МИНОБРНАУКИ РОССИИ

Глазовский инженерно-экономический институт (филиал) Федерального государственного бюджетного образовательногоучреждения высшего образования «Ижевский государственный технический университет имени М.Т. Калашникова» (ГИЭИ (филиал) ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 3DA273AA38C240BA869FA062FFFB332C
Владелец: Золотов Алексей Васильевич
Действителен: с 29.07.2026 до 22.10.2027



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Программирование

направление подготовки: **09.03.01 «Информатика и вычислительная техника»**

направленность (профиль): **Автоматизированные системы обработки информации и управления**

уровень образования: **бакалавриат**

форма обучения: **очная**

общая трудоемкость дисциплины составляет: **11 зачетных единиц**

Кафедра «Машиностроение и информационные технологии»

Составитель:

Рабочая программа составлена в соответствии с требованиями федерального государственногообразовательного стандарта высшего образования по направлению подготовки 09.03.01 «Информатика и вычислительная техника» и рассмотрена на заседании кафедры.

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.01 «Информатика и вычислительная техника» и рассмотрена на заседании кафедры.

Протокол от 22.05.2023 г. № 5

Заведующий кафедрой  А.Г. Горбушин

22.05.2023 г.

СОГЛАСОВАНО

Количество часов рабочей программы и формируемые компетенции соответствуют учебному плану по направлению подготовки 09.03.01 «Информатика и вычислительная техника», профиль «Автоматизированные системы обработки информации и управления».

Протокол заседания учебно-методической комиссии от 22 мая 2023 г. № 2

Председатель учебно-методической комиссии ГИЭИ

 А.Г. Горбушин

Руководитель образовательной программы  А.Г. Горбушин

22.05.2023 г.

Аннотация к дисциплине

Название дисциплины	Программирование
Направление подготовки (специальность)	09.03.01 "Информатика и вычислительная техника"
Направленность (профиль/программа/специализация)	Автоматизированные системы обработки информации и управления
Место дисциплины	Дисциплина относится к обязательной части Блока 1 «Дисциплины (модули)» ООП.
Трудоемкость (з.е. / часы)	11 з.е. / 396 часов
Цель изучения дисциплины	Получение базовых теоретических сведений и практических навыков программирования на языке высокого уровня (ЯВУ).
Компетенции, формируемые в результате освоения дисциплины	ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности; ОПК-5 Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем; ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;
Содержание дисциплины (основные разделы и темы)	обзор существующих программных сред, языков программирования, их классификация; алгоритмизация; процедурный язык программирования Паскаль; инсталляция и настройка программной среды; основы ООП;
Форма промежуточной аттестации	Экзамен (1 и 2 семестр)

1. **Цели и задачи дисциплины:**

Целью освоения дисциплины является получение базовых теоретических сведений и практических навыков программирования на языке высокого уровня (ЯВУ).

Задачи дисциплины:

- изучение структур данных компьютера и языка высокого уровня;
- изучение приемов алгоритмизации и программирования при решении задач на ЭВМ;
- освоение объектно-ориентированной методологии решения задач на ЭВМ;
- приобретение навыков работы в системе программирования Visual Studio;
- получение навыков оформления программной документации.

2. **Планируемые результаты обучения**

В результате освоения дисциплины у студента должны быть сформированы

Знания, приобретаемые в ходе освоения дисциплины

№ п/п	Знания
1.	Основные этапы решения задачи. Понятия алгоритма, блок-схемы алгоритма, основные виды вычислительных процессов
2.	Алфавит и лексику языков программирования, типы данных, правила записи выражений и операций
3.	Операторы ввода-вывода, операторы ветвления, операторы цикла
4.	Простые типы данных: множество значений, способ хранения, операции, особенности обработки
5.	Структурированные ссылочные типы данных: множество значений, способ хранения, операции, особенности обработки
6.	Основы процедурного программирования (процедуры, функции, библиотеки)
7.	Основы объектно-ориентированного программирования (классы, наследование, инкапсуляция, полиморфизм)
8.	Возможности современных интегрированные среды программирования
9.	Принципы рефакторинга, отладки, тестирования программы
10.	Инсталляция и параметрическая настройка интегрированной среды программирования

