

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 «ЮУГПУ»)

РАБОЧАЯ ПРОГРАММА

Шифр	Наименование дисциплины (модуля)
Б1.В	Программирование
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) .....	4
3. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий .....	5
4. Учебно-методическое и информационное обеспечение дисциплины .....	11
5. Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю) .....	12
6. Методические указания для обучающихся по освоению дисциплины .....	17
7. Перечень образовательных технологий .....	18
8. Описание материально-технической базы .....	19

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1 Дисциплина «Программирование» относится к модулю части, формируемой участниками образовательных отношений, Блока 1 «Дисциплины/модули» основной профессиональной образовательной программы по направлению подготовки 09.03.02 «Информационные системы и технологии» (уровень образования бакалавр). Дисциплина является дисциплиной по выбору.

1.2 Общая трудоемкость дисциплины составляет 3 з.е., 108 час.

1.3 Изучение дисциплины «Программирование» основано на знаниях, умениях и навыках, полученных при изучении обучающимися дисциплин образовательной программы общего среднего образования.

1.4 Дисциплина «Программирование» формирует знания, умения и компетенции, необходимые для освоения следующих дисциплин: «Алгоритмы и структуры данных», «Инструментальные средства информационных систем», «Объектно-ориентированное программирование», «Технологии программирования».

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 базовых структур данных;
- 2) изучение основ процедурного программирования;
- 3) изучение основ модульного 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
<b>Код и наименование индикатора достижения компетенции</b>	
1	ПК-2 способность выполнять интеграцию программных модулей и компонент
	ПК.2.1 Знать модели процесса и принципы разработки информационных систем, основные подходы к интегрированию программных модулей в информационные системы.
	ПК.2.2 Уметь интегрировать модули в информационную систему; отлаживать программные модули.
	ПК.2.3 Иметь навыки интеграции модулей в информационную систему; отладки программных модулей

№ п/п	Код и наименование индикатора достижения компетенции	Образовательные результаты по дисциплине
1	ПК.2.1 Знать модели процесса и принципы разработки информационных систем, основные подходы к интегрированию программных модулей в информационные системы.	3.1 Знать основные алгоритмические конструкции и структуры данных
2	ПК.2.2 Уметь интегрировать модули в информационную систему; отлаживать программные модули.	У.1 Уметь использовать основные алгоритмические конструкции и структуры данных для решения задач
3	ПК.2.3 Иметь навыки интеграции модулей в информационную систему; отладки программных модулей	В.1 Владеть навыками разработки программ с использованием основных алгоритмических конструкций и структур данных

**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>18</b>	<b>30</b>	<b>60</b>	<b>108</b>
<b>Первый период контроля</b>				
<b>Платформа .NET. Введение в C#</b>	<b>6</b>	<b>18</b>	<b>37</b>	<b>61</b>
Платформа .NET	2		1	3
Типы данных C#.Операторы в C#	2		2	4
Циклы в C#	2		2	4
Знакомство с интегрированной средой		2	2	4
Простейшие программы.Работа с типами-значениями в C#		2	2	4
Условный оператор		2	4	6
Составное условие, условный оператор		2	4	6
Циклы		4	8	12
Типовые циклические алгоритмы		4	8	12
Оператор выбора. Перечисляемый тип.		2	4	6
<b>Классы в C#</b>	<b>12</b>	<b>12</b>	<b>23</b>	<b>47</b>
ООП	2		1	3
Классы в c#	2		1	3
Массивы. Строки	2		1	3
Структуры. Файлы	2		2	4
Коллекции	4		2	6
Методы класса		4	4	8
Массивы		4	8	12
Строки		4	4	8
Итого по видам учебной работы	18	30	60	108
<b>Форма промежуточной аттестации</b>				
Зачет				
<b>Итого за Первый период контроля</b>				<b>108</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. Платформа .NET. Введение в C#</b>  <i>Формируемые компетенции, образовательные результаты:</i> ПК-2: 3.1 (ПК.2.1), У.1 (ПК.2.2), В.1 (ПК.2.3)	<b>6</b>
1.1. Платформа .NET <ul style="list-style-type: none"> <li>1. Microsoft.NET (.NET Framework) – программная платформа.</li> <li>2. CLS (Common Language Specification)</li> <li>3. FCL (.NET Framework Class Library)</li> <li>4. MSIL (Microsoft Intermediate Language)</li> <li>5. Исполняемый модуль, сборка</li> <li>6. Структура среды выполнения CLR</li> <li>7. Сборка мусора</li> <li>8. Пространство имён</li> </ul> Учебно-методическая литература: 1, 2, 4	2
1.2. Типы данных C#.Операторы в C# <ul style="list-style-type: none"> <li>1. История создания языка C#</li> <li>2. Язык C# и первый проект</li> <li>3. Система типов языка C#</li> <li>4. Типы значения</li> <li>5. Ссылочные типы</li> <li>6. Целые и вещественные типы данных</li> <li>7. Операции в C#</li> <li>8. Класс Math и его функции</li> <li>1. Операция присваивания</li> <li>2. Блок или составной оператор</li> <li>3. Пустой оператор</li> <li>4. Условная трёхместная операция</li> <li>5. Условный оператор</li> <li>6. Оператор выбора</li> <li>7. Операторы break и continue</li> </ul> Учебно-методическая литература: Учебно-методическая литература: 1, 2, 4	2
1.3. Циклы в C# <ul style="list-style-type: none"> <li>Циклы в C# 2</li> <li>1. Цикл for</li> <li>2. Цикл While</li> <li>3. Цикл Do while</li> <li>4. Отладка и тестирование программы</li> </ul> Учебно-методическая литература: 1, 2, 4	2
<b>2. Классы в C#</b>  <i>Формируемые компетенции, образовательные результаты:</i> ПК-2: 3.1 (ПК.2.1), У.1 (ПК.2.2), В.1 (ПК.2.3)	<b>12</b>
2.1. ООП <ul style="list-style-type: none"> <li>1. Основные понятия ООП: объект и класс.</li> <li>2. Инкапсуляция</li> <li>3. Наследование</li> <li>4. Полиморфизм</li> </ul> Учебно-методическая литература: 1, 2, 4	2

