции ограничения. Операция *Where*. Операции проекции *Select* и *SelectMany*. Операции разбиения *Skip* и *SkipWhile*, *Take* и *TakeWhile*. Операция конкатенации *Concat*. Операции упорядочивания *OrderBy*, и *OrderByDescending*. Интерфейс *IComparer<T>*. Операция соединения *Join*. Операции работы со множествами . Операции преобразования и эквивалентности.

Операции выбора элемента, квантификаторы, агрегация. Операции выбора элемента *First*, *Last*. Квантификаторы *Any*, *All*, *Contains*. Агрегация. Операция *Count*. Операции *Sum*, *Min*, *Max*, *Average*. Операция *Aggregate*.

Тема 5. Основные принципы WPF. Общие сведения о элементах управления. ОПК-3.1

Основные принципы WPF. Язык разметки XAML. Объектная модель WPF. Процесс компоновки. Простая компоновка с помощью *StackPanel*. Элемент *Border*. Панели *WrapPanel* и *DockPanel*. Вложение контейнеров компоновки. Панели *Grid* и *UniformGrid*. Разделенные окна. Панель *Canvas*. Контейнер *InkCanvas*.

Общие сведения о элементах управления, базовый класс *Control*. Элементы управления содержимым *ContentControl*: Метки, Кнопки (Класс *Button*, *RepeatButton*, *ToggleButton*, класс *CheckBox*, *RadioButton*), всплывающие подсказки, контекстное меню (Класс *Popup*), специализированные контейнеры. Элементы управления содержимым с заголовками. Текстовые элементы управления. Элементы управления списками *ItemsControl*. Элементы управления, основанные на диапазонах значений. Элементы управления датами.

Тема 6. Создание приложений с графическим пользовательским интерфейсом на C#. ОПК-3.1

Свойства зависимости, пузырьковые и туннелируемые события. Привязка данных. Связывание элементов. Привязка к объектам, не являющимся элементами. Проектирование объектов данных. Построение интерфейса и простого хранилища данных. Создание хранилища данных с оповещением об

изменениях.

Ресурсы. Статические и динамические ресурсы. Поиск ресурсов. Коллекция ресурсов. Доступ к ресурсам в коде. Ресурсы приложения. Создание и использование словаря ресурсов. Стили. Создание и применение объекта стиля. Триггеры. Фигуры, кисти и трансформации.

Основы анимации. Анимация на основе таймера. Анимация на основе свойств. Классы анимации. Шаблоны элементов управления. Логические и визуальные деревья. Создание шаблонов элементов управления.

Тема 7. Технологии ввода-вывода и хранения данных. ОПК-3.1

Технологии ввода-вывода и хранения данных. Поток данных, ввод и вывод данных. Работа с форматом данных *JSON*. Работа с форматом данных *XML*. Взаимодействие с базами данных, технология объектно-ориентированной трансляции и библиотека *Entity Framework Core*. Создание баз данных, привязка к базе данных. Библиотека построения графиков и диаграмм *ScottPlot*.

4. Типы оценочных материалов, показатели и критерии оценивания

4.1 Оценочные материалы по дисциплине Б1.О.11 Программирование входят в состав оценочных материалов по образовательной программе. Совокупность оценочных материалов образовательной программы составляет фонд оценочных средств (далее – ФОС). ФОС используется при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся с целью оценивания достижения обучающимися планируемых результатов обучения.

4.2. ФОС разработан как комплекс проверочных заданий различного типа и уровня сложности, включает критерии и шкалы оценивания, а также «ключи» правильных ответов. ФОС формируется как отдельный документ и хранится в электронном виде, доступ к ФОС предоставлен ограниченному кругу лиц.

4.3. Для самостоятельной работы обучающихся при подготовке к текущему контролю успеваемости и промежуточной аттестации в рабочих программах дисциплин размещены типовые проверочные задания, которые можно условно разделить на задания закрытого, комбинированного и открытого типов.

Задания закрытого типа — это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных.

Задания комбинированного типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных и обосновать свой выбор.

Задания открытого типа — это задания, в которых на каждый вопрос должен быть предложен развернутый обоснованный ответ.

В зависимости от типа задания рекомендованы определенная последовательность выполнения и система оценивания выполнения заданий.

4.4 Типы заданий, сценарии выполнения, критерии оценивания

ТИП ЗАДАНИЯ	ИНСТРУКЦИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	КРИТЕРИИ ОЦЕНИВАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных	Прочитайте текст, выберите правильный ответ	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В). 	Ответ считается верным, если правильно указана цифра или буква
Задание закрытого типа на установление соответствия	Прочитайте текст и установите соответствие	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4). 	Ответ считается верным, если правильно указаны цифры или буквы
Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных	Прочитайте текст, выберите правильные ответы	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранных вариантов ответа (например, 1 4 или А Г). 	Ответ считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)
Задание закрытого типа на установление последовательности	Прочитайте текст и установите последовательность	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты 	Ответ считается верным, если правильно указана вся последовательность цифр

		<p>ответа.</p> <p>3. Построить верную последовательность из предложенных элементов.</p> <p>4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135).</p>	
<p>Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора</p>	<p>Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа</p>	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.</p> <p>2. Внимательно прочитать предложенные варианты ответа.</p> <p>3. Выбрать один верный ответ.</p> <p>4. Записать только номер (или букву) выбранного варианта ответа.</p> <p>5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования).</p>	<p>Ответ считается верным, если правильно указана цифра или буква и приведены корректные аргументы, используемые при выборе ответа</p>
<p>Задание открытого типа с развернутым ответом</p>	<p>Прочитайте текст и запишите развернутый обоснованный ответ</p>	<p>1. Внимательно прочитать текст задания и понять суть вопроса.</p> <p>2. Продумать логику и полноту ответа.</p> <p>3. Записать ответ, используя четкие компактные формулировки.</p> <p>4. В случае расчетной задачи, записать решение и ответ</p>	<p>Ответ считается верным:</p> <p>1. Отсутствие фактических ошибок.</p> <p>2. Раскрытие объема используемых понятий (полнота ответа).</p> <p>3. Обоснованность ответа (наличие аргументов).</p> <p>4. Логическая последовательность излагаемого материала.</p>

4.5 Общая шкала оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся с применением БРС

Итоговая балльная оценка	Традиционная система	Бинарная система	ECTS	
			Для традиционной системы	Для бинарной системы
95-100	Отлично	Зачтено	<i>A</i>	<i>P/ Passed</i>
85-94			<i>B</i>	<i>P/ Passed</i>
75-84	Хорошо		<i>C</i>	<i>P/ Passed</i>
65-74			<i>D</i>	<i>P/ Passed</i>
55-64	Удовлетворительно		<i>E</i>	<i>P/ Passed</i>
0-54	Неудовлетворительно	Не зачтено	<i>F</i>	<i>F/Failed</i>

Соотношение баллов за текущий контроль успеваемости и промежуточную аттестацию, а также повторную промежуточную аттестацию:

Максимальная сумма баллов за текущий контроль успеваемости	Максимальная сумма баллов за промежуточную аттестацию	Максимальная итоговая балльная оценка	Максимальная сумма баллов за повторную промежуточную аттестацию
60 баллов	40 баллов	100 баллов	100 баллов

5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам

5.1. В ходе реализации дисциплины используются следующие формы текущего контроля успеваемости обучающихся (в том числе, задания к контрольным точкам):

Тестирование (Т), практические контрольные задания (ПКЗ), контрольная работа (КР).

Тема 1. Введение в теорию алгоритмов.

Тест 1.1.

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитать предложенные варианты ответа.
3. Выбрать один верный ответ.
4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В)

- 1) Определить понятие алгоритма, как
- 1) Последовательность действий, выполняемых пользователем
- 2) Точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи
- 3) Последовательное выполнение команд в процессоре
- 4) Метод в математике

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитать предложенные варианты ответа.
3. Выбрать один правильный ответ.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

2) Определить конечное значение переменной Z.