Умения, приобретаемые в ходе освоения дисциплины

№ п/п	Умения
1	Составлять алгоритмы решения задачи, оформлять блок-схемы алгоритмов
2	Выбирать необходимые типы данных для решения задачи
3	Реализовывать программы, реализующие линейные, разветвляющиеся и циклические вычислительные процессы
4	Обрабатывать структуры данных, файлы.
5	Создавать программы на процедурном языке
6	Создавать программы с использованием идеологии ООП
7	Работать в среде программирования, реализовывать рефакторинг, отладку, тестирование программы
8	Выбирать программные средства для решения задачи
9.	Выполнять настройку среды программирования

Навыки, приобретаемые в ходе освоения дисциплины

№ п/п	**Навыки**

н	
1.	Выбора и работы в интегрированной среде разработки
2.	Настройки среды разработки
3.	Алгоритмизации и программирования на процедурных и объектно-ориентированных языках
4.	Владения техникой рефакторинга, раздельной компиляции, отладчиком среды программирования, тестирования.

Компетенции, приобретаемые в ходе освоения дисциплины

Компетенции	Индикаторы	Знания	Умения	Навыки
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;	ОПК-2.1 Знать: современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности ОПК-2.2 Уметь: выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности ОПК-2. Владеть: навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	8	8	1
ОПК-5 Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;	ОПК-5.1Знать: основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем ОПК-5.2 Уметь: выполнять параметрическую настройку информационных и автоматизированных систем ОПК-5.3 Владеть: навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем	10	9	2
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;	ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы	1-7,9	1-7	3-4

3. Место дисциплины в структуре ООП

Дисциплина относится к обязательной части Блока 1 «Дисциплины (модули)» ООП.
Дисциплина изучается на 1 курсе в 1,2 семестрах.

Изучение дисциплины базируется на знаниях, умениях и навыках, полученных при освоении дисциплин (модулей): «Информатика», «Математика» (среднее(полное) общее образование).

Перечень последующих дисциплин (модулей), для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной (модулем): «Объектно-ориентированное программирование», «Базы данных», «Информационные системы».

4. Структура и содержание дисциплины

4.1 Структура дисциплины

№ п/п	Раздел дисциплины. Форма промежуточной аттестации *(по семестрам)*	Всего часов на раздел	Семестр	Распределение трудоемкости раздела (в часах) по видам учебной работы					Содержание самостоятельной работы
				контактная				СРС	
				лек	пр	лаб	КЧА		
1	2	3	4	5	6	7	8	10	11
1	Обзор существующих программных сред, языков программирования, их классификация	4	1	2	2				Самостоятельное изучение теоретического материала
2	Алгоритмизация	34	1	2	6	8		18	Подготовка к практическим лабораторным занятиям работы, тестам.
3	Процедурный язык программирования Паскаль	106	1	12	24	24		46	Подготовка к практическим лабораторным занятиям работы, тестам.
	Экзамен	36					0,4	35,6	Экзамен выставляется по совокупности результатов текущего контроля успеваемости или проводится в устной форме
	Итого за 1 семестр	**180**		**16**	**32**	**32**	**0,4**	**99,6**	
4	Инсталляция и настройка программной среды	12	2	2	2			8	Подготовка к практическим лабораторным занятиям работы, тестам.
5	Основы ООП	168	2	14	30	32		92	Подготовка к практическим лабораторным занятиям работы, тестам.
	Экзамен	36					0,4	35,6	Экзамен выставляется по совокупности результатов текущего контроля успеваемости или проводится в письменной форме.
	Итого за 2 семестр	**216**		**16**	**32**	**32**	**0,4**	**135,6**	
	Итого	**396**		**32**	**64**	**64**	**0,8**	**235,2**	

4.2 Содержание разделов курса и формируемых в них компетенций

№ п/п	Раздел дисциплины	Коды компетенции и индикаторов	Знания	Умения	Навыки	Форма контроля
1	Обзор существующих	ОПК-2.1, ОПК-	8	8	1	