<p>2.2. Классы в с#</p> <ol style="list-style-type: none"> <li>1. Основные понятия ООП: объект и класс.</li> <li>2. Объявление класса</li> <li>3. Поля</li> <li>4. Константы</li> <li>5. Методы. Параметры методов.</li> <li>6. Конструкторы</li> <li>7. Деструкторы</li> <li>8. Свойства</li> <li>9. События</li> <li>10. Операторы</li> <li>11. Структуры</li> <li>12. Другие члены классов</li> </ol> <p>Учебно-методическая литература: 1, 2, 4</p>	2
<p>2.3. Массивы. Строки</p> <ol style="list-style-type: none"> <li>3. Массивы. Строки 2</li> <li>1. Класс Agtau: методы и свойства</li> <li>2. Создание линейного массива, работа с элементами массива.</li> <li>3. Цикл foreach</li> <li>4. Примеры программ на линейные массивы</li> <li>5. Матрицы</li> <li>6. Класс string: методы и свойства</li> <li>7. Примеры работы со строками</li> </ol> <p>Учебно-методическая литература: 1, 2, 4</p>	2
<p>2.4. Структуры. Файлы</p> <ol style="list-style-type: none"> <li>1. Описание структуры</li> <li>2. Массив структур</li> <li>3. Типы файлов</li> <li>4. Чтение и данных из файлов</li> <li>5. Пространство имен IO</li> </ol> <p>Учебно-методическая литература: 1, 2, 4</p>	2
<p>2.5. Коллекции</p> <ol style="list-style-type: none"> <li>1. Коллекции</li> <li>2. ArrayList</li> <li>3. Список List&lt;T&gt;</li> <li>4. Двухсвязный список LinkedList&lt;T&gt;</li> <li>5. Очередь Queue&lt;T&gt;</li> <li>6. Стек Stack&lt;T&gt;</li> <li>7. Словарь Dictionary&lt;T, V&gt;</li> <li>8. Класс ObservableCollection</li> <li>9. Интерфейсы IEnumerable и IEnumerator</li> </ol> <p>Учебно-методическая литература: 1, 2, 4</p>	4