$x = 1; y = 1; z = 0;$

если $(y - x) > 0$ то

если $z <> 0$ то

$z = z + 1;$

если $y > 0$ то

$z = x + 1;$

иначе

$z = 2 * y;$

вывод z;

- 1) 0
- 2) 1
- 3) 2
- 4) 3

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитать предложенные варианты ответа.
3. Выбрать один верный ответ.
4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В)

Задание закрытого типа с выбором одного правильного ответа из

нескольких вариантов предложенных.

1. Внимательно прочитайте текст задания и поймите, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитайте предложенные варианты ответов.
3. Выберите один верный ответ.
4. Запишите только номер (или букву) выбранного варианта ответа (например, 3 или В)

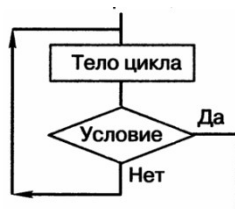
3) Какое логическое выражение истинно, если $x \in [-10, 10]$?

- 1) $(x > 10)$ И $(x < -10)$
- 2) $(x > 10)$ ИЛИ $(x < -10)$
- 3) $(x < 10)$ ИЛИ $(x \geq -10)$
- 4) $(x \geq -10)$ И $(x \leq 10)$

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитайте текст задания и поймите, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитайте предложенные варианты ответов.
3. Выберите один верный ответ.
4. Запишите только номер (или букву) выбранного варианта ответа (например, 3 или В)

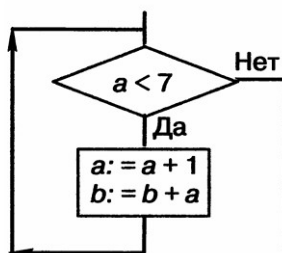
4) К какому виду алгоритмов можно отнести алгоритм, схема которого представлена ниже?



- 1) разветвляющийся с неполным ветвлением
- 2) разветвляющийся с полным ветвлением
- 3) цикл с заданным условием окончания работы
- 4) цикл с заданным числом повторений
- 5) цикл с предусловием

Задание открытого типа с развернутым ответом. Прочитайте текст и запишите развернутый обоснованный ответ

1. Внимательно прочитайте текст задания и поймите суть вопроса.
2. Продумайте логику и полноту ответа.
3. Запишите ответ, используя четкие компактные формулировки.
4. В случае расчетной задачи, запишите решение и ответ
- 5) Выполните фрагмент алгоритма при $a = 2$ и $b = 0$.



Определите значение переменной b после выполнения фрагмента алгоритма.

Тема 2. Основные конструкции языка программирования C#.

Тест 2.1.

Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора.

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.

2. Внимательно прочитайте предложенные варианты ответа.

3. Выбрать один верный ответ.

4. Записать только номер (или букву) выбранного варианта ответа.

5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования).

1) В теле метода `Main()` содержится следующий код:

```
string s = "?234",
s2 = "56";
s = '1' + s[1] + s[2] + s[3];
s = s + s2;
Console.Write(s);
```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение, если код символа 1 равен 49:

- 1) 1
- 2) 123456
- 3) 112356
- 4) программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок
- 5) 69

Задание открытого типа с развернутым ответом. Прочитайте текст и запишите развернутый обоснованный ответ

1. Внимательно прочитайте текст задания и понять суть вопроса.

2. Продумать логику и полноту ответа.

3. Записать ответ, используя четкие компактные формулировки.

4. В случае расчетной задачи, записать решение и ответ

2) В теле метода `Main()` содержится следующий код:

```
string S = null;
```

```
int B = 30;  
S += B + 10;  
System.Console.Write(S);
```

Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитайте предложенные варианты ответа.
3. Выбрать несколько правильных ответов.
4. Записать только номера (или буквы) выбранных вариантов ответа (например, 1 4 или А Г).

3) Какие преобразования типов НЕ выполняются автоматически (возможно, несколько вариантов):

- 1) Из *int* в *short*
- 2) Из *short* в *int*
- 3) Из *bool* в *string*
- 4) Из *byte* в *float*

Задание закрытого типа на установление последовательности.

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается последовательность элементов.
2. Внимательно прочитайте предложенные варианты ответа.
3. Построить верную последовательность из предложенных элементов.
4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БАВ или 135).

4) Расставьте в правильной последовательности операторы для создания массива из 10 целых чисел.

Ключевые слова:

- 1) []
- 2) [10]
- 3) *int*
- 4) *nums*
- 5) = (4)
- 6) *new*

Задание закрытого типа на установление соответствия

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидаются пары элементов.
2. Внимательно прочитайте оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д.
3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов.

4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).

5) Задайте соответствие терминов и определений

1) переопределенный метод	1) метод, унаследованный от класса-родителя и измененный в классе-наследнике
2) перегруженный метод	2) метод, имеющий имя уже разработанного в классе, но отличающийся числом или типом входных параметров

Тема 3. Основы объектно-ориентированного программирования на C#.

Контрольная работа 3.1.

Задание 1. Консольный вывод и вывод

Даны два числа типа *double* 1,56 и 3,45. Вычислите их частное и выведите результат в консоль с точностью до трех знаков.

Задание 2. Преобразование типов

Вычислите квадрат числа 3.45, преобразуйте результат в тип данных целое число. Результат выведите в консоль.

Задание 3. Логические выражения

Вычислить значение логического выражения при следующих значениях логических величин *A*, *B* и *C*: *A* = Истина, *B* = Ложь, *C* = Ложь:

а) *A* или *B*; б) *A* и *B*; в) *B* или *C*.

Задание 4. Вычисление формул с оператором *if*

Вычислите значение выражения

$$Z = \begin{cases} 10,5x^2 + 1, & \text{если } x < 0; \\ 5(x+1)^3, & \text{если } 0 \leq x < 1, \\ \sqrt{x^2 + 1}, & \text{если } x \geq 1 \end{cases}$$

Значение *x* вводится пользователем. Значения *Z* выведите в консоль в виде $Z(x = \text{значение}) = \text{значение}$. Задачу решить используя оператор ветвления *if...else*.

Задание 5. Вычисление формул с оператором *switch*

Вычислите значение выражения

$$Z = \begin{cases} \frac{-1}{x^2} + 1, & \text{если } x \leftarrow 1; \\ x^2 + 1, & \text{если } -1 \leq x \leq 2; \\ 5, & \text{если } x > 2 \end{cases}$$

Значение x вводится пользователем. Значения Z выведете в консоль в виде $Z(x = \text{значение}, k = \text{значение}) = \text{значение}$. Задачу решить используя оператор ветвления *switch...case*.

Задание 6. Использование сложных условий

Пользователь вводит некоторое двузначное число. Определить:

- а) входят ли в него цифры 4 или 7;
- б) входят ли в него цифры 3, 6 или 9.

Задание 7. Использование цикла *for*

Используя цикл *for*, вычислите значение выражения

$$A = \prod_{k=1}^5 (x^2 - k),$$

x принимает значение от -2 до 2 с шагом 0,2. Значения Z выведете в консоль в виде

$$A(x = \text{значение}) = \text{значение}.$$

Задание 8. Использование цикла *while*

В программе задан массив [1, -2, 5, 8, -4, 3, -6, 7, -9, 11]. Рассчитайте сумму элементов массива, модуль которых больше 4. Для решения задачи используйте цикл *while*.

Задание 9. Использование цикла *foreach*

В программе задан массив [1, -2, 5, 8, -4, 3, -6, 7, -9, 3, 2, 0, -1, 11]. Обнулите элементы массива, модуль значения которых находится в интервале [2, 4]. Для решения задачи используйте цикл *foreach*.

Задание 10. Матрицы и многомерные массивы

Раскрасьте матрицу - шахматную доску размера 8*8 в черный и белый цвета. Элемент с индексом [0][0] раскрасьте в белый цвет, а далее в шахматном порядке. Для обозначения черного цвета используйте символ "b", для обозначения белого цвета — символ "w". (Подсказка - сумма $i+j$ для белых клеток четное, для черных нечетное).

Задание 11. Перечисления

Создайте перечисление *Summer*, в основе которого лежит тип *byte*. В перечислении названия летних месяцев, которым должна соответствовать числовая константа номера месяца в году. Создайте переменную *summer* типа *Summer*, задайте ей значение «June», выведете на экран строковое и числовое значение переменной.

