

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: КУЗНЕЦОВ АЛЕКСАНДР ИГОРЕВИЧ  
Должность: РЕКТОР  
Дата подписания: 02.02.2026 13:48:55  
Уникальный программный ключ:  
0ec0d544ced914f6d2e031d381fc0ed0880d90a0



МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ  
ГУМАНИТАРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ЮУГГПУ»)

РАБОЧАЯ ПРОГРАММА

Шифр	Наименование дисциплины (модуля)
Б1.О	Технологии программирования

Код направления подготовки	09.03.02
Направление подготовки	Информационные системы и технологии
Наименование (я) ОПОП (направленность / профиль)	Информационные технологии в образовании
Уровень образования	бакалавр
Форма обучения	очная

Разработчики:

Должность	Учёная степень, звание	Подпись	ФИО
Старший преподаватель			Боровская Елена Владимировна

Рабочая программа рассмотрена и одобрена (обновлена) на заседании кафедры (структурного подразделения)

Кафедра	Заведующий кафедрой	Номер протокола	Дата протокола	Подпись
Кафедра математики и информатики	Звягин Константин Алексеевич	3	23.11.2025г.	

## ОГЛАВЛЕНИЕ

1. Пояснительная записка .....	3
2. Трудоемкость дисциплины (модуля) и видов занятий по дисциплине (модулю) .....	5
3. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий .....	6
4. Учебно-методическое и информационное обеспечение дисциплины .....	14
5. Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю) .....	15
6. Методические указания для обучающихся по освоению дисциплины .....	22
7. Перечень образовательных технологий .....	24
8. Описание материально-технической базы .....	25

# 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1 Дисциплина «Технологии программирования» относится к модулю обязательной части Блока 1 «Дисциплины/модули» основной профессиональной образовательной программы по направлению подготовки 09.03.02 «Информационные системы и технологии» (уровень образования бакалавр). Дисциплина является обязательной к изучению.

1.2 Общая трудоемкость дисциплины составляет 5 з.е., 180 час.

1.3 Изучение дисциплины «Технологии программирования» основано на знаниях, умениях и навыках, полученных при изучении обучающимися следующих дисциплин: «Информатика», «Программирование».

1.4 Дисциплина «Технологии программирования» формирует знания, умения и компетенции, необходимые для освоения следующих дисциплин: «Алгоритмы и структуры данных», «Инструментальные средства разработки образовательных технологий», «Инструментальные средства информационных систем», «Методы и средства проектирования информационных систем и технологий», «Разработка мобильных приложений».

1.5 Цель изучения дисциплины:

знакомство студентов с теоретическими основами современных технологий программирования и применением этих технологий для решения практических задач.

1.6 Задачи дисциплины:

- 1) изучение основных подходов к организации процесса разработки программного обеспечения;
- 2) изучение базовых структур данных;
- 3) изучение основ процедурного, модульного и объектно-ориентированного программирования.

1.7 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы:

№ п/п	Код и наименование компетенции по ФГОС
Код и наименование индикатора достижения компетенции	
1	ОПК-6 способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий ОПК.6.1 Знать методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий ОПК.6.2 Уметь применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий ОПК.6.3 Иметь навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач
2	ОПК-3 способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности ОПК.3.1 Знать принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности ОПК.3.2 Уметь решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности. ОПК.3.3 Иметь навыки подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.

№ п/п	Код и наименование индикатора достижения компетенции	Образовательные результаты по дисциплине
1	ОПК.6.1 Знать методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий	3.2 Знать методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий
2	ОПК.6.2 Уметь применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий	У.2 Уметь применять методы алгоритмизации, языки и технологии программирования при разработке ПО

3	ОПК.6.3 Иметь навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач	B.2 Иметь навыки программирования, отладки и тестирования ПО
1	ОПК.3.1 Знать принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	3.1 принципы, базовые концепции технологий программирования
2	ОПК.3.2 Уметь решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.	У.1 использовать технологии программирования при разработке систем автоматизации проектирования прикладных программ;
3	ОПК.3.3 Иметь навыки подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.	B.1 навыками разработки проектной документации на ПО

**2. ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ (МОДУЛЯ) И ВИДОВ ЗАНЯТИЙ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Наименование раздела дисциплины (темы)	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Итого часов
	Л	ЛЗ	СРС	
<b>Итого по дисциплине</b>	<b>24</b>	<b>40</b>	<b>80</b>	<b>144</b>
<b>Первый период контроля</b>				
<i>Основные понятия технологий программирования.</i>	<b>10</b>		<b>10</b>	<b>20</b>
Основные понятия технологий программирования	2		2	4
Этапы развития технологий программирования	2		2	4
Жизненный цикл ПО	2		2	4
Качество ПО	2		2	4
Семантический подход к ЯП	2		2	4
<i>Реализация принципов ООП в C#. Визуальное программирование.</i>	<b>14</b>	<b>40</b>	<b>70</b>	<b>124</b>
Парадигмы программирования. ООП и объекты реального мира.	2		4	6
Инкапсуляция	2		2	4
Наследование	2		2	4
Полиморфизм	2		2	4
Интерфейсы	2		2	4
Основы визуального проектирования приложений. Приложения Windows Forms	2		2	4
Создание приложений для работы с данными	2		2	4
Интегрированная среда для визуального проектирования приложений		2	2	4
Просмотр рисунков		2	2	4
Лабиринт		2	2	4
Викторина		2	2	4
Найди пару		2	2	4
Графика		2	2	4
Построим город		2	2	4
Основные свойства и компоненты		4	10	14
Динамическое создание компонентов		4	10	14
Создание БД		2	2	4
Создание приложения для работы с БД		4	2	6
Работа со связанными таблицами БД		4	4	8
Работа с запросами		4	6	10
Создание отчетов		4	6	10
Итого по видам учебной работы	<b>24</b>	<b>40</b>	<b>80</b>	<b>144</b>
<i>Форма промежуточной аттестации</i>				
Экзамен				36
<b>Итого за Первый период контроля</b>				<b>180</b>
<b>Второй период контроля</b>				
Итого по видам учебной работы				
<i>Форма промежуточной аттестации</i>				
Курсовая работа				
<b>Итого за Второй период контроля</b>				

**3. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ), СТРУКТУРИРОВАННОЕ ПО ТЕМАМ (РАЗДЕЛАМ) С УКАЗАНИЕМ ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ И ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ**