	программных сред, языков программирования, их классификация	2.2, ОПК-2.3 ОПК-5.1, ОПК-5.2, ОПК-5.3 ОПК-8.1, ОПК-8.2, ОПК-8.3				
2	Алгоритмизация	ОПК-2.1, ОПК-2.2, ОПК-2.3 ОПК-5.1, ОПК-5.2, ОПК-5.3 ОПК-8.1, ОПК-8.2, ОПК-8.3	1	1	3	работа на практических занятиях: решения задач; защита лабораторной работы
3	Процедурный язык программирования Паскаль	ОПК-2.1, ОПК-2.2, ОПК-2.3 ОПК-5.1, ОПК-5.2, ОПК-5.3 ОПК-8.1, ОПК-8.2, ОПК-8.3	2-6	2-5	3	работа на практических занятиях: решения задач; защита лабораторной работы, тест
4	Инсталляция и настройка программной среды	ОПК-2.1, ОПК-2.2, ОПК-2.3 ОПК-5.1, ОПК-5.2, ОПК-5.3 ОПК-8.1, ОПК-8.2, ОПК-8.3	10	9	2	работа на практических занятиях: решения задач; защита лабораторной работы, тест
5	Основы ООП	ОПК-2.1, ОПК-2.2, ОПК-2.3 ОПК-5.1, ОПК-5.2, ОПК-5.3 ОПК-8.1, ОПК-8.2, ОПК-8.3	2-5, 7,9	2-4 6-7	3-4	работа на практических занятиях: решения задач; защита лабораторной работы, тест

4.3 Наименование тем лекций, их содержание и объем в часах

4.3.1. Первый семестр

№ п/п	№ раздела дисциплины	Наименование лекций	Трудоемкость (час)
1	**1**	1. Классификация языков программирования	**2**
2	**2**	1. Основные этапы решения задачи 2. Основные виды вычислительных процессов (разветвления, циклы) 3. Базовые алгоритмы (сумма, максимум, минимум) и методы решения	**2**
3	**3**	1. Простые типы данных Паскаля: целочисленные, логические, символьные, вещественные 2. Операторы разветвления 3. Операторы цикла 4. Структурированные типы данных: множества, массивы, строки, записи 5. Функции, процедуры, библиотеки 6. Файлы 7. Работа с динамической памятью (связные списки)	**12**
	Всего		**16**

4.3.2. Второй семестр

№	№ раздела	Наименование лекций	Трудоемкость

п/п	дисциплины		(час)
1	**4**	1. Инсталляция и настройка программной среды	**2**
2	**5**	1. Простые типы данных C# и строки 2. Ошибки, отладка, рефакторинг 3. Ветвления в C# 4. Циклы в C# 5. Массивы, коллекции, файлы в C# 6. Классы 7. Статическое и динамическое (инкапсуляция) 8. Наследование 9. Полиморфизм 10. Свойства и конструкторы 11. Структуры 12. Тестирование (основы)	**14**
	Всего		**16**

4.4 Наименования тем практических занятий, их содержание и объем в часах

4.4.1. Первый семестр

№	№ раздела дисциплины	Название практических работ	Трудоемкость (час)
1	1	Основные виды программных сред	2
2	2	Основные виды вычислительных процессов, виды блоков блок-схемы	2
3	2	Разветвляющиеся вычислительные процессы	2
4	2	Циклические вычислительные процессы	2
5	3	Целочисленный тип данных языка программирования Паскаль	2
6	3	Символьные тип данных языка программирования Паскаль	2
7	3	Логический тип данных языка программирования Паскаль	2
8	3	Вещественный тип данных языка программирования Паскаль	2
9	3	Тип Множество языка программирования Паскаль	2
10	3	Тип Массив языка программирования Паскаль	2
11	3	Тип Строка языка программирования Паскаль	2
12	3	Тип Запись языка программирования Паскаль	2
13	3	Функции процедуры языка программирования Паскаль	2
14	3	Библиотеки языка программирования Паскаль	2
15	3	Типизированные файлы (последовательный и прямой доступ), Текстовые файлы	2
16	3	Динамические связные списки (стеки, очереди, деревья)	2
Всего			32