Задание 12. Классы, поля, свойства

Создайте класс *Sum*, имеющий:

закрытые поля: *num1* типа целое число без знака, *num2* типа целое число без знака;

открытое свойство *Num* типа целое число, имеющее только аксессор *get* и возвращающее сумму *num1* и *num2*.

Задание 13. Классы и методы

Создайте класс *Sum*, имеющий:

- закрытые поля: *num1* типа целое число без знака, *num2* типа целое число без знака;
- открытый метод *GetNum*, принимающий один параметр *num2* типа целое число, и возвращающий сумму двух полей и параметра в виде числа с плавающей запятой.

Создайте объект класса *Div*, выведете в консоль метода результат работы метода *GetNum*.

Задание 14. Инициализация классов, конструкторы

Создайте класс *B*, имеющий:

- закрытое поле *num1* типа целое число без знака;
- закрытое поле *num2* типа число с плавающей запятой;
- конструктор, принимающий два параметра *num1* и *num2* и инициализирующий соответствующие поля;
- конструктор без параметров, инициализирующий поля *num1* и *num2* значениями по умолчанию 5 и 5.0 соответственно.

Создайте объекты класса *B*, различными способами.

Задание 15. Классы и перегрузка стандартного метода *ToString*

Создайте класс *B*, имеющий:

- закрытое поле *num1* типа целое число без знака;
- закрытое поле *num2* типа число с плавающей запятой.

Перегрузите стандартный метод класса *object* – *ToString* так, чтобы он выводил строку в следующем формате:

num1 = значение, *num1* = значение

Задание 16. Классы и перегрузка стандартного метода *Equals*

Создайте класс *B*, имеющий:

- открытое поле *num1* типа целое число без знака;
- открытое поле *num2* типа число с плавающей запятой.

Перегрузите стандартный метод класса *object* – *Equal* так, чтобы он определял эквивалентность объектов по равенству соответствующих полей *num1* и *num2* сравниваемых объектов.

Создайте два объекта класса *B*, выведите в консоль результат работы перегруженного метода *Equals*.

Задание 17. Перегрузка операций

Создайте класс *Person*, имеющий:

- автоматическое открытое свойство *Name* строкового типа;
- автоматическое открытое свойство *Years* типа целое число без знака.

Перегрузите бинарную операцию сложения для объектов класса *Person*, результатом операции должен быть новый объект класса *Person*. В новом объекте свойство *Name* имеет значение свойства *Name* первого слагаемого, свойство *Years* имеет значение суммы значений свойств *Years* объектов участвующих в операции.

Создайте два объекта класса *Person*. Создайте третий объект типа *Person* как результат сложения первых двух объектов.

Задание 18. Классы и наследование

Создайте класс *TwoD* имеющий:

- два закрытых поля *x* и *y*, имеющих тип число с плавающей запятой;
- метод *GetCoord* доступный для перегрузки в классах наследниках, возвращающий строку «*X* = значение, *Y* = значение».

Создайте класс *ThreeD*, являющийся производным классом от класса *TwoD* и имеющий дополнительное поле *z* типа число с плавающей запятой.

Переопределите метод *GetCoord* так, чтобы он возвращал строку «*X* = значение, *Y* = значение, *Z* = значение».

Создайте два объекта базового класса и класса наследника, выведите в консоль результат метода *GetCoord* обоих объектов.

Задание 19. Интерфейсы, реализация интерфейсов

Создайте интерфейс *ICalc*, содержащий сигнатуру одного метода *Incr*. Метод *Incr* принимает один параметр типа целое число и возвращает результат типа целое число.

Создайте класс *B*, реализующий данный интерфейс. Реализация метода *Incr* в классе *B* должна возвращать значение параметра плюс 3.

Задание 20. Обобщения

Создайте обобщенный класс *General* с одним универсальным параметром *T*. Класс *General* имеет поле обобщенного типа, а также метод *Display* выводящий значение обобщенного поля в консоль.

Создайте объект обобщенного класса *General* с универсальным параметром типа *int*.

Тема 4. Концепция абстрактных типов данных. Структуры данных C#.

Тест 4.1

Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитайте предложенные варианты ответа.
3. Выбрать один верный ответ.
4. Записать только номер (или букву) выбранного варианта ответа.
5. Записать аргументы, обосновывающие выбор ответа.

1) Какой абстрактный тип данных представлен на рисунке?



- 1) Дек
- 2) Очередь
- 3) Вектор
- 4) Двусвязный список
- 5) Односвязный список

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитайте предложенные варианты ответа.
3. Выбрать один правильный ответ.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

2) Какие из приведенных утверждений верны в отношении коллекции *Stack*?

- 1) Его можно использовать для вычисления выражений
- 2) Он используется для ведения списка *FIFO*
- 3) Доступ к самому верхнему элементу коллекции *Stack* можно получить с помощью *Peek()*
- 4) Все упомянутые

Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.
2. Внимательно прочитать предложенные варианты ответа.
3. Выбрать один верный ответ.
4. Записать только номер (или букву) выбранного варианта ответа.
5. Записать аргументы, обосновывающие выбор ответа.

3) После выполнения следующего фрагмента кода на экран будет выведено:

```
Stack stack = new Stack();
stack.Push("son");
stack.Push("father");
stack.Push("grandfather");
while (stack.Count!=0)
{
    Console.Write("{0} ", stack.Pop().ToString());
}
```

- 1) grandfather father son
- 2) son father grandfather
- 3) Ошибка компиляции

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитать предложенные варианты ответа.
3. Выбрать один правильный ответ.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

4) Какой абстрактный тип данных представлен на рисунке?



- 1) Стек
- 2) Очередь
- 3) Вектор
- 4) Двусвязный список
- 5) Односвязный список

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитать предложенные варианты ответа.
3. Выбрать один правильный ответ.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

5) Какой тип данных используется в следующем коде?

```
var prodPrice = new Dictionary<string, double>()
{
    ["bread"] = 23.3,
    ["apple"] = 45.2
};
Console.WriteLine($"bread price: {prodPrice["bread"]}");
```

- 1) Словарь
- 2) Очередь
- 3) Вектор
- 4) Двусвязный список
- 5) Односвязный список

Тема 5. Введение в WPF. Создание приложений с графическим пользовательским интерфейсом на C#.

Тест 5.1

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

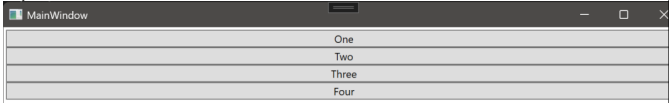
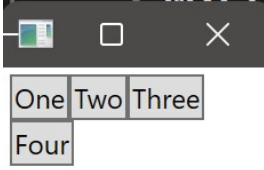
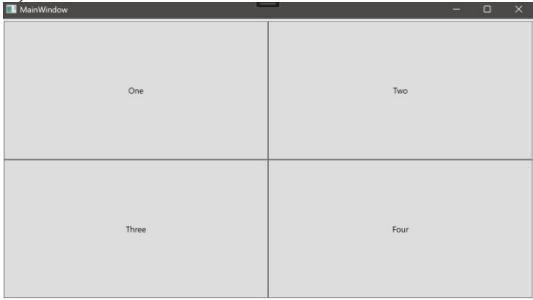
1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитать предложенные варианты ответа.
3. Выбрать несколько правильных ответов.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).


- 1) Какой элемент WPF используется в качестве корневого элемента окна?
 - 1) *Window*
 - 2) *StackPanel*
 - 3) *Canvas*
 - 4) *DockPanel*

Задание закрытого типа на установление соответствия

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов.
2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д.
3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов.
4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).

2) Какому коду соответствует окно на рисунке ниже?