**3.1 Лекции**

Наименование раздела дисциплины (модуля)/ Тема и содержание	Трудоемкость (кол-во часов)
<b>1. Основные понятия технологий программирования.</b>  <b>Формируемые компетенции, образовательные результаты:</b> ОПК-3: 3.1 (ОПК.3.1), У.1 (ОПК.3.2), В.1 (ОПК.3.3) ОПК-6: 3.2 (ОПК.6.1), У.2 (ОПК.6.2), В.2 (ОПК.6.3)	<b>10</b>
1.1. Основные понятия технологий программирования 1. Программная инженерия 2. Понятие технологии программирования 3. Методы 4. Процедуры 5. Средства 6. CASE средства Учебно-методическая литература: 1, 3, 6	2
1.2. Этапы развития технологий программирования 1. Архитектура программы 2. 1 этап – отсутствие технологии 3. 2 этап – структурное программирование. Метод пошаговой детализации. 4. 3 этап – модульное программирование 5. 4 этап – объектно-ориентированное программирование 6. Компонентное программирование Учебно-методическая литература: 1, 2, 3, 5	2
1.3. Жизненный цикл ПО Понятие жизненного цикла ПО 2. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств 3. Основные термины и определения 4. Категории процессов жизненного цикла 5. Описания процессов ЖЦ 6. Модели жизненного цикла программных средств Учебно-методическая литература: 1, 2, 3	2
1.4. Качество ПО 1. Понятие качества ПО 2. Качество ПО в истории развития ЯП 3. ГОСТ Р ИСО/МЭК 9126 Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению. 4. Основные термины и определения 5. Характеристики качества ПО и атрибуты качества. 6. Примеры метрик качества ПО. Учебно-методическая литература: 1, 2, 3	2
1.5. Семантический подход к ЯП 1. История развития ЯП 2. Система программирования 3. Основные понятия ЯП 4. Средства описания данных и действий (алгоритма) 5. Объекты данных. Типы данных. 6. Выражения и операторы. Учебно-методическая литература: 1, 2, 3	2
<b>2. Реализация принципов ООП в C#. Визуальное программирование.</b>  <b>Формируемые компетенции, образовательные результаты:</b> ОПК-3: 3.1 (ОПК.3.1), У.1 (ОПК.3.2), В.1 (ОПК.3.3) ОПК-6: У.2 (ОПК.6.2), 3.2 (ОПК.6.1), В.2 (ОПК.6.3)	<b>14</b>

<p>2.1. Парадигмы программирования. ООП и объекты реального мира.</p> <ol style="list-style-type: none"> <li>1. Парадигма программирования на машинном языке</li> <li>2. Парадигма процедурного программирования</li> <li>3. Парадигма модульного программирования</li> <li>4. Парадигма Объектно-Ориентированного Программирования</li> <li>5. Цели применения ООП в разработке программного обеспечения</li> <li>6. Надежность ПО</li> <li>7. Повторное использование</li> <li>8. Способность к расширению</li> <li>9. Периодический выпуск (издание) новых версий</li> <li>10. Другие парадигмы программирования</li> </ol> <p>2. Абстрактные объекты и экземпляры</p> <p>3. Наследование</p> <p>4. Зачем использовать наследование</p> <p>5. Объекты в деловом мире</p> <p>6. Объекты реального мира</p> <p>и объектно-ориентированное программирование</p> <p>Учебно-методическая литература: 1, 2, 3, 6</p>	2
<p>2.2. Инкапсуляция</p> <ol style="list-style-type: none"> <li>1. Понятие инкапсуляции</li> <li>2. Интерфейс</li> <li>3. Реализация</li> <li>4. Пример применения концепции отделения интерфейса от реализации</li> <li>5. Модификаторы доступа</li> <li>6. Инкапсуляция при помощи традиционных методов доступа и изменения</li> <li>7. Инкапсуляция при помощи применение свойств класса</li> </ol> <p>Учебно-методическая литература: 1, 2, 3, 6</p>	2
<p>2.3. Наследование</p> <ol style="list-style-type: none"> <li>1. Понятие «наследование»</li> <li>2. Наследственная иерархия</li> <li>3. Потомственный класс</li> <li>4. Родительский класс</li> <li>5. Пример реализации наследования</li> </ol> <p>Учебно-методическая литература: 1, 2, 3, 4, 5</p>	2
<p>2.4. Полиморфизм</p> <ol style="list-style-type: none"> <li>1. Понятие полиморфизма</li> <li>2. Формы полиморфизма <ul style="list-style-type: none"> <li>• полиморфизм включения;</li> <li>• переопределение метода;</li> <li>• перегрузка метода.</li> </ul> </li> <li>3. Примеры реализации полиморфизма</li> </ol> <p>Учебно-методическая литература: 1, 2, 3, 4, 6</p>	2
<p>2.5. Интерфейсы</p> <ol style="list-style-type: none"> <li>1. Интерфейс - частный случай класса.</li> <li>2. Пример интерфейса</li> <li>3. Стандартные интерфейсы C# <ul style="list-style-type: none"> <li>• ICloneable</li> <li>• IComponent</li> <li>• IContainer</li> <li>• Form</li> <li>• INumerator</li> <li>• ISerializable</li> <li>• ICollection</li> <li>• IDictionary</li> <li>• IComparable</li> </ul> </li> <li>4. Примеры работы со стандартными интерфейсами</li> </ol> <p>Учебно-методическая литература: 1, 2, 3, 6</p>	2

<p>2.6. Основы визуального проектирования приложений.¶Приложения Windows Forms</p> <ol style="list-style-type: none"> <li>1) Программная модель Windows Forms – основа для разработки приложений .NET Framework с графическим интерфейсом пользователя</li> <li>2) Классовые операции IS и AS</li> <li>3) Как Windows взаимодействует с пользователем</li> <li>4) Windows-формы</li> <li>5) Важные свойства, события, методы класса Control</li> <li>6) Видеоурок по правильному дизайну приложений</li> </ol> <p>Учебно-методическая литература: 1, 2, 3, 6</p>	2
<p>2.7. Создание приложений для работы с данными</p> <ol style="list-style-type: none"> <li>1. Основы ADO .NET</li> <li>2. Принцип единобразной работы с базами данных</li> <li>3. ADO .NET. Объектная модель</li> <li>4. объект DataTable</li> <li>5. свойство Relations</li> <li>6. класс Constraint</li> <li>7. объект DataSet</li> <li>8. MVC (Model-View-Controller)</li> </ol> <p>Учебно-методическая литература: 1, 2, 3, 6</p>	2