4.4.2. Второй семестр

№	№ раздела дисциплины	Название практических работ	Трудоемкость (час)
1	4	Система программирования Visual Studio – инсталляция и настройка	2
2		C# Основные понятия. Базовые структуры данных.	2
3	5	C# Реализация разветвлений циклов	2
4	5	C# Массивы	2
5	5	C# Коллекции	2
6	5	C# Строки	2
7	5	C# Классы (поля и методы)	2

8	5	C# Статические и динамические поля	2
9	5	C# Статическое и динамическое методы	2
10	5	C# Инкапсуляция (знакомство)	2
11	5	C# Наследование (знакомство)	2
12	5	C# Полиморфизм (знакомство)	2
13	5	C# Интерфейсы (знакомство)	2
14	5	C# Целостность данных. Свойства Конструкторы	2
15	5	C# Структуры	2
16	5	Тестирование и отладка	2
Всего			32

4.5 Наименование тем лабораторных работ, их содержание и объем в часах

4.5.1. Первый семестр

№ п/п	№ раздела дисциплины	Наименование лабораторных работ	Трудоемкость (час)
1	2	Решение задач с данными простых типов на языке Паскаль	8
2	3	Решение задач с данными структурированных типов на языке Паскаль	8
3	3	Решение задач с разработкой функций и процедур на языке Паскаль	8
4	3	Решение задач с разработкой библиотек на языке Паскаль	8
	Всего		32

4.5.2. Второй семестр

№ п/п	№ раздела дисциплины	Наименование лабораторных работ	Трудоемкость (час)
1	4	Решение задач обработки массивов числовых данных на языке C#	8
2	4	Решение задач обработки текстовых данных на C#	8
3	4	Решение задач с использованием классов на C#	16
	Всего		32

5. Оценочные материалы для текущего контроля успеваемости и промежуточной аттестации по дисциплине

Для контроля результатов освоения дисциплины проводятся:

– тестирование.
– практические работы;
– защиты лабораторных работ;
– экзамен.

Примечание: оценочные материалы (типовые варианты тестов, контрольных работ и др.) приведены в приложении к рабочей программе дисциплины.

Промежуточная аттестация по итогам освоения дисциплины – экзамен.

6 Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература

1 Разумавская, Е. А. Алгоритмизация и программирование [Электронный ресурс] : практическое пособие / Е. А. Разумавская. — Электрон. текстовые данные. — СПб. : Санкт-Петербургский юридический институт (филиал) Академии Генеральной прокуратуры РФ, 2015. — 49 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/65427.html

2 Основы программирования на языке Паскаль. Основные понятия алгоритмического языка Паскаль [Электронный ресурс] : учебное пособие для самостоятельной работы по

дисциплине «Информатика» студентов 2-го курса всех направлений подготовки / сост. А. Д. Кононов, А. А. Кононов. — Электрон. текстовые данные. — Воронеж : Воронежский государственный архитектурно-строительный университет, ЭБС АСВ, 2017. — 53 с. — 978-5-7731-0504-6. — Режим доступа: http://www.iprbookshop.ru/72929.html
3 Павловская, Т. А. Программирование на языке высокого уровня C# [Электронный ресурс] / Т. А. Павловская. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 245 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73713.html

б) Дополнительная литература

1 Петров, В. Ю. Информатика. Алгоритмизация и программирование. Часть 1 [Электронный ресурс] : учебное пособие / В. Ю. Петров. — Электрон. текстовые данные. — СПб. : Университет ИТМО, 2016. — 93 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/66473.html
2 Тюльпинова, Н. В. Алгоритмизация и программирование [Электронный ресурс] : учебное пособие / Н. В. Тюльпинова. — Электрон. текстовые данные. — Саратов : Вузовское образование, 2019. — 200 c. — 978-5-4487-0470-3. — Режим доступа: http://www.iprbookshop.ru/80539.html
3 Павловская, Т. А. Программирование на языке высокого уровня Паскаль [Электронный ресурс] / Т. А. Павловская. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 153 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73714.html
4 Павловская, Т. А. Программирование на языке высокого уровня C# [Электронный ресурс] / Т. А. Павловская. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 245 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73713.html