### 3.2 Лабораторные

Наименование раздела дисциплины (модуля)/ Тема и содержание	Трудоемкость (кол-во часов)
<b>1. Платформа .NET. Введение в C#</b>	<b>18</b>
<b>Формируемые компетенции, образовательные результаты:</b> ПК-2: 3.1 (ПК.2.1), У.1 (ПК.2.2), В.1 (ПК.2.3)	
<p>1.1. Знакомство с интегрированной средой</p> <ol style="list-style-type: none"> <li>1. Настройка среды MS Visual Studio под разработку приложений на языке C#</li> <li>2. Создание и сохранение консольного приложения</li> <li>3. Работа с редактором кода</li> <li>4. Свойства проекта</li> </ol> <p>Учебно-методическая литература: 1, 2, 4</p>	2
<p>1.2. Простейшие программы. Работа с типами-значениями в C#</p> <ol style="list-style-type: none"> <li>1. Ответы на контрольные вопросы по теории</li> <li>2. Решение устных задач (подготовка к тесту с практическими заданиями)</li> <li>3. Решение простейших задач</li> </ol> <p>Учебно-методическая литература: 1, 3, 4</p> <p>Профессиональные базы данных и информационные справочные системы: 1</p>	2

1.3. Условный оператор 1. Ответы на контрольные вопросы по теории 2. Решение устных задач (подготовка к тесту с практическими заданиями) 3. Решение задач (3 задачи) Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	2
1.4. Составное условие, условный оператор 1. Ответы на контрольные вопросы по теории 2. Запись условного оператора с составным условием по блок-схеме 3. Запись условного оператора с составным условием (вычисление значения функции по рисунку графика функции) 4. Решение задач Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	2
1.5. Циклы 1. Ответы на контрольные вопросы по теории 2. Решение устных задач (подготовка к тесту с практическими заданиями) 3. Решение задач 4. Отладка программы Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	4
1.6. Типовые циклические алгоритмы 1. Разбор типовых циклических алгоритмов: о Вычисление произведения последовательности чисел о Организация цикла с несколькими одновременно изменяющимися параметрами. о Программирование вложенных циклов о Поиск минимального (максимального) элемента 2. Решение задач 3. Отладка программ Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	4
1.7. Оператор выбора. Перечисляемый тип. 1. Ответы на контрольные вопросы по теории 2. Решение устных задач (подготовка к тесту с практическими заданиями) 3. Решение задач (5 задачи) 4. Отладка программы Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	2
<b>2. Классы в C#</b>	<b>12</b>
<b>Формируемые компетенции, образовательные результаты:</b> ПК-2: 3.1 (ПК.2.1), У.1 (ПК.2.2), В.1 (ПК.2.3)	
2.1. Методы класса 1. Контрольные вопросы по теории 2. Решение устных упражнений 3. Решение задач на методы с помощью метода пошаговой детализации программ (4 задачи, в том числе описание функций для исполнителя Черепаха) 4. Отладка программы Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	4
2.2. Массивы 1. Контрольные вопросы по теории 2. Решение задач на линейные массивы (4 задачи) 3. Отладка программы Учебно-методическая литература: 2, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	4
2.3. Строки 1. Контрольные вопросы по теории 2. Форматирование строк 3. Решение задач на строки (8 задачи) 4. Отладка программы Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	4

### 3.3 СРС

Наименование раздела дисциплины (модуля)/ Тема для самостоятельного изучения	Трудоемкость (кол-во часов)
<b>1. Платформа .NET. Введение в C#</b>	<b>37</b>
<b>Формируемые компетенции, образовательные результаты:</b> ПК-2: 3.1 (ПК.2.1), У.1 (ПК.2.2), В.1 (ПК.2.3)	
1.1. Платформа .NET <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу Учебно-методическая литература: 1, 4	1
1.2. Типы данных C#.Операторы в C# <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу Учебно-методическая литература: 1, 2, 4	2
1.3. Циклы в C# <b>Задание для самостоятельного выполнения студентом:</b> Повторение теории. Подготовка к итоговому тестированию по разделу Учебно-методическая литература: 1, 2, 4	2
1.4. Знакомство с интегрированной средой <b>Задание для самостоятельного выполнения студентом:</b> Инвариантная часть Написать программу для решения линейной задачи Вариативная часть Решение задачи повышенной сложности Учебно-методическая литература: 2, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	2
1.5. Простейшие программы.Работа с типами-значениями в C# <b>Задание для самостоятельного выполнения студентом:</b> Инвариантная часть Написать программу для решения линейной задачи Вариативная часть Решение задачи повышенной сложности Учебно-методическая литература: 2, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	2
1.6. Условный оператор <b>Задание для самостоятельного выполнения студентом:</b> Инвариантная часть Написать программу для решения задачи Вариативная часть Решение задачи повышенной сложности Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	4
1.7. Составное условие, условный оператор <b>Задание для самостоятельного выполнения студентом:</b> Инвариантная часть Написать программу для решения задачи Вариативная часть Решение задачи повышенной сложности Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	4
1.8. Циклы <b>Задание для самостоятельного выполнения студентом:</b> Инвариантная часть Написать программу для решения задачи Вариативная часть Решение задачи повышенной сложности Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	8