### 3.2 Лабораторные

Наименование раздела дисциплины (модуля)/ Тема и содержание	Трудоемкость (кол-во часов)
<b>1. Реализация принципов ООП в C#. Визуальное программирование.</b>	<b>40</b>
<b>Формируемые компетенции, образовательные результаты:</b>	
ОПК-3: 3.1 (ОПК.3.1), У.1 (ОПК.3.2), В.1 (ОПК.3.3) ОПК-6: У.2 (ОПК.6.2), З.2 (ОПК.6.1), В.2 (ОПК.6.3)	
<p>1.1. Интегрированная среда для визуального проектирования¶приложений</p> <ol style="list-style-type: none"> <li>1. Создание проекта Windows Forms</li> <li>2. Окно Свойств</li> <li>3. Окно Toolbox</li> <li>4. Режимы дизайна и кода</li> <li>5. События мыши</li> </ol> <p>Учебно-методическая литература: 1, 5</p>	2
<p>1.2. Просмотр рисунков</p> <ol style="list-style-type: none"> <li>1. Создание приложения Windows Forms</li> <li>2. Изменение внешнего вида формы</li> <li>3. Добавление в форму элементов управления TableLayoutPanel, PictureBox, CheckBox, кнопок.</li> <li>4. Добавление в форму компонентов OpenFileDialog и ColorDialog.</li> <li>5. Создание кода с помощью средства IntelliSense.</li> <li>6. Просмотр и тестирование кода. Добавление необходимых комментариев.</li> <li>7. Создание кода с помощью IntelliSense для описания поведения кнопок и флагков.</li> <li>8. Запуск программы и настройка цвета фона. Изучение других возможностей.</li> </ol> <p>Учебно-методическая литература: 1, 6</p>	2
<p>1.3. Лабиринт</p> <ol style="list-style-type: none"> <li>1. Создание проекта и добавление элемента управления Panel.</li> <li>2. Создание лабиринт с помощью элементов управления Label.</li> <li>3. Создание конца игры.</li> <li>4. Добавление обработчика событий для отправки указателя мыши назад к старту, если указатель касается стены.</li> <li>5. В игру "Лабиринт" добавляются звуки.</li> <li>6. Выполняется запуск программы и проверка ее работы. Изучаются другие возможности, например, изменение цветов и звуков.</li> </ol> <p>Учебно-методическая литература: 1, 6</p>	2

<p>1.4. Викторина</p> <ol style="list-style-type: none"> <li>1. Создание проекта и добавление в форму элементов управления</li> <li>2. Создание задачи на сложение случайных чисел</li> <li>3. Добавление таймера с обратным отсчетом</li> <li>4. Добавление метода CheckTheAnswer()</li> <li>5. Добавление обработчиков событий входа для элементов управления NumericUpDown</li> <li>6. Добавление задачи на вычитание</li> <li>7. Добавление задач на умножение и деление</li> <li>8. Изучение других функций</li> </ol> <p>Учебно-методическая литература: 1, 6</p>	2
<p>1.5. Найди пару</p> <ol style="list-style-type: none"> <li>1. Создание проекта и добавление таблицы в форму</li> <li>2. Добавление случайного объекта и списка значков</li> <li>3. Назначение каждой метке случайного значка. Добавление к каждой метке обработчика событий щелчка мышью</li> <li>4. Добавление таймера</li> <li>5. Отмена исчезновения пар значков</li> <li>6. Добавление метода для проверки, выиграл ли игрок</li> <li>7. Изучение других функций</li> </ol> <p>Учебно-методическая литература: 1, 2, 6</p>	2
<p>1.6. Графика</p> <ol style="list-style-type: none"> <li>1) Кисти и краски</li> <li>2) Паутина Безье</li> <li>3) Масштабируемое изображение</li> <li>4) Градиентные заливки</li> <li>5) Преобразования</li> <li>6) Создание анимации</li> </ol> <p>Учебно-методическая литература: 1, 2, 5</p>	2
<p>1.7. Построим город</p> <ol style="list-style-type: none"> <li>1) Создание проекта и добавление в форму элементов управления</li> <li>2) Описание собственного класса для хранения изображений</li> <li>3) Реализация рисования объектов города</li> <li>4) Реализация рисования асфальта</li> <li>5) Реализация отмены последнего действия</li> </ol> <p>Учебно-методическая литература: 1, 2, 6</p>	2
<p>1.8. Основные свойства и компоненты</p> <ol style="list-style-type: none"> <li>1) работа с цветом формы</li> <li>2) Многооконное приложение</li> <li>3) События мыши</li> <li>4) События клавиатуры</li> <li>5) Работа с компонентами <ul style="list-style-type: none"> <li>• Метки</li> <li>• TextBox</li> <li>• RichTextBox</li> <li>• CheckListBox</li> <li>• NumericUpDown</li> <li>• MaskedTextBox</li> <li>• ToolTip</li> <li>• ProgressBar</li> </ul> </li> </ol> <p>Учебно-методическая литература: 1, 2, 5</p>	4
<p>1.9. Динамическое создание компонентов</p> <p>Динамическое создание компонентов в момент работы приложения:</p> <ol style="list-style-type: none"> <li>1) DateTimePicker</li> <li>2) MonthCalendar</li> <li>3) Форматы представления времени</li> <li>4) Вычисление промежутков</li> <li>5) Определение даты</li> </ol> <p>Учебно-методическая литература: 1, 2, 6, 7</p>	4

1.10. Создание БД 1) Создание БД. 2) Подключение к БД 3) Изменение структуры БД с помощью DataSet Designer. Учебно-методическая литература: 1, 2, 5, 6	2
1.11. Создание приложения для работы с БД 1) Подключение к БД 2) Разработка интерфейса приложения, для работы с БД на основе таблиц и списков полей. 3) Различные способы визуализации значений полей БД Учебно-методическая литература: 1, 2, 4, 5, 6	4
1.12. Работа со связанными таблицами БД 1. Создание интерфейса приложения на основе двух взаимосвязанных таблиц 2. Проверка каскадной целостности БД 3. Работа с ограничениями Учебно-методическая литература: 1, 2, 5, 6	4
1.13. Работа с запросами 1) Выборка строк по некоторому условию. 2) Поиск записи с максимальным значением 3) Создание запросов на извлечение данных из 2-х таблиц. 4) Удаление записей из БД. Учебно-методическая литература: 1, 2, 6	4
1.14. Создание отчетов 1) Генератор отчетов 2) Отчеты на основе запросов 3) Отчеты на основе запросов к 2 таблицам Учебно-методическая литература: 1, 2, 6	4