в) методические указания:

1. Малина О.В. «Решение задач на языке программирования Паскаль» Методические указания к лабораторным работам для обучающихся по направлению 09.03.01 «Информатика и вычислительная техника», всех форм обучения при изучении дисциплины «Программирование». Ижевск: ИжГТУ, 2019 (Элект. издание) Рег.номер 044/53-ИИВТ

2. Малина О.В. «Решение задач на языке программирования C#» Методические указания к лабораторным работам для обучающихся по направлению 09.03.01 «Информатика и вычислительная техника», всех форм обучения при изучении дисциплины «Программирование». Ижевск: ИжГТУ, 2019 (Элект. издание) Рег.номер 043/53-ИИВТ

г) **перечень ресурсов информационно-коммуникационной сети Интернет**

1. Электронно-библиотечная система IPRbooks http://istu.ru/material/elektronno-bibliotechnaya-sistema-iprbooks
2. Электронный каталог научной библиотеки ИжГТУ имени М.Т. Калашникова Web ИРБИС http://94.181.117.43/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=IBIS&P21DBN=IBIS
3. Национальная электронная библиотека - http://нэб.рф
4. Мировая цифровая библиотека - http://www.wdl.org/ru
5. Международный индекс научного цитирования Web of Science - http://webofscience.com
6. Научная электронная библиотека eLIBRARY.RU – https://elibrary.ru/defaultx.asp
7. Справочно-правовая система КонсультантПлюс http://www.consultant.ru

д) лицензионное и свободно распространяемое программное обеспечение:

1. Microsoft Imagine Premium: Visual Studio
2. Microsoft Office Standard 2007
3. Doctor Web Enterprise Suite

7. Материально-техническое обеспечение дисциплины

1. Лекционные занятия.

Учебные аудитории для лекционных занятий укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории (наборы демонстрационного оборудования (проектор, экран, компьютер/ноутбук), учебно-наглядные пособия, тематические иллюстрации).

2. Практические занятия.

Учебные аудитории для практических занятий укомплектованы специализированной мебелью и техническими средствами обучения (проектор, экран, компьютер/ноутбук).

3. Лабораторные работы.

Для лабораторных занятий используется аудитория № 204, оснащенная следующим оборудованием: доской, компьютерами с возможностью подключения к сети «Интернет», столами, стульями.

4. Самостоятельная работа.

Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде ИжГТУ имени М.Т. Калашникова:

- научная библиотека ИжГТУ имени М.Т. Калашникова;
- помещение для самостоятельной работы обучающихся.

При необходимости рабочая программа дисциплины (модуля) может быть адаптирована для обеспечения образовательного процесса инвалидов и лиц с ограниченными возможностями здоровья, в том числе для обучения с применением дистанционных образовательных технологий. Для этого требуется заявление студента (его законного представителя) и заключение психолого-медико-педагогической комиссии (ПМПК).

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Ижевский государственный технический университет имени М.Т. Калашникова»

Оценочные средства по дисциплине

Программирование
наименование – полностью

направление 09.03.01 «Информатика и вычислительная техника»
код, наименование – полностью

профиль Автоматизированные системы обработки информации и управления
наименование – полностью

уровень образования: бакалавриат

форма обучения: очная
очная/очно-заочная/заочная

общая трудоемкость дисциплины составляет: 11 зачетных единиц

1. Оценочные средства

Оценивание формирования компетенций производится на основе результатов обучения, приведенных в п. 2 рабочей программы и ФОС. Связь разделов компетенций, индикаторов и форм контроля (текущего и промежуточного) указаны в таблице 4.2 рабочей программы дисциплины.

Оценочные средства соотнесены с результатами обучения по дисциплине и индикаторами достижения компетенций, представлены ниже.