1.9. Типовые циклические алгоритмы <i>Задание для самостоятельного выполнения студентом:</i> Инвариантная часть Написать программу для решения задачи Вариативная часть Решение задачи повышенной сложности Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	8
1.10. Оператор выбора. Перечисляемый тип. <i>Задание для самостоятельного выполнения студентом:</i> Инвариантная часть Написать программу для решения задачи Вариативная часть Решение задачи повышенной сложности Учебно-методическая литература: 1, 4 Профессиональные базы данных и информационные справочные системы: 1	4
<b>2. Классы в C#</b> <i>Формируемые компетенции, образовательные результаты:</i> ПК-2: 3.1 (ПК.2.1), У.1 (ПК.2.2), В.1 (ПК.2.3)	<b>23</b>
2.1. ООП <i>Задание для самостоятельного выполнения студентом:</i> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 2, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	1
2.2. Классы в с# <i>Задание для самостоятельного выполнения студентом:</i> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	1
2.3. Массивы. Строки <i>Задание для самостоятельного выполнения студентом:</i> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	1
2.4. Структуры. Файлы <i>Задание для самостоятельного выполнения студентом:</i> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 2, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	2
2.5. Коллекции <i>Задание для самостоятельного выполнения студентом:</i> Повторение теории. Подготовка к итоговому тестированию по разделу. Учебно-методическая литература: 2, 4 Профессиональные базы данных и информационные справочные системы: 1	2
2.6. Методы класса <i>Задание для самостоятельного выполнения студентом:</i> Инвариантная часть Написать программу для решения задачи Вариативная часть Решение задачи повышенной сложности  Учебно-методическая литература: 2, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	4
2.7. Массивы <i>Задание для самостоятельного выполнения студентом:</i> Инвариантная часть Написать программу для решения задачи Вариативная часть Решение задачи повышенной сложности Учебно-методическая литература: 1, 3, 4 Профессиональные базы данных и информационные справочные системы: 1	8

2.8. Строки

4

***Задание для самостоятельного выполнения студентом:***

Инвариантная часть

Написать программу для решения задачи

Вариативная часть

Решение задачи повышенной сложности

Учебно-методическая литература: 1, 3, 4

Профессиональные базы данных и информационные справочные системы: 1

## 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.1 : 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	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>		
3	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>
4	Сайт <a href="https://docs.microsoft.com/ru-ru/">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>

### 4.2. Современные профессиональные базы данных и информационные справочные системы, используемые при осуществлении образовательного процесса по дисциплине

№ п/п	Наименование базы данных	Ссылка на ресурс
1	Официальный информационный портал ЕГЭ	<a href="http://www.ege.edu.ru">http://www.ege.edu.ru</a>

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

### 5.1. Описание показателей и критериев оценивания компетенций

Код компетенции по ФГОС			
Код образовательного результата дисциплины	Текущий контроль		Помежуточная аттестация
	Тест	Задача	
ПК-2			
3.1 (ПК.2.1)	+		+
У.1 (ПК.2.2)		+	+
В.1 (ПК.2.3)		+	+

### 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 "Платформа .NET. Введение в C#":

##### 1. Задача

Пример индивидуального задания «Простейшие программы».

1 Вычислить сумму элементов гармонического ряда

Пример индивидуального задания «Условный оператор»

2 Даны две переменные целого типа: А и В. Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных А и В.

Пример индивидуального задания «Циклы»

3 Вычислить сумму элементов гармонического ряда

Количество баллов: 50

## 2. Тест

1. Пусть объявлен целочисленный массив arr размерности 10:

```
int[] arr = new int[10];
```

Необходимо проинициализировать его элементы так, чтобы значение i-го элемента было равно i, 0 <= i < 10.