### 3.3 СРС

Наименование раздела дисциплины (модуля)/ Тема для самостоятельного изучения	Трудоемкость (кол-во часов)
<b>1. Основные понятия технологий программирования.</b>	<b>10</b>
<b>Формируемые компетенции, образовательные результаты:</b>	
ОПК-3: 3.1 (ОПК.3.1), У.1 (ОПК.3.2), В.1 (ОПК.3.3) ОПК-6: 3.2 (ОПК.6.1), У.2 (ОПК.6.2), В.2 (ОПК.6.3)	
1.1. Основные понятия технологий программирования <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 2, 3	2
1.2. Этапы развития технологий программирования <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 2, 3	2
1.3. Жизненный цикл ПО <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 2, 3	2
1.4. Качество ПО <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 2, 3	2
1.5. Семантический подход к ЯП <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 2, 3	2
<b>2. Реализация принципов ООП в C#. Визуальное программирование.</b>	<b>70</b>
<b>Формируемые компетенции, образовательные результаты:</b>	
ОПК-3: 3.1 (ОПК.3.1), У.1 (ОПК.3.2), В.1 (ОПК.3.3) ОПК-6: У.2 (ОПК.6.2), 3.2 (ОПК.6.1), В.2 (ОПК.6.3)	

2.1. Парадигмы программирования. ООП и объекты реального мира. <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 2, 3	4
2.2. Инкапсуляция <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 2, 3, 6	2
2.3. Наследование <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 2, 3, 6	2
2.4. Полиморфизм <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 2, 3, 6	2
2.5. Интерфейсы <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 2, 3, 6	2
2.6. Основы визуального проектирования приложений. Windows Forms <b>Задание для самостоятельного выполнения студентом:</b> Написать программы для решения задач с графическим пользовательским интерфейсом Вариативная часть Решение задач повышенной сложности Учебно-методическая литература: 1, 2, 3, 6	2
2.7. Создание приложений для работы с данными <b>Задание для самостоятельного выполнения студентом:</b> Написать программы для решения задач с графическим пользовательским интерфейсом Вариативная часть Решение задач повышенной сложности Учебно-методическая литература: 1, 2, 6	2
2.8. Интегрированная среда для визуального проектирования приложений <b>Задание для самостоятельного выполнения студентом:</b> Написать программы для решения задач с графическим пользовательским интерфейсом Вариативная часть Решение задач повышенной сложности Учебно-методическая литература: 1, 2, 6	2
2.9. Просмотр рисунков <b>Задание для самостоятельного выполнения студентом:</b> Написать программы для решения задач с графическим пользовательским интерфейсом Вариативная часть Решение задач повышенной сложности Учебно-методическая литература: 1, 6	2
2.10. Лабиринт <b>Задание для самостоятельного выполнения студентом:</b> Написать программы для решения задач с графическим пользовательским интерфейсом Вариативная часть Решение задач повышенной сложности Учебно-методическая литература: 1, 6	2

2.11. Викторина		2
<p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 6</p>		
2.12. Найди пару		2
<p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 6</p>		
2.13. Графика		2
<p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 6</p>		
2.14. Построим город		2
<p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 6</p>		
2.15. Основные свойства и компоненты		10
<p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 5</p>		
2.16. Динамическое создание компонентов		10
<p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 5</p>		
2.17. Создание БД		2
<p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом и БД</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 5, 6</p>		
2.18. Создание приложения для работы с БД		2
<p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом и БД</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 2, 6</p>		

<p>2.19. Работа со связанными таблицами БД</p> <p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом и БД</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 2, 6</p>	4
<p>2.20. Работа с запросами</p> <p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом и БД</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 2, 6</p>	6
<p>2.21. Создание отчетов</p> <p><b>Задание для самостоятельного выполнения студентом:</b></p> <p>Написать программы для решения задач с графическим пользовательским интерфейсом и БД</p> <p>Вариативная часть</p> <p>Решение задач повышенной сложности</p> <p>Учебно-методическая литература: 1, 2, 6</p>	6
<p><b>3. Курсовая работа</b></p> <p>См. пункт 5.2.2</p>	18 часов из трудоемкости СРС