№ п/п	Коды компетенции и индикаторов	Результат обучения *(знания, умения и навыки)*	Формы текущего и промежуточного контроля
1	ОПК-2.1 ОПК-2.2 ОПК-2.3	З8: возможности современных интегрированных сред программирования; У8: выбирать программные средства для решения задачи; Н1: навыками работы в интегрированной среде разработки;	Работа на практических занятиях: решение задач Тест Защита лабораторных работ Экзамен
2	ОПК-5.1 ОПК-5.2 ОПК-5.3	З10: инсталляция и параметрическая настройка интегрированной среды программирования У9: выполнять настройку среды программирования; Н2: навыками настройки среды разработки;	Работа на практических занятиях: решение задач Тест Защита лабораторных работ Экзамен
3	ОПК-8.1 ОПК-8.2 ОПК-8.3	З1: основные этапы решения задачи, понятия алгоритма, блок-схемы алгоритма, основные виды вычислительных процессов; З2: алфавит и лексику языков программирования, типы данных, правила записи выражений и операций; З3: операторы ввода-вывода, операторы ветвления, операторы цикла; З4: простые типы данных: множество значений, способ хранения, операции, особенности обработки; З5: структурированные ссылочные типы данных: множество значений, способ хранения, операции, особенности обработки; З6: основы процедурного программирования (процедуры, функции, библиотеки); З7: основы объектно-ориентированного программирования (классы, наследование, инкапсуляция, полиморфизм); З9: принципы рефакторинга, отладки, тестирования программы; У1: составлять алгоритмы решения задачи, оформлять блок-схемы алгоритмов; У2: выбирать необходимые типы данных для решения задачи; У3: реализовывать программы, реализующие линейные, разветвляющиеся и циклические вычислительные процессы; У4: обрабатывать структуры данных,	Работа на практических занятиях: решение задач Тест Защита лабораторных работ Экзамен

		файлы; У5: создавать программы на процедурном языке; У6: создавать программы с использованием идеологии ООП; У7: работать в среде программирования, реализовывать рефакторинг, отладку, тестирование программы; Н3: навыками алгоритмизации и программирования на процедурных и объектно-ориентированных языках; Н4: техникой рефакторинга, раздельной компиляции, отладчиком среды программирования, тестирования.	

Типовые задания для оценивания формирования компетенций

Наименование: экзамен

***Представление в ФОС*:** перечень вопросов

Перечень вопросов для проведения экзамена:

1) Паскаль. Типы данных в языках программирования.
2) Паскаль. Базовые типы данных.
3) Паскаль. Оператор присваивания.
4) Паскаль. Оператор условный.
5) Паскаль. Оператор выбора.
6) Паскаль. Оператор цикла с порядком.
7) Паскаль. Оператор цикла с предусловием.
8) Паскаль. Оператор цикла с постусловием.
9) Паскаль. Конструирование типов данных.
10) Паскаль. Перечислимый тип данных.
11) Паскаль. Ограниченный тип данных (диапазон).
12) Паскаль. Регулярный тип данных (массив).
13) Паскаль. Комбинированный тип данных (запись).
14) Паскаль. Множественный тип данных.
15) Паскаль. Модель ввода-вывода.
16) Паскаль. Двоичные (типизированные) файлы.
17) Паскаль. Текстовые файлы.
18) Паскаль. Расположение данных в текстовом файле.
19) Паскаль. Процедуры и функции. Назначение, принципы построения и использования.
20) Паскаль. Процедуры и функции. Передача параметров.
21) Библиотеки
22) Паскаль. Динамические структуры данных.
23) Паскаль. Обработка линейных списков.
24) Паскаль. Обработка очередей.
25) Паскаль. Обработка стеков.
26) Паскаль. Обработка деревьев.
27) C#. Базовые структуры данных.
28) C#. Класс TObject.
29) C# .Жизнь классов и объектов.
30) C# Поля и свойства
31) C# . Статические методы.
32) C# . Виртуальные методы.
33) C# . Динамические методы.
34) C# . Инкапсуляция.
35) C# . Наследование.
36) C# . Полиморфизм.
37) C#. Структуры.
38) C#. Интерфейсы.
39) C#. Массивы.
40) C#. Коллекции.