<p>1)</p> <pre><StackPanel Margin="3"> <Button>One</Button> <Button>Two</Button> <Button>Three</Button> <Button>Four</Button> </StackPanel></pre>	<p>a)</p> 
<p>2)</p> <pre><WrapPanel Margin="3"> <Button>One</Button> <Button>Two</Button> <Button>Three</Button> <Button>Four</Button> </WrapPanel></pre>	<p>b)</p> 
<p>3)</p> <pre><Grid Margin="3"> <Grid.ColumnDefinitions> <ColumnDefinition/> <ColumnDefinition/> </Grid.ColumnDefinitions> <Grid.RowDefinitions> <RowDefinition/> <RowDefinition/> </Grid.RowDefinitions> <Button One</Button> <Button Grid.Column="1">Two</Button> <Button Grid.Row="1">Three</Button> <Button Grid.Column="1" Grid.Row="1">Four</Button> </Grid></pre>	<p>c)</p> 

<p>4)</p> <pre> <DockPanel Margin="3"> <Button DockPanel.Dock="Top">One</Button> <Button DockPanel.Dock="Bottom">Two</Button> <Button DockPanel.Dock="Left">Three</Button> <Button DockPanel.Dock="Right">Four</Button> </DockPanel> </pre>	<p>d)</p> 
---	--

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.

2. Внимательно прочитать предложенные варианты-ты ответа.

3. Выбрать несколько правильных ответов.

4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

3) Какой режим привязки позволяет постоянно обновлять исходное свойство (источника привязки) при изменении пользовательского интерфейса?

1) *OneWay*

2) *TwoWay*

3) *OneTime*

4) *OneWayToSource*

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.

2. Внимательно прочитать предложенные варианты-ты ответа.

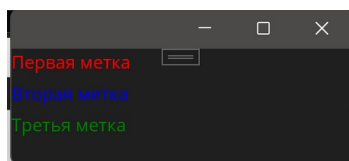
3. Выбрать несколько правильных ответов.

4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

Задание открытого типа с развернутым ответом.

1. Внимательно прочитать текст задания и понять суть вопроса.
2. Продумать логику и полноту ответа.
3. Записать ответ, используя четкие компактные формулировки.
4. В случае расчетной задачи, записать решение и ответ.

4) Запишите вариант XAML разметки окна представленного ниже.



Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитать предложенные варианты-ты ответа.
3. Выбрать несколько правильных ответов.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

5) Какое присоединяемое свойство используется для определения вертикальной позиции элемента в панели *Canvas*?

- 1) *Canvas.Left*
- 2) *Canvas.VerticalAlignment*
- 3) *Canvas.VerticalPosition*
- 4) *Canvas.Top*

Тема 6. Создание приложений с графическим пользовательским интерфейсом на C#.

Практическое контрольное задание 6.1

Сеть имеет 3 кинотеатра. Вместимость первого кинотеатра N_1 зрителей, второго N_2 зрителей, третьего N_3 зрителей. Расход на проведение сеанса составляет в первом кинотеатре R_1 рублей, второго R_2 рублей, третьего R_3 рублей. Стоимость билета в первом кинотеатре c_1 рублей, во втором c_2 рублей. Рассчитать стоимость билета в третьем кинотеатре так, чтобы общая суммы дохода была не менее чем на 25% больше расходов.

Разработать программу с графическим интерфейсом для решения описанной задачи. Требования:

- интерфейс должен содержать поля для ввода данных, поле для вывода результата и кнопку для запуска вычисления;
- форма должна иметь соответствующие заголовки и иконку;
- при попытке закрытия окна пользователю должен выдаваться вопрос с текстом «Выйти из программы?»;
- в полях ввода должна быть возможность ввода только корректных данных (Например, при попытке ввода текста вместо числа, должно выводиться соответствующее предупреждение);
- поле вывода результата должно быть доступно только для чтения.

Тема 7. Технологии ввода-вывода и хранения данных. ОПК-3.1

Практическое контрольное задание 7.1

Разработайте хранилище данных на основе коллекций C#. Наполните хранилище тестовыми данными продуктов и продаж. Разработайте графический пользовательский интерфейс. Осуществите привязку данных хранилища и использование хранилища с помощью графического пользовательского интерфейса.

Первая коллекция должна включать объекты, содержащие сведения о товарах:

Код товара	Наименование товара	Категория	Цена	Количество	Описание
------------	---------------------	-----------	------	------------	----------

Вторая коллекция должна включать сведения о клиентах:

Код клиента	Фамилия / Организация	Имя	Отчество	Физ. лицо или организация
-------------	-----------------------	-----	----------	---------------------------

Третья коллекция содержит сведения о заказах. Каждый заказ хранит сведения о товаре и клиентах.

Код заказа	Клиент	Товар	Количество	Дата	Описание
------------	--------	-------	------------	------	----------

5.2. Типовые оценочные материалы для текущего контроля успеваемости обучающихся (вне контрольных точек):

приведены в п.6.2.

5.3. Один или несколько тематических блоков дисциплины завершаются контрольной точкой (далее – КТ). Текущий контроль успеваемости по дисциплине предусматривает не менее 2 (двух) и не более 10 (десяти) КТ в течение периода освоения дисциплины.

Максимальное количество баллов за любой тип работ в рамках КТ составляет 100 (сто) баллов.

Распределение весовых коэффициентов по КТ в рамках текущего контроля успеваемости по дисциплине и формулы расчета:

Наименование контрольной точки	Максимальное количество баллов за	Коэффициент веса	Результат контрольной точки, участвующий в
--------------------------------	-----------------------------------	------------------	--

	работу в рамках КТ, которое может набрать студент	контрольной точки	формировании итоговой балльной оценки по дисциплине (отражается в журнале БРС в СДО)
1 семестр			
Тема 1 - Введение в теорию алгоритмов. Основы языка программирования C#.			
КТ - 1 Тест 1.1	100	0,1	10
Тема 2 - Основы языка программирования C#.			
КТ - 2 Тест 2.1	100	0,2	20
Тема 3 - Основы объектно-ориентированного программирования на C#.			
КТ - 3 КР 3.1	100	0,2	10
Тема 4 - Концепция абстрактных типов данных. Структуры данных C#.			
КТ - 4 Тест 4.1	100	0,1	26
Итого:	x	0,6	60
2 семестр			
Тема 5 - Введение в WPF. Создание приложений с графическим пользовательским интерфейсом на C#.			
КТ - 5 Тест 5.1	100	0,2	20
Тема 6 - Создание приложений с графическим пользовательским интерфейсом на C#.			
КТ - 6 ПКЗ 6.1	100	0,2	40
Тема 7 - Технологии ввода-вывода и хранения данных.			
КТ - 7 ПКЗ 7.1	100	0,2	
Итого:	x	0,6	60

Формула расчета результата контрольной точки:

Результат контрольной точки = Количество баллов за работу в рамках КТ X Коэффициент веса контрольной точки.

5.4. Формы текущего контроля успеваемости обучающихся в рамках КТ и типовые оценочные материалы:

1 семестр

- КТ-1: Тема 1. Тестирование 1.1.**
- КТ-2: Тема 2. Тестирование 2.1.**
- КТ-3: Тема 3. Контрольная работа 3.1**
- КТ-4: Тема 4. Тестирование 4.1.**

2 семестр

- КТ-1: Тема 5. Тестирование (ПКЗ) 5.1.**
- КТ-2: Тема 6. Практическое контрольное задание (ПКЗ) 6.1.**
- КТ-3: Тема 7. Практическое контрольное задание (ПКЗ) 7.1.**

Для каждой формы текущего контроля успеваемости обучающихся в рамках КТ определены критерии оценивания результатов выполнения задания.

1. Критерии оценивания тестирования:

Критерии оценки	Диапазон баллов	Описание критерия
Количество правильных ответов	0	Количество правильных ответов менее 55%
	25	Количество правильных ответов от 55% до 64%
	50	Количество правильных ответов от 65% до 74%
	75	Количество правильных ответов от 75% до 84%
	100	Количество правильных ответов от 85% до 100%
Итого максимально:	100	

2. Критерии оценивания ПКЗ:

Критерии оценки	Диапазон баллов	Описание критерия
Содержание и раскрытие выбранных понятий	41-70	Детальное описание всех понятий на примере разработанной программы
	21-40	Поверхностное описание всех понятий на примере разработанной программы
	0-20	Понятия раскрыты минимально или не раскрыты вовсе
Количество выполненных заданий	30	Количество выполненных заданий от 85% до 100%
	15	Количество выполненных заданий от 55% до 84%
	0	Количество выполненных заданий менее 55%
Итого максимально:	100	

3. Критерии оценивания КР

Критерии оценки	Диапазон баллов	Описание критерия
Содержание и раскрытие выбранных понятий	41-70	Детальное описание всех понятий на примере разработанной программы
	21-40	Поверхностное описание всех понятий на примере разработанной программы
	0-20	Понятия раскрыты минимально или

		<i>не раскрыты вовсе</i>
<i>Количество выполненных заданий</i>	<i>30</i>	<i>Количество выполненных заданий от 85% до 100%</i>
	<i>15</i>	<i>Количество выполненных заданий от 55% до 84%</i>
	<i>0</i>	<i>Количество выполненных заданий менее 55%</i>
Итого максимально:	100	

5.5. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий (*при необходимости*).