## 4. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 4.1. Учебно-методическая литература

№ п/п	Библиографическое описание (автор, заглавие, вид издания, место, издательство, год издания, количество страниц)	Ссылка на источник в ЭБС
<b>Основная литература</b>		
1	Горелов, С. В. Современные технологии программирования: разработка Windows-приложений на языке C#. В 2 томах. Т. I : учебник / С. В. Горелов ; под редакцией П. Б. Лукьянова. — Москва : Прометей, 2019. — 362 с. — ISBN 978-5-907100-09-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <a href="http://www.iprbookshop.ru/94532.html">http://www.iprbookshop.ru/94532.html</a> (дата обращения: 03.06.2020). — Режим доступа: для авторизир. пользователей	URL: <a href="http://www.iprbookshop.ru/94532.html">http://www.iprbookshop.ru/94532.html</a>
2	Горелов, С. В. Современные технологии программирования: разработка Windows-приложений на языке C#. В 2 томах. Т. II : учебник / С. В. Горелов ; под редакцией П. Б. Лукьянова. — Москва : Прометей, 2019. — 378 с. — ISBN 978-5-907100-18-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <a href="http://www.iprbookshop.ru/94533.html">http://www.iprbookshop.ru/94533.html</a> (дата обращения: 03.06.2020). — Режим доступа: для авторизир. пользователей	URL: <a href="http://www.iprbookshop.ru/94533.html">http://www.iprbookshop.ru/94533.html</a>
3	Давыдова, Н. А. Программирование : учебное пособие / Н. А. Давыдова, Е. В. Боровская. — 4-е изд. — Москва : Лаборатория знаний, 2020. — 239 с. — ISBN 978-5-00101-788-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <a href="http://www.iprbookshop.ru/6485.html">http://www.iprbookshop.ru/6485.html</a> (дата обращения: 03.06.2020). — Режим доступа: для авторизир. пользователей	<a href="http://www.iprbookshop.ru/6485.html">http://www.iprbookshop.ru/6485.html</a>
<b>Дополнительная литература</b>		
4	Терехов, А. Н. Технология программирования : учебное пособие / А. Н. Терехов. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 152 с. — ISBN 978-5-4487-0070-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <a href="http://www.iprbookshop.ru/67370.html">http://www.iprbookshop.ru/67370.html</a> (дата обращения: 03.06.2020). — Режим доступа: для авторизир. пользователей	URL: <a href="http://www.iprbookshop.ru/67370.html">http://www.iprbookshop.ru/67370.html</a>
5	Лебедева, Т. Н. Технология программирования : учебное пособие / Т. Н. Лебедева, С. С. Юнусова. — 2-е изд. — Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. — 140 с. — ISBN 978-5-4486-0664-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <a href="http://www.iprbookshop.ru/81500.html">http://www.iprbookshop.ru/81500.html</a> (дата обращения: 03.06.2020). — Режим доступа: для авторизир. пользователей	URL: <a href="http://www.iprbookshop.ru/81500.html">http://www.iprbookshop.ru/81500.html</a>
6	Сайт <a href="https://docs.microsoft.com">docs.microsoft.com</a> — это хранилище документации Майкрософт для пользователей, разработчиков и ИТ-специалистов.	<a href="https://docs.microsoft.com/ru-ru/">https://docs.microsoft.com/ru-ru/</a>
7	Златопольский, Д. М. Программирование: типовые задачи, алгоритмы, методы / Д. М. Златопольский. — 4-е изд. — Москва : Лаборатория знаний, 2020. — 224 с. — ISBN 978-5-00101-789-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <a href="http://www.iprbookshop.ru/12264.html">http://www.iprbookshop.ru/12264.html</a> (дата обращения: 03.06.2020). — Режим доступа: для авторизир. пользователей	<a href="http://www.iprbookshop.ru/12264.html">http://www.iprbookshop.ru/12264.html</a>

## 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

### 5.1. Описание показателей и критерии оценивания компетенций

Код компетенции по ФГОС			
Код образовательного результата дисциплины	Текущий контроль		Помежуточная аттестация
	Тест	Задача	
ОПК-3			
3.1 (ОПК.3.1)	+		+
У.1 (ОПК.3.2)	+	+	+
В.1 (ОПК.3.3)		+	+
ОПК-6			
3.2 (ОПК.6.1)	+		+
У.2 (ОПК.6.2)	+	+	+
В.2 (ОПК.6.3)		+	+

### 5.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

#### 5.2.1. Текущий контроль.

Типовые задания к разделу "Основные понятия технологий программирования.":

##### 1. Задача

Создать 2 формы для игры или ЦОР по любой теме школьного курса:

Анимационная заставка

Поле для настольной игры.

Формы должна быть «резиновыми». Для рисования поля использовать кисти с рисунками. Для рисования фишек или фигур различные градиентные заливки. В анимации должно участвовать не менее 3 различных фигур.

База данных хроники восхождений в альпинистском клубе.

В базе данных должны записываться даты начала и завершения каждого восхождения, имена и адреса участвовавших в нем альпинистов, название и высота горы, страна и район, где эта гора расположена. Дайте выразительные имена таблицам и полям, в которые могла бы заноситься указанная информация. Написать пакет, состоящий из процедур и функций, которые позволили бы выполнить следующие действия с базой данных:

- 1) для каждой горы показать список групп, осуществлявших восхождение, в хронологическом порядке;
  - 2) предоставить возможность добавления новой вершины, с указанием названия вершины, высоты и страны местоположения;
  - 3) предоставить возможность изменения данных о вершине, если на нее не было восхождения;
  - 4) показать список альпинистов, осуществлявших восхождение в указанный интервал дат;
- 3.1. Варианты заданий
- 5) предоставить возможность добавления нового альпиниста в состав указанной группы;
  - 6) показать информацию о количестве восхождений каждого альпиниста на каждую гору;
  - 7) показать список восхождений (групп), которые осуществлялись в указанный пользователем период времени;
  - 8) предоставить возможность добавления новой группы, указав ее название, вершину, время начала восхождения;
  - 9) предоставить информацию о том, сколько альпинистов побывали на каждой горе.

Количество баллов: 100

## 2. Тест

#2/

Под семантикой языка программирования понимают

§

правила, определяющие какие операции, и в какой последовательности должна выполнять ЭВМ, работающая по программе.

набор символов, включающий буквы, цифры и специальные знаки (знаки операций, пунктуации и т.д.

множество текстов некоторого алфавита, удовлетворяющих правилам синтаксиса и задающих порядок вычислений в соответствии с правилами семантики.

совокупность правил записи, которым должна удовлетворять программа, включает также правила ввода текстов программ в ЭВМ.

#2/

Семантика языка программирования задается

§

определенением средств описания данных и действий (алгоритм).

определенением правил записи текстов программ на языке программирования

определенением набора символов, входящих в язык программирования

#2/

Базовые средства описания данных включают в себя:

§

Простые и структурированные типы данных

Выражения, операторы действий, операторы управления, подпрограммы

Абстрактные типы данных

Обработку ошибок, работу с файлами, параллельную обработку

#2/

Базовые средства описания действий включают в себя:

§

Выражения, операторы действий, операторы управления, подпрограммы

Простые и структурированные типы данных

Абстрактные типы данных

Обработку ошибок, работу с файлами, параллельную обработку

Количество баллов: 50

Типовые задания к разделу "Реализация принципов ООП в C#. Визуальное программирование.":

### 1. Задача

Анимационная заставка

Поле для настольной игры.

Формы должна быть «резиновыми». Для рисования поля использовать кисти с рисунками. Для рисования фишек или фигур различные градиентные заливки. В анимации должно участвовать не менее 3 различных фигур.

База данных хроники восхождений в альпинистском клубе.