Критерии оценки:
Приведены в разделе 2

Наименование: тест
***Представление в ФОС*:** набор вариантов заданий
Варианты заданий:
по разделу «Процедурный язык программирования Паскаль»

Вариант № 18 Тест №1 Описательная часть программы на Паскале

1	Укажите строки на языке программирования Паскаль, написанные без ошибок	1	PROGRAM ольга1;
		2	PROGRAM OLGA1;
		4	PROGRAM ОЛЬГА1;
		8	PROGRAM 1OLGA;
2	Выбирете корректные утверждения для Паскаля	1	Идентификатор не может содержать пробел
		2	Идентификатор должен содержать специальные символы
		4	В качестве идентификаторов следует использовать зарезервированные слова
		8	Заглавные и прописные буквы в идентификаторе считаюся одинаковыми
3	Укажите строки на языке программирования Паскаль, написанные без ошибок	1	PROGRAM Пример 1
		2	PROGRAM Пример1
		4	PROGRAM Primer 1
		8	PROGRAM Primer1
4	Действительное положительное число в программе на Паскале может быть записано следующим образом:	1	78,5
		2	7,85E1
		4	78.5
		8	7.85E1
5	Блок TYPE необходим для описания	1	типов
		2	меток
		4	констант
		8	переменных

Вариант № 10 Тип Integer, boolean

1	Множество значений типа BOOLEAN	1	0 и 1
		2	true и false
		4	да и нет
		8	истина и ложь

2	Чему равны переменные a, k и m в резельтате выполнения программы:
	var a : boolean;
	k,m:integer;
	begin
	a:=true; k:=ord(a);
	m:=58 div (k+8); a:=m>k;
	end.
		1	a=1
			k=1 m=6

		2	a=true k=1 m=6
		4	a=false k=1 m=6
		8	a=true k=2 m=5
3	Какую функцию реализует MOD	1	находит остаток от деления
		2	находит квадрат числа
		4	находит частное
		8	находит целую часть от деления
4	Какие из указанных операций (функций) выполняются над данными типа BOOLEAN	1	<
		2	>
		4	ABS
		8	NOT

5	Какого типа может быть переменная в операторе READLN(a), чтобы не возникла ошибка	1	INTEGER
		2	BOOLEAN
		4	INTEGER или BOOLEAN
		8	не один из ранее перечисленных

Вариант № 34

1	Укажите операции (функции) над данными типа CHAR с результатом типа INTEGER	1	AND
		2	PRED
		4	=
		8	ORD
2	Какие значения примут переменные ch, n, в результате выполнения программы: var ch:CHAR; n:integer; begin n:=5; n:=SUCC(n); ch:=CHR(n); ch:=SUCC(ch); n:=ORD(ch); end.	1	ch=5 n='5'
		2	ch='5' n='5'
		4	ch=#5 n='5'
		8	ch="5" n='5'
3	Какие значения примут переменные ch, n, в результате выполнения программы: var ch:CHAR; n:integer; begin n:=5; n:=PRED(n); ch:=CHR(n); ch:=PRED(ch); n:=ORD(ch); end.	1	ch=3 n='3'
		2	ch='3' n='3'
		4	ch=#3 n='3'
		8	ch="3" n='3'
4	Какие значения примут переменные ch, n, в результате выполнения программы: var ch:CHAR; n:integer; begin n:=5; ch:=CHR(n); ch:=PRED(ch); n:=ORD(ch); end.	1	ch=#5 n=5
		2	ch='5' n=5
		4	ch=#6 n=6
		8	ch='6' n=6
5	Какие значения примут переменные ch, n, в результате выполнения программы: var ch:CHAR; n:integer; begin n:=5; n:=PRED(n); ch:=CHR(n); ch:=SUCC(ch); n:=ORD(ch); end.	1	ch=7 n='7'
		2	ch='7' n='7'
		4	ch=#7 n='7'
		8	ch="7" n='7'