Для решения задач открытого типа (кейсов, ПКЗ), тестовых заданий студенту разрешается использование онлайн компиляторов, справочной системы Microsoft и других сторонних сайтов. Для построения блок-схем алгоритмов студенту можно использовать любой соответствующий онлайн-инструмент.

6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине

6.1. Промежуточная аттестация проводится в форме:

1 семестр - **зачет**; 2 семестр - **курсовой проект, экзамен**.

Зачет и экзамен проводится в компьютерном классе в письменной форме. Обучающийся получает экзаменационный билет с вариантами 3-х заданий различного типа. На выполнение заданий дается 40-60 минут. По завершении подготовки необходимо представить ответы в письменном виде, подробно изложив ход выполнения задания, сделать выводы (*при необходимости*).

При реализации промежуточной аттестации в ЭО/ДОТ могут быть использованы следующие формы: устно в ДОТ - в форме обоснованных ответов на задания различного типа; письменно в СДО - в форме письменного решения заданий различного типа; тестирование в СДО.

Курсовой проект представляет собой законченную программу с графическим пользовательским интерфейсом на языке программирования C#. Защита курсовой работы (проекта) проводится в форме публичного выступления студента. Защита курсового проекта состоит из доклада студента по теме курсовой работы (проекта) и ответов студента на вопросы преподавателя(ей).

6.2. Типовые оценочные материалы промежуточной аттестации.

Вопросы для подготовки к зачету.

1. Что такое среда выполнения *Common Language Runtime (CLR)*?
2. Дать определение алгоритма, формализованное определение алгоритма на основе частично-рекурсивных функций. Перечислите основные свойства алгоритмов. Дайте описание способам представления алгоритма.
3. Что такое наследование? Поддерживает ли C # множественное наследование?

Привести примеры

4. В чем разница между структурой и классом в C #? Привести примеры.
5. Что такое свойства в C#? Что такое *enum* в C#?
6. Что такое массивы? Что такое неровные массивы? Привести примеры.
7. Что такое пространства имен в C #? Привести примеры использования.
8. Что такое индексомеры в C # .NET? Привести примеры использования.
9. Что такое классы *System.String* и *System.Text.StringBuilder*? Где используются, основные операции?
10. Рассказать об использовании оператора *if ...else, switch .. case* и привести примеры.
11. Сформулировать назначение обработчика ошибок и рассмотреть обзор типовой структуры обработчика ошибок.
12. Дать определение цикла, перечислить различные типы циклов, привести примеры блок схем.
13. Дать определение массиву, перечислить различные виды массивов.
14. Перечислить принципы объектно ориентированного программирования.
15. Сделать обзор объектно-ориентированных языков программирования.
16. Дать определение объекту, привести примеры объектов.
17. Дать определение понятиям свойство, событие, метод.
18. Дать обзор типов классов в C #?
19. Сделать обзор абстрактных классов и интерфейсов. В чем разница между абстрактным классом C # и интерфейсом?
20. Что такое частичные классы и частичные методы в C #? Приведите примеры.

Типовые задания для зачета.

1. Создайте класс *Condition*, реализующий метод *Calc* вычисления выражения, представленного ниже. Метод *Calc* должен принимать параметр *x* и возвращать значение *y*. Задание выполнить используя шаблон консольного приложения *VS* на языке C#.

$$y = \begin{cases} |x-5|, x \leq 4 \\ x^3 - x^2, 21 \geq x > 4 \\ -x^3(1+2x+x^2), x > 21 \end{cases}$$

2. Создайте класс *IterFunc*, реализующий метод *CalcFunc* вычисления выражения, представленного ниже. Метод *CalcFunc* должен принимать параметр *x* и возвращать значение *y*. Задание выполнить используя шаблон консольного приложения на языке C#.

$$y = \sum_{i=1}^{10} x^3 \cos x - x^2 \sin x.$$

3. Создайте класс *Person*, содержащий закрытые поля *name*, *surname* (фамилия), массив *salary* (зарплата за каждый месяц в году), *extraSalary* (премия за каждый месяц в году). Доступ к закрытым полям обеспечивается через

открытые аксессоры соответствующих свойств. Кроме того, класс *Person* должен реализовывать метод *TotalSalary*, возвращающий общий доход за год.

Вопросы для подготовки к экзамену.

1. Какие парадигмы программирования существуют? Какие компоненты включает в себя *.Net*? Как выполняются программы *.Net*?
2. Сформулировать принципы структурного программирования. Сформулировать принципы объектно-ориентированного программирования. Поддерживает ли *C #* множественное наследование? Что такое пространства имен в *C #*? Привести примеры использования.
3. Дать определение алгоритму. Перечислите основные свойства алгоритмов. Перечислите основные алгоритмические структуры и способы описания алгоритма.
4. Дать определение интегрированной среды разработки. Сделать обзор структуры ИСР *VisualStudio*.
5. Рассмотреть в деталях структуру оператора *if..*, оператора *switch...case*. Привести примеры использования.
6. Дать определение синтаксическим ошибкам, логическим или алгоритмическим ошибкам и ошибкам времени выполнения. Что делает блок операторов *try-catch*. Сделать обзор основных операторов, используемых при обработке исключительных ситуаций.
7. Перечислить различные виды циклов. В чем отличие между *break* и *continue*.
8. Дать определение массива, коллекции. Что такое *enum* в *C#*?
9. Какие встроенные типы данных существуют в *C#*. Что обозначает ключевое слово *var*.
10. Какие бывают массивы? Что такое неровные массивы? Привести примеры. Что такое индексы в *C # .NET*? Привести примеры использования.
11. Что такое класс? В чем разница между структурой и классом в *C #*? Привести примеры. Дать определение объекту. Что такое частичные классы и частичные методы в *C #*? Приведите примеры.
12. Что такое абстрактные классы и интерфейсы. В чем разница между абстрактным классом и интерфейсом в *C #*?
13. Что такое поле, свойство, методов класса? В чем различие свойства и поля? В чем разница между автоматическими и полными свойствами? Привести примеры
14. Объясните порядок вызова конструкторов в иерархии наследования. Что произойдет, если в классе наследнике определить переопределить не виртуальный метод? Как в пользовательских классах производится переопределение операций?
15. Что такое *WPF*? Каковы его преимущества по сравнению с *WinForms*? Что такое *XAML*? Где применяется этот язык. Приведите простой пример. Объясните процесс компоновка пользовательского интерфейса в приложениях *WPF*? Какие контейнеры компоновки наиболее часто используются?

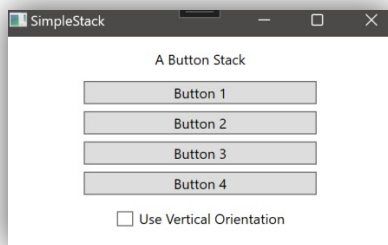
16. Какие элементы управления содержимым существуют? Определите их основные свойства.
17. Что такое свойства зависимости, прикрепляемые свойства, маршрутизируемые события? Приведите примеры. Что такое присоединенные свойства?
18. Что такое ресурсы в приложениях *WPF*, и какой способ определения ресурсов является наилучшим? Что такое кисти в приложениях *WPF*? Какие кисти поддерживает *WPF*?
19. Что такое *MVVM*? Может ли использование *MVVM* принести пользу разработчику? Что такое привязка данных?
20. Что такое шаблон элемента управления в приложениях *WPF*? Как можно использовать шаблон элемента управления в *WPF*?
21. Что такое анимация в приложениях *WPF*? Какая бывает анимация, как она реализуется.
22. Какие коллекции данных существуют в *C#*? Какие основные свойства и методы коллекций *C#*.

Типовые задания для экзамена.