Какие из ниже перечисленных вариантов инициализации верны?

a) for (int i = 0; i < 10; i++)  
{  
arr[i] = i;  
}

---

b) int i = 0;  
foreach (int a in arr)  
a = i++;

---

c) int i = 0;  
while (i < 10)  
arr[i++] = i++;

---

d) int i = 0;  
do  
{  
arr[i] = i;  
i++;  
} while (i < 10);

2. Какое число получится в результате выполнения фрагмента программы?

```
int[] a = {3, 8, 0, -6, 0, -1, -9, 3};  
int i, k = Math.Abs(a[2]);  
for (i = 0; i < 8; i++)  
if (a[i] > k) k = Math.Abs(a[i]) - 2;  
Console.WriteLine(k);
```

3. Какое число получится в результате выполнения фрагмента программы?

```
int[] a = {3, 8, 0, -6, 0, -1, -9, 3};  
int j = 2, k = 0;  
for (int i = 0; i < 8; i++)  
if (a[i] < j) ++k;  
Console.WriteLine(k);
```

4. Что будет выведено на экран в результате выполнения фрагмента программы?

```
char[] a = {'a', 'b', 'c', 'r', 'c', 'a', 'a', 'b'};  
char k = 'a';  
for (int i = 0; i < 8; i++)  
if (a[i] > k)  
{ k = a[i];  
Console.Write(k); }
```

Количество баллов: 50

Типовые задания к разделу " Классы в C#":

## 1. Задача

Пример индивидуального задания «Массивы»

4 В городе N банков. Вам известен курс обмена местной валюты на тугрики. Необходимо написать программу для поиска банка с самым выгодным курсом.

Пример индивидуального задания «Строки»

5 Даны строка содержащая имя фамилию и отчество. Например: Сидоров Иван Петрович.

Необходимо вывести строку в следующем виде: Фамилия, инициалы

Сидоров И.П.

Пример индивидуального задания «файлы»

6 Опишите тип записи - сведения о книге (например, по информатике). Сформируйте файл книг, необходимых преподавателю информатики. Составьте программу, которая подбирает книги для курса, номер которого вводится, печатает имена их авторов и год издания. Расширьте возможности Вашей программы другими процедурами обработки файлов по Вашему выбору.

Списки

Формат входных данных

В каждой строке сначала записан номер класса (число, равное 9, 10 или 11), затем (через пробел) – фамилия ученика. Общее число строк в файле не превосходит 100000. Длина каждой фамилии не превосходит 50 символов.

Формат выходных данных

Необходимо вывести список школьников по классам: сначала всех учеников 9 класса, затем – 10, затем – 11. Внутри одного класса порядок вывода фамилий должен быть таким же, как на входе.

Очередь:

В игре в пьяницу карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остается без карт – проигрывает.

Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту ("шестерка берет туза").

Игрок, который забирает себе карты, сначала кладет под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды).

Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9.

Входные данные

Программа получает на вход две строки: первая строка содержит 5 чисел, разделенных пробелами — номера карт первого игрока, вторая – аналогично 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой.

Выходные данные

Программа должна определить, кто выигрывает при данной раздаче, и вывести слово first или second, после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении 106 ходов игра не заканчивается, программа должна вывести слово botva.

Примеры

входные данные

1 3 5 7 9

2 4 6 8 0

выходные данные

second 5

Количество баллов: 50

## 2. Text

2. Какое число получится в результате выполнения фрагмента программы?

```
int[] a = {3, 8, 0, -6, 0, -1, -9, 3};  
int i, k= Math.Abs(a[2]);  
for (i = 0; i < 8; i++)  
if (a[i] > k) k = Math.Abs(a[i]) - 2;  
Console.WriteLine(k);
```

3. Какое число получится в результате выполнения фрагмента программы?

```
int[] a = {3, 8, 0, -6, 0, -1, -9, 3};  
int j = 2, k = 0;  
for (int i = 0; i < 8; i++)  
if (a[i] < j) ++k;  
Console.WriteLine(k);
```

4. Что будет выведено на экран в результате выполнения фрагмента программы?

```
char[] a = {'a', 'b', 'c', 'r', 'c', 'a', 'a', 'b'};  
char k = 'a';  
for (int i = 0; i < 8; i++)  
if (a[i] > k)  
{ k = a[i];  
Console.Write(k); }
```

5. Что будет выведено на экран?

```
String s1="мама", s2="мыла", s3="раму";  
System.Console.WriteLine(String.Compare(s1,s1)); _____  
System.Console.WriteLine(String.Compare(s1,s2)); _____  
System.Console.WriteLine(String.Compare(s1,s3)); _____
```

6. Что будет выведено на экран?

```
String s1="информатика", s2;  
s2=s1.Substring(3, 2);  
System.Console.WriteLine(s2);
```