В базе данных должны записываться даты начала и завершения каждого восхождения, имена и адреса участвовавших в нем альпинистов, название и высота горы, страна и район, где эта гора расположена. Дайте выразительные имена таблицам и полям, в которые могла бы заноситься указанная информация. Написать пакет, состоящий из процедур и функций, которые позволил бы выполнить следующие действия с базой данных:

1) для каждой горы показать список групп, осуществлявших

восхождение, в хронологическом порядке;

2) предоставить возможность добавления новой вершины, с

указанием названия вершины, высоты и страны местоположения;

3) предоставить возможность изменения данных о вершине, если на нее не было восхождения;

4) показать список альпинистов, осуществлявших восхождение

в указанный интервал дат;

3.1. Варианты заданий

5) предоставить возможность добавления нового альпиниста в состав указанной группы;

6) показать информацию о количестве восхождений каждого

альпиниста на каждую гору;

7) показать список восхождений (групп), которые осуществлялись в указанный пользователем период времени;

8) предоставить возможность добавления новой группы, указав

ее название, вершину, время начала восхождения;

9) предоставить информацию о том, сколько альпинистов побывали на каждой горе.

Предусмотреть разработку триггер

Количество баллов: 100

## 2. Тест

Что будет напечатано на экране, после выполнения этого кода?

```
int f(int x)
{
    return x+1;
}
static void Main(string[] args)
{
    int x = 3;
    int y = 6;
    Console.WriteLine(f(x));
}
```

8. Что будет напечатано на экране, после выполнения этого кода?

```
int foo(int x, int y)
{
    return x+y;
}
static void Main(string[] args)
{
    double p=2,q=1;
    int s;
    s = foo (p,q);
    Console.WriteLine(s);
}
```

9. Что будет напечатано на экране, после выполнения этого кода?

```
int foo(int x, ref int y)
{
    X++;
    Y++;
    return x+y;
}
static void Main(string[] args)
{
    int x = 3;
    int y = 2;
    int s = foo (x,ref y);
    Console.WriteLine(s);
    Console.WriteLine(x);
    Console.WriteLine(y);
}
```

Количество баллов: 50

### 5.2.2. Промежуточная аттестация

Промежуточная аттестация проводится в соответствии с Положением о текущем контроле и промежуточной аттестации в ФГБОУ ВО «ЮУрГГПУ».

#### Первый период контроля

##### 1. Экзамен

Вопросы к экзамену:

1. the common language runtime (CLR)
2. the .NET Framework class library (.NET FCL).
3. Среда Времени Выполнения или Виртуальная Машина.
4. .NET Framework Class Library
5. MSIL (Microsoft Intermediate Language, он же IL – Intermedia Language) – промежуточный язык
6. JIT-компилятор
7. CTS - Common Type System Стандартная Система Типов.
8. Сборка мусора
9. Пространство имён
10. Структура проекта консольного приложения
11. Типы данных языка. Типы значений и ссылочные типы
12. Какие идентификаторы и диапазоны значений имеют числовые типы данных?
13. Какие операции можно осуществлять над переменными целого и вещественного типов?

14. Каков формат записи операции присваивания?
15. Какие встроенные функции существуют для работы с переменными целых и вещественных типов? Что необходимо сделать, чтобы использовать их в программе?
16. Как выполняется преобразование типов в выражениях?
17. Как выполняется ввод и вывод данных на экран в C#?
18. Каков приоритет выполнения операций в C#?
19. Что представляет собой логический тип данных в языке C#?
20. Как записываются логические операции в C#?
21. Как записывается условный оператор в языке C#?
22. Каковы правила записи составных условий в C#?
23. Какие виды циклов определены в языке C#?
24. Для каждого вида цикла ответьте на следующие вопросы:
25. В каких ситуациях удобно использовать тот или иной оператор цикла?
26. Какова структура цикла?
27. Объясните принцип работы цикла. Каково условие выхода из цикла?
28. Как записывается оператор выбора в языке C#?
29. Какой идентификатор имеет перечисляемый тип в языке C#? Как он записывается?
30. Что может быть значением в перечисляемом типе?
31. Класс Random и его функции
32. Класс Array: методы и свойства (из справки в лабораторной работе и примеров лекций)
33. Создание линейного массива, работа с элементами массива.
34. Цикл foreach
35. Примеры программ на линейные массивы и матрицы (типа задач ЕГЭ, примеры <http://inf.reshuege.ru/test?theme=284>)
36. Матрицы (примеры тестов в лабораторной работе и из примеров выше)
37. Класс string: методы и свойства (из справки в лабораторной работе и примеров лекций)
38. Объявление класса
39. Методы. Параметры методов (Примеры тестов в лабораторной работе)
40. Что такое процедурное программирование?
41. Какими преимуществами обладает процедурное программирование по сравнению с неструктурным программированием?
42. Что такое модульное программирование?
43. Какими преимуществами обладает модульное программирование по сравнению с процедурным программированием?
44. Перечислите недостатки процедурного и модульного программирования.
45. Что такое объектно-ориентированное программирование?
46. Каковы шесть преимуществ и целей объектно-ориентированного программирования?
47. Объясните одну из целей объектно-ориентированного программирования.
48. Дайте определения следующих терминов: класс и объект
49. Как объекты обмениваются информацией?
50. Что такое конструктор?
51. Что такое средство доступа?
52. Что такое this?
53. Что такое переменная экземпляра?
54. Как определить размер класса?
55. Объясните шаги, необходимые для создания экземпляра класса.
56. Как объявляются переменные экземпляра?
57. Что такое тип данных?
58. Объясните части определения метода.
59. Какая разница между списком аргументов и списком параметров?
60. Объясните шаги, необходимые для вызова метода-члена.
61. Что такое конструктор и как он определяется?
62. Как обращаться к переменным экземпляра из программы?
63. Что такое инкапсуляция?
64. Каковы преимущества использования инкапсуляции?
65. Что такое спецификатор(модификатор) доступа?
66. Что такое спецификатор доступа public?
67. Что такое спецификатор доступа private?