1. Создайте класс *Rectangles* содержащий, поля *Description* (описание набора прямоугольников), массивы: *height* (высота) и *width* (ширина) для 10 прямоугольников. Доступ к закрытым полям обеспечивается через открытые аксессоры соответствующих свойств. В классе должен реализован метод *TotalSquare*, возвращающий общую площадь всех прямоугольников.

2. Создайте класс *Student*, содержащий закрытые поля *name* (имя), *surname* (фамилия), массив *grade_1* (12 оценок по учебной дисциплине 1), *grade_2* (12 оценок по учебной дисциплине 2), а также метод *AverageGrade* возвращающий средний балл по двум учебным дисциплинам. Доступ к закрытым полям обеспечивается через открытые аксессоры соответствующих свойств.

3. Создайте приложение с простым графическим интерфейсом, показанным ниже.



Типовые темы курсового проекта:

С целью повышения конкурентоспособности руководителем предприятия, принято решение автоматизировать документооборот путем создания автоматизированной информационной системы. Отделу бизнес-аналитики поставлена задача разработать приложение с графическим пользовательским

интерфейсом на платформе *WPF* с использованием языка *C#*, обеспечивающее следующий функционал:

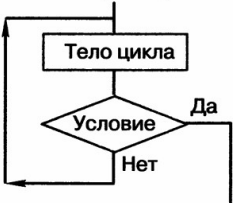
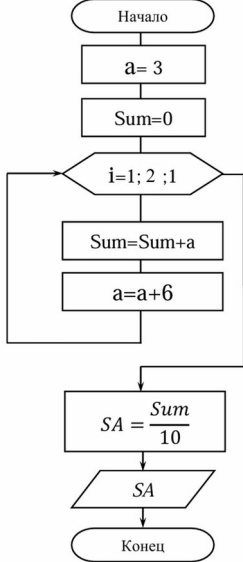
- 1) Возможность долгосрочного хранения информации в БД.
- 2) Возможность отображения, добавления, удаления и изменения данных.
- 3) Возможность поиска требуемой информации в БД по коду объекта.
- 6) Возможность сортировки информации.
- 7) Возможность фильтрации.
- 8) Возможность получения отчетов в виде диаграмм и *excel* файлов.

Варианты:

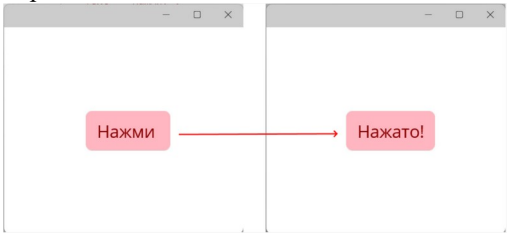
1. Разработка модуля ИС для гостиницы
2. Разработка модуля ИС для общежития
3. Разработка модуля ИС для библиотеки
4. Разработка модуля ИС для агентства недвижимости
5. Разработка модуля ИС для туристического агентства
6. Разработка модуля ИС для мебельного магазина
7. Разработка модуля ИС для страховой компании
8. Разработка модуля ИС для строительной компании
9. Разработка модуля ИС для определения сферы деятельности
10. Разработка модуля ИС для кафе
11. Разработка модуля ИС для автовокзала
12. Разработка модуля ИС для экскурсионного бюро
13. Разработка модуля ИС для спортивного клуба
14. Разработка модуля ИС «Расчет зарплаты»
15. Разработка модуля ИС «Расчет пенсии»
16. Разработка модуля ИС «Расчет военной пенсии»
17. Разработка модуля ИС «Страхование недвижимости»
18. Разработка модуля ИС «Страхование автомобиля»
19. Разработка модуля ИС «Кадровое агентство»
20. Разработка модуля ИС «Психологический тест»
21. Разработка модуля ИС «Тестирование знаний по иностранному языку»
22. Разработка теста остаточных знаний
23. Разработка модуля ИС «Кастинг на должность»
24. Разработка модуля ИС «Регистрация на самолет»
25. Разработка модуля ИС «Успеваемость»
26. Разработка модуля ИС «Управление заказами»
27. Разработка модуля ИС «Управление человеческими ресурсами»
28. Разработка модуля ИС «Планирование бюджета»
29. Разработка модуля ИС «Электронный переводчик»
30. Разработка модуля ИС «Управление портфелем заказов»

Типовые проверочные задания для самоподготовки обучающегося к промежуточной аттестации:

ТИП ЗАДАНИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	ТИПОВЫЕ ЗАДАНИЯ
-------------	------------------------	-----------------

<p>Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных</p>	<p>1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитайте предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, А или В).</p>	<p>1. Какое логическое выражение истинно, если $x \in [-10, 10]$?</p> <ol style="list-style-type: none"> 1) $(x > 10) \text{ И } (x < -10)$ 2) $(x > 10) \text{ ИЛИ } (x < -10)$ 3) $(x < 10) \text{ ИЛИ } (x > -10)$ 4) $(x > -10) \text{ И } (x < 10)$ <p>2. К какому виду алгоритмов можно отнести алгоритм, схема которого представлена ниже?</p>  <ol style="list-style-type: none"> 1) разветвляющийся с неполным ветвлением 2) разветвляющийся с полным ветвлением 3) цикл с заданным условием окончания работы 4) цикл с заданным числом повторений 5) цикл с предусловием <p>3. Выполните фрагмент алгоритма при $a = 2$ и $b = 0$.</p>  <p>Определите значение переменной SA после выполнения фрагмента алгоритма.</p> <ol style="list-style-type: none"> 1) 0,6 2) 1,2 3) 1,8 4) 1,9 5) 2,0 6) 0,7 												
<p>Задание закрытого типа на установление соответствия</p>	<p>1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитайте оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2,</p>	<p>1. Установите соответствие между оператором и выполняемым действием.</p> <table border="1" data-bbox="852 1749 1442 1939"> <tr> <td>1) %</td> <td>а) Тернарная операция, аналог условной конструкции <i>if</i></td> </tr> <tr> <td>2) ?</td> <td>б) Остаток от деления</td> </tr> <tr> <td>3) !=</td> <td>в) Логическое И</td> </tr> <tr> <td>4) &&</td> <td>г) Не равно</td> </tr> </table> <p>2. Установите соответствие между оператором и выполняемым действием.</p> <table border="1" data-bbox="852 2007 1442 2085"> <thead> <tr> <th>Операция</th> <th>Приоритет</th> </tr> </thead> <tbody> <tr> <td>1. $-i$</td> <td>а) инкремент</td> </tr> </tbody> </table>	1) %	а) Тернарная операция, аналог условной конструкции <i>if</i>	2) ?	б) Остаток от деления	3) !=	в) Логическое И	4) &&	г) Не равно	Операция	Приоритет	1. $-i$	а) инкремент
1) %	а) Тернарная операция, аналог условной конструкции <i>if</i>													
2) ?	б) Остаток от деления													
3) !=	в) Логическое И													
4) &&	г) Не равно													
Операция	Приоритет													
1. $-i$	а) инкремент													

	сформировать пары элементов. 4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).	2.	b) выделение памяти
		3. <i>new</i>	c) инверсия
		4. $\dot{=} \dot{=}$	d) умножение
		5. $\dot{=}$	e) проверка на эквивалентность
		6. $+ + \dot{=}$	f) вычитание с присвоением
Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).	<p>1. Чем могут быть M1, M2, M3, M4, если дан следующий код? <code>public class C1 : M1, M2 { }</code> <code>public interface S1 : M3, M4 { };</code> Выберите все подходящие варианты: a) M1 – класс; b) M1 – интерфейс; c) M2 – класс; d) M2 – интерфейс; e) M3 – класс; f) M3 – интерфейс; g) M4 – класс; h) M4 – интерфейс;</p> <p>2. Что верно относительно следующего фрагмента кода? <pre>class A { public virtual void m1() { } } class B : A { public override void m1() { } } class C : B { public override void m1() { /* программный код */ } }</pre> 1) Из класса C невозможно обратиться к методу <i>m1()</i> класса A для одного и того же объекта; 2) Из класса C можно получить доступ к методу <i>m1()</i> класса B используя вызов <code>base.m1()</code>; 3) Из класса C можно получить доступ к методу <i>m1()</i> класса A с помощью вызова <code>((A) this).m1()</code> 4) Из класса C можно получить доступ к методу <i>m1()</i> класса A с помощью вызова <code>base.base.m1()</code>.</p>	
Задание закрытого типа на установление последовательности	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты ответа. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например,	<p>1. Расставьте в правильной последовательности операторы создания списка из 10 целых чисел. Ключевые слова: 1) <code>List<int></code> 2) <code>(10)</code> 3) <code>nums</code> 4) <code>=</code> 5) <code>New</code> 6) <code><int></code> 7) <code>List</code></p> <p>2. Расставьте в правильной последовательности основные этапы создания обобщенного класса <i>M1</i> с одним параметром:</p>	