7. 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);  
}
```

### 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. Зачет

Вопросы к зачету:

1. Платформа .NET Framework
2. the common language runtime (CLR)
3. the .NET Framework class library (.NET FCL).
4. Среда Времени Выполнения или Виртуальная Машина.
5. .NET Framework Class Library
6. MSIL (Microsoft Intermediate Language, он же IL – Intermedia Language) – промежуточный язык
7. JIT-компилятор
8. CTS - Common Type System Стандартная Система Типов.
9. Сборка мусора
10. Пространство имён
11. Структура проекта консольного приложения
12. Типы данных языка. Типы значений и ссылочные типы
13. Какие идентификаторы и диапазоны значений имеют числовые типы данных?
14. Какие операции можно осуществлять над переменными целого и вещественного типов?
15. Каков формат записи операции присваивания?
16. Какие встроенные функции существуют для работы с переменными целых и вещественных типов? Что
17. необходимо сделать, чтобы использовать их в программе?
18. Как выполняется преобразование типов в выражениях?
19. как выполняется ввод и вывод данных на экран в C#?
20. Каков приоритет выполнения операций в C#?
21. Что представляет собой логический тип данных в языке C#?
22. Как записываются логические операции в C#?
23. Как записывается условный оператор в языке C#?
24. Каковы правила записи составных условий в C#?
25. Какие виды циклов определены в языке C#?
26. Для каждого вида цикла ответьте на следующие вопросы:
27. В каких ситуациях удобно использовать тот или иной оператор цикла?
28. Какова структура цикла?
29. Объясните принцип работы цикла. Каково условие выхода из цикла?
30. Как записывается оператор выбора в языке C#?
31. Какой идентификатор имеет перечисляемый тип в языке C#? Как он записывается?
32. Что может быть значением в перечисляемом типе?
33. Класс Random и его функции
34. Класс Array: методы и свойства (из справки в лабораторной работе и примеров лекций)
35. Создание линейного массива, работа с элементами массива.
36. Цикл foreach
37. Примеры программ на линейные массивы и матрицы (типа задач ЕГЭ, примеры
38. <http://inf.reshuege.ru/test?theme=284>)
39. 37. Матрицы (примеры тестов в лабораторной работе и из примеров выше)
40. 38. Класс string: методы и свойства (из справки в лабораторной работе и примеров лекций)
41. 39. Объявление класса
42. 40. Поля класса
43. 41. Константы класса
44. 42. Методы. Параметры методов (Примеры тестов в лабораторной работе)

Типовые практические задания:

1. Пример индивидуального задания «Массивы» В городе N банков. Вам известен курс обмена местной валюты на тугрики. Необходимо написать программу для поиска банка с самым выгодным курсом.
2. Пример индивидуального задания «Строки» Дано строка содержащая имя фамилию и отчество. Например: Сидоров Иван Петрович. Необходимо вывести строку в следующем виде: Фамилия, инициалы ¶ Сидоров И.П.

3. Пример индивидуального задания «файлы»¶ Опишите тип записи - сведения о книге (например, по информатике). Сформируйте файл книг, необходимых преподавателю информатики. Составьте программу, которая подбирает
4. книги для курса, номер которого вводится, печатает имена их авторов и год издания. Расширьте возможности
5. Вашей программы другими процедурами обработки файлов по Вашему выбору.¶

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

<b>Отметка</b>	<b>Критерии оценивания</b>
"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. Зачет

Цель зачета – проверка и оценка уровня полученных студентом специальных знаний по учебной дисциплине и соответствующих им умений и навыков, а также умения логически мыслить, аргументировать избранную научную позицию, реагировать на дополнительные вопросы, ориентироваться в массиве информации.

Подготовка к зачету начинается с первого занятия по дисциплине, на котором обучающиеся получают предварительный перечень вопросов к зачету и список рекомендуемой литературы, их ставят в известность относительно критериев выставления зачета и специфике текущей и итоговой аттестации. С самого начала желательно планомерно осваивать материал, руководствуясь перечнем вопросов к зачету и списком рекомендуемой литературы, а также путем самостоятельного конспектирования материалов занятий и результатов самостоятельного изучения учебных вопросов.

По результатам сдачи зачета выставляется оценка «зачтено» или «не зачтено».

### 4. 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.

### 5. 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 прочтите 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. Проблемное обучение
3. Цифровые технологии обучения

## **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