68. Что такое спецификатор доступа `protected`?
69. Какие части базового класса могут использоваться производным классом?
70. Почему программисты требуют, чтобы доступ к некоторым атрибутам класса осуществлялся только с помощью методов-членов?
71. Каким образом использование инкапсуляции помогает достичь целей объектно-ориентированного программирования?
72. Дайте определение понятию "абстракция" и приведите пример применения абстракции.
73. Дайте определение понятию "реализация".
74. Дайте определение понятию "интерфейс".
75. Объясните разницу между интерфейсом и реализацией.
76. Почему для достижения эффективной инкапсуляции важно четко распределить ответственность?
77. Определите понятие типа.
78. Что такое абстрактный тип данных?
79. Назовите два типа конструкторов.
80. Какие недостатки имеет простое многократное использование?
81. Что такое наследование?
82. Назовите три формы наследования.
83. В чем состоит опасность, когда применяется наследование?
84. Что такое программирование отличий?
85. Когда класс-наследник может иметь три типа методов и свойств? Что это за типы? В чем состоит выгода от применения программирования отличий?
86. Какие классы являются потомками класса `Permission` (пример в лекциях)?
87. Корневой класс и классы-листья.
88. Что такое наследование для заменяемости типов?
89. Как наследование разрушает инкапсуляцию? Как применить инкапсуляцию при наследовании
90. Что такое полиморфизм?
91. Как реализуется полиморфизм?
92. Что такое позднее связывание?
93. Что такое раннее связывание?
94. Каковы преимущества полиморфизма времени выполнения?
95. Каковы преимущества полиморфизма времени компиляции?
96. Что такое интерфейс?
97. Как полиморфизм позволяет реализовать интерфейсы?
98. Что такое виртуальная функция?
99. Что такое перегрузка метода?
100. Что такое интерфейс?
101. Чем интерфейс отличается от абстрактного класса?
102. С какими недостатками языка справляются интерфейсы?
103. В чем состоит основное назначение интерфейсов?
104. Стандартные интерфейсы C#
105. Пространство имен `System.Collections`
106. Операции `is` и `as`
107. Как Windows взаимодействует с пользователем?
108. Форма инкапсулирует основные функции, необходимые для чего?
109. Класс `Control`
110. Важные свойства объекта `Control`
111. Важные методы объекта `Control`
112. Важные методы объекта `Control`
113. Важные события класс `Control`
114. Дополнительные события класса `Control`
115. Передача параметров между окнами
116. Окна родитель и потомок
117. Языковая кастомизация
118. Значок в трее – `NotifyIcon`
119. Процессы работы с данными:
120. Основы ADO .NET
121. Принцип единообразной работы с базами данных
122. Доступ к отсоединенными данным

123. ADO .NET. Доступ к данным
124. Набор данных, провайдер данных
125. ADO .NET. Объектная модель
126. Классы подсоединенных объектов
127. Классы отсоединенных объектов
128. объект DataTable
129. свойство Relations
130. класс Constraint
131. объект DataSet
132. Платформа .NET Framework

Типовые практические задания:

1. У вас есть доллары. Вы хотите обменять их на рубли. Есть информация о стоимости купли-продажи в банках города. В городе N банков. Составьте программу, определяющую, какой банк выбрать, чтобы выгодно обменять доллары на рубли. Какой банк выбрать, если необходимо купить доллары?
2. Даны результаты соревнования по бегу (время в секундах). Вывести на экран результаты первой тройки (использовать сортировку массива).
3. Ввести с клавиатуры в одну строку фамилию, имя и отчество, разделив их пробелом. Вывести инициалы и фамилию.
4. Тестирование коллектива. Пусть матрица размером N x M содержит информацию о результатах тестирования студентов некоторой подгруппы из N человек. В j-ом столбце каждого студента стоит 1, если он правильно ответил на j-ое тестовое задание, и 0 – в противном случае. Всего M тестовых заданий.

### Второй период контроля

#### 1. Курсовая работа

Примерные темы курсовых работ:

1. Разработка программы-тренажера по развитию навыков работы с компьютерной мышью
2. Разработка интерактивной игры по развитию логики и внимания
3. Программная реализация алгоритма поиска пути в лабиринте
4. Создание интерактивных приложений средствами платформы Unity
5. Программная демонстрация методов кодирования растровых графических изображений
6. Разработка интерактивных физических игр средствами Unity
7. Разработка интерактивных математических игр
8. Разработка чат-бота для учебного центра
9. Разработка образовательной игры "Графы"
10. Разработка приложения для игры в ребусы
11. Разработка игрового приложения "Головоломки"
12. Алгоритмы закрашивания геометрических фигур
13. Разработка сетевого чата для системного администратора
14. Разработка демонстрационной программы по помехоустойчивому кодированию
15. Реализация алгоритмов компьютерной алгебры
16. Программирование аналоговых датчиков аппаратно-программной системы Arduino
17. Программирование цифровых датчиков аппаратно-программной системы Arduino
18. Программирование цифровых датчиков аппаратно-программной системы Arduino с помощью графической среды FLProg
19. Создание интерактивной игры по изучению основных терминов информатики на Unity 3D
20. Создание интерактивной игры по изучению иностранных слов на Unity 3D
21. Разработка программы-тренажера по информатике по теме "Логика"

#### 5.3. Примерные критерии оценивания ответа студентов на экзамене (зачете):

Отметка	Критерии оценивания
"Отлично"	<ul style="list-style-type: none"> <li>-дается комплексная оценка предложенной ситуации</li> <li>-демонстрируются глубокие знания теоретического материала и умение их применять</li> <li>-последовательное, правильное выполнение всех заданий</li> <li>-умение обоснованно излагать свои мысли, делать необходимые выводы</li> </ul>

"Хорошо"	<ul style="list-style-type: none"> <li>- дается комплексная оценка предложенной ситуации</li> <li>- демонстрируются глубокие знания теоретического материала и умение их применять</li> <li>- последовательное, правильное выполнение всех заданий</li> <li>- возможны единичные ошибки, исправляемые самим студентом после замечания преподавателя</li> <li>- умение обоснованно излагать свои мысли, делать необходимые выводы</li> </ul>
"Удовлетворительно" ("зачтено")	<ul style="list-style-type: none"> <li>- затруднения с комплексной оценкой предложенной ситуации</li> <li>- неполное теоретическое обоснование, требующее наводящих вопросов преподавателя</li> <li>- выполнение заданий при подсказке преподавателя</li> <li>- затруднения в формулировке выводов</li> </ul>
"Неудовлетворительно" ("не зачтено")	<ul style="list-style-type: none"> <li>- неправильная оценка предложенной ситуации</li> <li>- отсутствие теоретического обоснования выполнения заданий</li> </ul>

## **6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

### **1. Лекции**

Лекция - одна из основных форм организации учебного процесса, представляющая собой устное, монологическое, систематическое, последовательное изложение преподавателем учебного материала с демонстрацией слайдов и фильмов. Работа обучающихся на лекции включает в себя: составление или слежение за планом чтения лекции, написание конспекта лекции, дополнение конспекта рекомендованной литературой.