	БВА или 135).	a) <i>class</i> b) { c) } d) <i>public</i> e) <i><T></i> f) <i>MI</i> g) <i>value</i>
Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа. 5. Записать аргументы, обосновывающие выбор ответа.	1. В каком случае требуется применить явное преобразование типов. 1) Из <i>byte</i> в <i>int</i> 2) Из <i>short</i> в <i>int</i> 3) Из <i>int</i> в <i>float</i> 4) Из <i>double</i> в <i>float</i> ; 5) Из <i>float</i> в <i>double</i> . Преобразование невозможно в случае если целевой диапазон значений не вмещает источник. 2. Какой результат выполнения следующего кода? <pre>public class A {public int num;} A obj1 = new A{num = 3}; A obj2 = new A(num = 5); A obj3 = new A(num = 6); obj2 = obj1 obj2.A += 3; Console.WriteLine(obj1.A==obj3.A)</pre> 1) <i>True</i> ; 2) <i>False</i> ; 3) Ошибка на этапе компиляции; 4) Ошибка на этапе выполнения. Присвоение значений ссылочных переменных ведет к копированию ссылок но не значений
Задание открытого типа с развернутым ответом	1. Внимательно прочитать текст задания и понять суть вопроса. 2. Продумать логику и полноту ответа. 3. Записать ответ, используя четкие компактные формулировки. 4. В случае расчетной задачи, записать решение и ответ.	1. Запишите вариант <i>XAML</i> разметки окна, представленного ниже. Запишите <i>C#</i> код обработчика события.  <pre><StackPanel> <Button Name="btn" Click="btnClick"> Нажми </Button> </StackPanel> public btnClick(object sender, EventArgs e) { btn.Content = "Нажато"; }</pre>

6.3. Критерии и шкала оценивания на основе БРС.

Критерии и балльная шкала определяются преподавателем

Зачет

КРИТЕРИИ ОЦЕНИВАНИЯ	РЕЗУЛЬТАТ
---------------------	-----------

	В БАЛЛАХ
<i>Дан полный, в логической последовательности развернутый ответ на поставленный вопрос, где он продемонстрировал знания предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно, и исчерпывающе отвечает на дополнительные вопросы, приводит собственные примеры по проблематике поставленного вопроса, решил предложенные практические задания без ошибок</i>	40
<i>Дан развернутый ответ на поставленный вопрос, где студент демонстрирует знания, приобретенные на лекционных и семинарских занятиях, а также полученные посредством изучения обязательных учебных материалов по курсу, дает аргументированные ответы, приводит примеры, в ответе присутствует свободное владение монологической речью, логичность и последовательность ответа. Однако допускается неточность в ответе. Решил предложенные практические задания с небольшими неточностями.</i>	30-39
<i>Дан ответ, свидетельствующий в основном о знании процессов изучаемой дисциплины, отличающийся недостаточной глубиной и полнотой раскрытия темы, знанием основных вопросов теории, слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры, недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа и решении практических заданий.</i>	20-29
<i>Дан ответ, который содержит ряд серьезных неточностей, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы, незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов, неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Выводы поверхностны. Решение практических заданий не выполнено, т.е. студент не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя.</i>	0-19

Экзамен

КРИТЕРИИ ОЦЕНИВАНИЯ	РЕЗУЛЬТАТ В БАЛЛАХ
<i>Дан полный, в логической последовательности развернутый ответ на поставленный вопрос, где он продемонстрировал знания предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно, и исчерпывающе отвечает на дополнительные вопросы, приводит собственные примеры по проблематике поставленного вопроса, решил предложенные практические задания без ошибок</i>	40
<i>Дан развернутый ответ на поставленный вопрос, где студент демонстрирует знания, приобретенные на лекционных и семинарских занятиях, а также полученные посредством изучения обязательных учебных материалов по курсу, дает аргументированные ответы, приводит примеры, в ответе присутствует свободное владение монологической речью, логичность и последовательность ответа. Однако допускается неточность в ответе. Решил предложенные практические задания с небольшими неточностями.</i>	30-39

<p><i>Дан ответ, свидетельствующий в основном о знании процессов изучаемой дисциплины, отличающийся недостаточной глубиной и полнотой раскрытия темы, знанием основных вопросов теории, слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры, недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа и решении практических заданий.</i></p>	20-29
<p><i>Дан ответ, который содержит ряд серьезных неточностей, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы, незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов, неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Выводы поверхностны. Решение практических заданий не выполнено, т.е. студент не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя.</i></p>	0-19

Курсовой проект

<p><i>Курсовой проект выполнен в полном соответствии с заданием, отличается глубиной проработки всех разделов содержательной части, пояснительная записка оформлена с соблюдением установленных правил (выполнено 100%: отображение, добавление, удаление, обновление данных, экспорт в Excel, столбиковая диаграмма + 2 дополнительных графика).</i></p> <p><i>Руководитель характеризует деятельность студента положительно (в частности, отмечает его инициативу, самостоятельность, систематичность работы на всех этапах ее выполнения).</i></p> <p><i>В докладе исчерпывающе, последовательно, четко и логически правильно изложена суть курсового проекта и его основные результаты. Студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач, сформулированных в задании. На все вопросы студент дает обстоятельные и правильные ответы, убедительно защищает свою точку зрения.</i></p>	Отлично
<p><i>Курсовой проект выполнен в соответствии с заданием, отличается глубиной проработки всех разделов содержательной части, пояснительная записка оформлена с соблюдением установленных правил (выполнено 75%: отображение, добавление, удаление, обновление данных).</i></p> <p><i>Руководитель характеризует деятельность студента положительно, возможно, с незначительными замечаниями.</i></p> <p><i>в докладе правильно изложена суть курсового проекта и его основные результаты.</i></p> <p><i>Студент достаточно твердо усвоил теоретический материал и может применять его самостоятельно и по указанию преподавателя;</i></p> <p><i>На большинство вопросов студентом даны правильные ответы, студент защищает свою точку зрения достаточно обоснованно.</i></p>	Хорошо
<p><i>Курсовой проект выполнен в основном правильно, но без необходимой проработки некоторых разделов (выполнено 55%: отображение, добавление, удаление, обновление данных простого репозитория).</i></p> <p><i>В докладе упущены некоторые принципиальные моменты содержательной части работы. На вопросы студент отвечает</i></p>	Удовлетворительно

<i>неуверенно или допускает серьезные ошибки.</i>	
<i>Содержание курсового проекта и доклад на защите не свидетельствует о формировании у студента соответствующих компетенций. Студент не может защитить свои решения, допускает грубые фактические ошибки при ответах на поставленные вопросы или вовсе не отвечает на них.</i>	Неудовлетворительно

6.4. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий (*при необходимости*).

Для решения задач открытого типа (кейсов, ПКЗ), тестовых заданий студенту разрешается использование онлайн компиляторов и справок по языку C#. Для построения блок-схем алгоритмов студенту можно использовать любой соответствующий онлайн-инструмент.

7. Методические материалы по освоению дисциплины

Для изучения основных вопросов образовательной программы необходимо конспектировать материалы лекций, работать с рекомендованной преподавателем литературой, а также ресурсами информационно-телекоммуникационной сети «Интернет». Для приобретения навыков активного использования знаний полезно обсуждать плановые и возникающие вопросы, а также решаемые задачи на практических занятиях. Чтобы легче и прочнее усвоить материал следует постоянно использовать конкретные примеры, сравнения из уже полученных областей наук.