Требования к конспекту лекций: краткость, схематичность, последовательная фиксация основных положений, выводов, формулировок, обобщений. В конспекте нужно помечать важные мысли, выделять ключевые слова, термины. Последующая работа над материалом лекции предусматривает проверку терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. В конспекте нужно обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии.

### **2. Лабораторные**

Лабораторные занятия проводятся в специально оборудованных лабораториях с применением необходимых средств обучения (лабораторного оборудования, образцов, нормативных и технических документов и т.п.).

При выполнении лабораторных работ проводятся: подготовка оборудования и приборов к работе, изучение методики работы, воспроизведение изучаемого явления, измерение величин, определение соответствующих характеристик и показателей, обработка данных и их анализ, обобщение результатов. В ходе проведения работ используются план работы и таблицы для записей наблюдений.

При выполнении лабораторной работы студент ведет рабочие записи результатов измерений (испытаний), оформляет расчеты, анализирует полученные данные путем установления их соответствия нормам и/или сравнения с известными в литературе данными и/или данными других студентов. Окончательные результаты оформляются в форме заключения.

### **3. Экзамен**

Экзамен преследует цель оценить работу обучающегося за определенный курс: полученные теоретические знания, их прочность, развитие логического и творческого мышления, приобретение навыков самостоятельной работы, умения анализировать и синтезировать полученные знания и применять их для решения практических задач.

Экзамен проводится в устной или письменной форме по билетам, утвержденным заведующим кафедрой. Экзаменационный билет включает в себя два вопроса и задачи. Формулировка вопросов совпадает с формулировкой перечня вопросов, доведенного до сведения обучающихся не позднее чем за один месяц до экзаменационной сессии.

В процессе подготовки к экзамену организована предэкзаменационная консультация для всех учебных групп.

При любой форме проведения экзаменов по билетам экзаменатору предоставляется право задавать студентам дополнительные вопросы, задачи и примеры по программе данной дисциплины. Дополнительные вопросы, также как и основные вопросы билета, требуют развернутого ответа.

Результат экзамена выражается оценкой «отлично», «хорошо», «удовлетворительно».

### **4. Курсовая работа**

Курсовая работа — студенческое научное исследование по одной из базовых дисциплин учебного плана либо специальности, важный этап в подготовке к написанию выпускной квалификационной работы. Темы работ предлагаются и утверждаются кафедрой. Студент может предложить тему самостоятельно, однако она не должна выходить за рамки учебного плана. На 1-2 курсах данная работа носит скорее реферативный характер, на старших — исследовательский. Работа обычно состоит из теоретической части (последовательное изложение подходов, мнений, сложившихся в науке по избранному вопросу) и аналитической (анализ проблемы на примере конкретной ситуации (на примере группы людей, организации). Объем курсовой работы составляет 20-60 страниц. По завершению работы над курсовой, студенты защищают ее публично перед своими однокурсниками и преподавателями.

Этапы выполнения курсовой работы:

1. выбор темы и ее согласование с научным руководителем;
2. сбор материалов, необходимых для выполнения курсовой работы;
3. разработка плана курсовой работы и его утверждение научным руководителем;
4. систематизация и обработка отобранного материала по каждому из разделов работы или проблеме с применением современных методов;
5. формулирование выводов и обсуждение их с научным руководителем;
6. написание работы согласно требованиям стандарта и методическим указаниям к ее выполнению (введение, главы основной части, заключение, приложения, список литературы).

При оформлении курсовой работы следует придерживаться рекомендаций, представленных в документе «Регламент оформления письменных работ».

### **5. Тест**

Тест это система стандартизованных вопросов (заданий), позволяющих автоматизировать процедуру измерения уровня знаний и умений обучающихся. Тесты могут быть аудиторными и внеаудиторными. Преподаватель доводит до сведения студентов информацию о проведении теста, его форме, а также о разделе (теме) дисциплины, выносимой на тестирование.

При самостоятельной подготовке к тестированию студенту необходимо:

- проработать информационный материал по дисциплине. Проконсультироваться с преподавателем по вопросу выбора учебной литературы;
- выяснить все условия тестирования заранее. Необходимо знать, сколько тестов вам будет предложено, сколько времени отводится на тестирование, какова система оценки результатов и т.д.
- работая с тестами, внимательно и до конца прочесть вопрос и предлагаемые варианты ответов; выбрать правильные (их может быть несколько); на отдельном листке ответов выписать цифру вопроса и буквы, соответствующие правильным ответам. В случае компьютерного тестирования указать ответ в соответствующем поле (полях);
- в процессе решения желательно применять несколько подходов в решении задания. Это позволяет максимально гибко оперировать методами решения, находя каждый раз оптимальный вариант.
- решить в первую очередь задания, не вызывающие трудностей, к трудному вопросу вернуться в конце.
- оставить время для проверки ответов, чтобы избежать механических ошибок.

### **6. Задача**

Задачи позволяют оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, узнавание объектов изучения в рамках определенного раздела дисциплины; умения синтезировать, анализировать, обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением причинно-следственных связей.

Алгоритм решения задач:

- 1.Внимательно прочтайте условие задания и уясните основной вопрос, представьте процессы и явления, описанные в условии.
- 2.Повторно прочтите условие для того, чтобы чётко представить основной вопрос, проблему, цель решения, заданные величины, опираясь на которые можно вести поиск решения.
- 3.Произведите краткую запись условия задания.
- 4.Если необходимо, составьте таблицу, схему, рисунок или чертёж.
- 5.Установите связь между искомыми величинами и данными; определите метод решения задания, составьте план решения.
- 6.Выполните план решения, обосновывая каждое действие.
- 7.Проверьте правильность решения задания.
- 8.Произведите оценку реальности полученного решения.
- 9.Запишите ответ.

## **7. ПЕРЕЧЕНЬ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ**

1. Дифференцированное обучение (технология уровневой дифференциации)
2. Проектные технологии
3. Проблемное обучение

## **8. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ**

1. компьютерный класс – аудитория для самостоятельной работы
2. учебная аудитория для лекционных занятий
3. компьютерный класс
4. Лицензионное программное обеспечение:
  - Операционная система Windows 10
  - Microsoft Office Professional Plus
  - Антивирусное программное обеспечение Kaspersky Endpoint Security для бизнеса - Стандартный Russian Edition
  - Справочная правовая система Консультант плюс
  - 7-zip
  - Adobe Acrobat Reader DC
  - MS Visual Studio COMMUNITY