Для закрепления изученного материала даны вопросы по каждой теме дисциплины, на которые следует самостоятельно найти ответы.

Важной составной частью учебного процесса в вузе являются практические занятия. Практические занятия проводятся главным образом по дисциплинам, требующим закрепления навыков решения задач. Все практические занятия проводятся в компьютерных классах с использованием ИСР *VisualStudio* среды *MSOffice*. Каждая работа должна быть защищена, т.е. студент должен ответить на вопросы преподавателя о ходе выполнения работы, а также на вопросы теоретического характера.

При подготовке к практическим занятиям необходимо проанализировать конспект лекции, ознакомиться с рекомендованной литературой по соответствующей теме, осуществить подготовку по рекомендованным в рабочей программе вопросам для обсуждения темы, выполнить домашнее задание (при необходимости).

Необходимо помнить, что на лекции обычно рассматривается не весь материал, а только его часть. Остальная его часть восполняется в процессе самостоятельной работы. В связи с этим работа с рекомендованной литературой обязательна. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение

практического приложения рассматриваемых теоретических вопросов. В процессе этой работы студент должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале.

В процессе подготовки к занятиям рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретается практика в изложении и разъяснении полученных знаний, развивается речь. При необходимости следует обращаться за консультацией к преподавателю (в том числе по электронной почте).

Планируя консультацию, необходимо хорошо продумать вопросы, которые требуют разъяснения. Заканчивать подготовку следует составлением плана (конспекта) по изучаемому материалу (вопросу). Это позволяет составить концентрированное, сжатое представление по изучаемым вопросам. Записи имеют первостепенное значение для самостоятельной работы студентов. Они помогают понять построение изучаемого материала, выделить основные положения, проследить их логику. Кроме того, ведение записей способствует превращению чтения в активный процесс, мобилизует, наряду со зрительной, и моторную память. Следует помнить: у студента, систематически ведущего записи, создается свой индивидуальный фонд методических материалов для быстрого повторения изученных вопросов, для мобилизации накопленных знаний. Особенно важны и полезны записи тогда, когда в них находят отражение мысли, возникшие при самостоятельной работе.

После изучения базовых тем курса проводится текущий контроль знаний студентов в виде опроса или письменного тестирования. Типовые тесты и задания по темам дисциплины приведены в специальном разделе данной рабочей программы.

Подготовка к текущему и промежуточному контролю предполагает изучение представленных вопросов к зачету, работу над тестами, представленными в данной рабочей программе, выполнение семестровой проектной работы по применению системного подхода и методов системного анализа к выбранной системе.

Работа в малых группах – это одна из самых популярных форм проведения занятий, так как она дает всем обучающимся (в том числе и стеснительным) возможность участвовать в работе, практиковать навыки сотрудничества, межличностного общения (в частности, умение активно слушать, вырабатывать общее мнение, разрешать возникающие разногласия). Цель данной формы проведения занятий: продемонстрировать сходство или различия определенных явлений, выработать стратегию или разработать план, выяснить отношение различных групп участников к одному и тому же вопросу. В ходе этой работы дополнительно решаются следующие задачи: развитие навыков общения и взаимодействия в группе, формирование ценностно-ориентационного единства

группы, поощрение к гибкой смене социальных ролей в зависимости от ситуации.

Группа студентов делится на несколько малых групп. Количество групп определяется числом творческих заданий, которые будут обсуждаться в процессе занятия. Малые группы формируются либо по желанию студентов, либо по родственной тематике для обсуждения. Каждая малая группа обсуждает творческое задание в течение отведенного времени. Основным этапом – проведение обсуждения творческого задания. Заслушиваются суждения, предлагаемые каждой малой группой по творческому заданию. Преподаватель дает оценочное суждение и работе малых групп, по решению творческих заданий, и эффективности предложенных путей решения.

С целью активизации самостоятельной работы студентов в системе дистанционного обучения Moodle разработан учебный курс «Программирование», включающий набор файлов с текстами лекций, заданиями для выполнения практических и лабораторных работ. Для активизации работы студентов во время контактной работы с преподавателем отдельные занятия проводятся в интерактивной форме. В основном, интерактивная форма занятий обеспечивается при проведении занятий в компьютерном классе. Интерактивная форма обеспечивается наличием разработанных файлов с заданиями, наличием контрольных вопросов, возможностью доступа к системе дистанционного обучения, а также к тестеру.

В качестве самостоятельной работы студентами выполняется семестровая работа по всем темам. При выполнении заданий по темам могут использоваться представленные студентом материалы по предыдущим темам. Выполненная семестровая работа представляется студентом на открытой защите на промежуточной аттестации.

8. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет

8.1. Основная литература

1. Албахари Д. С# 9.0 Справочник. Полное описание. Санкт-Петербург: Диалектика, 2021. 1056 с.
2. Прайс М. С# 10 и .NET 6. Современная кроссплатформенная разработка. Санкт-Петербург: Питер, , 2021. 848 с.
3. Казанский, Александр Анатольевич. Программирование на *Visual C#*: А. А. Казанский. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2024. — 192 с. — (Высшее образование). — ISBN 978-5-534-12338-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/537364> (дата обращения: 30.06.2024).

8.2. Дополнительная литература

1. Гниденко, Ирина Геннадиевна. Технологии и методы программирования: учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. - Москва: Юрайт, 2020. - 235 с. - (Высшее образование). - Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450999> (дата обращения: 24.09.2020). - ISBN 978-5-534-02816-4

2. Окулов, Станислав Михайлович. Основы программирования: учебное пособие / С.М. Окулов. - 10-е изд., электрон. - Москва: Лаборатория знаний, 2020. - 337 с.: ил. - Текст: электронный. - URL: <https://e.lanbook.com/book/135560> (дата обращения: 24.12.2020).

3. Кудрина, Евгения Вячеславовна. Основы алгоритмизации и программирования на языке C#. Учебное пособие для бакалавриата и специалитета – Москва: Издательство Юрайт, 2020, ISBN978-5-534-09796-2.

4. Гниденко, Ирина Геннадиевна. Технологии и методы программирования: учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. - Москва: Юрайт, 2020. - 235 с. - (Высшее образование). - Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450999> (дата обращения: 24.09.2020). - ISBN 978-5-534-02816-4.

8.3. Нормативные правовые документы и иная правовая информация

Не используются.

8.4 Интернет-ресурсы

Обучающимся обеспечен доступ к материалам курса в СДО Академии <http://lms.ranepa.ru>, а также через сайт научной библиотеки к следующим подписным электронным ресурсам:

Русскоязычные ресурсы

- Электронные учебники электронно-библиотечной системы (ЭБС) «Айбукс»
- Электронные учебники электронно-библиотечной системы (ЭБС) «Юрайт»
- Электронные учебники электронно-библиотечной системы (ЭБС) «Лань»
- Электронные учебники электронно-библиотечной системы (ЭБС) «ZNANIUM.COM»
- Электронные учебники электронно-библиотечной системы (ЭБС) «BOOK.RU»
- Электронные учебники электронно-библиотечной системы (ЭБС) «IPRSMART»

9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

№ п/п	Наименование
1.	Специализированные залы для проведения лекций, оснащенные персональным компьютером/ноутбуком и мультимедийным проектором
2.	Аудитории и компьютерные классы, оборудованные посадочными местами и персональными компьютерами с выходом в Интернет для проведения практических занятий. Программное обеспечение: <i>MSVisualStudio</i> , <i>MSVisualStudioCode</i>
3.	«МТС Линк» — российская платформа для онлайн-коммуникаций и совместной работы команд ; «Яндекс Телемост» — сервис для видеоконференций от Яндекса; Яндекс-мессенджер
4.	Технические средства обучения: персональные компьютеры; программные средства, обеспечивающие просмотр видеофайлов в форматах <i>AVI</i> , <i>MPEG-4</i> , <i>DivX</i> , <i>RMVB</i> , <i>WMV</i> ; программы для работы с электронными таблицами для обработки, анализа и визуализации данных; соответствующие онлайн-инструменты для построения интеллект-карты и моделей в различных нотациях
5.	Научная библиотека (в т.ч. электронные информационные ресурсы научной библиотеки)
6.	СДО Академии https://lms.ranepa.ru/