Вариант № 34

1	Чему будет равна переменная f в результате выполнения программы: TYPE AA=(RED,BLUE,GREEN,BLACK); VAR pp1,pp2:AA; kk1,kk2:integer; f:boolean; begin pp1:=GREEN; pp2:=PRED(pp1); kk1:=ORD(pp1); kk2:=ORD(pp2); f:=kk1>=kk2; END.	1	1
		2	0
		4	true
		8	false
2	Какие процедуры и функции можно использовать для ввода и вывода данных интервального типа, заданного на базовом типе INTEGER	1	READ
		2	READLN
		4	READKEY
		8	WRITE
3	Какое описание переменной цикла оператора FOR можно	1	var PER:char;
		2	var PER:(red,dark, magenta);

	считать корректным	4	var PER:real;
		8	var PER:1..10;
4	Сколько раз напечатается слово ПРИВЕТ при выполнении программы: var i:boolean; begin FOR i:=false to true do write('ПРИВЕТ'); end.	1	вместо печати результата выдаст сообщение об ошибке
		2	1
		4	2
		8	3
5	Сколько раз напечатается слово ПРИВЕТ при выполнении программы: var i,k:integer; begin FOR i:=1 to 1 do begin FOR k:= 1 to 5 do write('ПРИВЕТ'); end; end.	1	0
		2	5
		4	15
		8	50

Вариант № 34

1	Укажите корректные фрагменты программы в которых нет ошибок трансляции	1	type a=set of char; VAR f:a; ch:char; begin f:=f+ch;end.
		2	type a=set of char; VAR f:a; ch:char; begin f:=f+(ch);end.
		4	type a=set of char; VAR f:a; ch:char; begin f:=f+[ch];end.
		8	type a=set of char; VAR f:a; ch:char; begin f:=f+{ch};end.
2	Укажите, что произойдет по итогу выполнения программы: type a=set of char; VAR f,g:a; bb:boolean; ch:char; begin f:=['a'..'k']; g:=['d'..'m']; bb:=f=g; ch:='m'; f:=g-[ch] ;end.	1	будет выдано сообщение об ошибке
		2	переменная f станет равна ['d'..'l'], переменная bb станет равна TRUE
		4	переменная f станет равна ['d'..'l'], переменная bb станет равна FALSE
		8	переменная f станет равна ['d'..'k','l'], переменная bb станет равна FALSE
3	Выбирете корректные утверждения для МНОЖЕСТВА в Паскале	1	Элементы в множестве не дублируются
		2	Элементы в множестве могут повторяться
		4	Множество может быть пустым
		8	Мощность множества -это количество его элементов
4	Укажите корректные фрагменты программы в которых нет ошибок трансляции	1	type aa=set of char; VAR a,b,c:aa; f:boolean; begin a:=b+c; f:=a<b; end.
		2	type aa=set of char; VAR a,b,c:aa; f:boolean; begin f:=b+c; c:=a<b; end.
		4	type aa=set of char; VAR a,b,c:aa; f:boolean; begin a:=a+c; f:=c<b; end.
		8	type aa=set of char; VAR a,b,c:aa; f:boolean; begin a:=b+f; f:=a<b; end.
5	Укажите, что произойдет по итогу выполнения программы: type a=set of char; VAR f,g:a; bb:boolean; ch:char; begin f:=['a'..'z']; g:=['d'..'m']; ch:='m'; bb:=[ch] IN f; f:=g-f ;end.	1	будет выдано сообщение об ошибке
		2	переменная f станет равна [], переменная bb станет равна TRUE
		4	переменная f станет равна [], переменная bb станет равна FALSE
		8	переменная f станет равна ['a'..'c', 'n'..'z'], переменная bb станет равна TRUE

Вариант № 1 Массив

Массив это -	1	упорядоченный набор однотипных элементов
	2	неупорядоченный набор однотипных элементов
	4	упорядоченный набор разнотипных элементов
	8	неупорядоченный набор разнотипных элементов
Укажите строки на языке программирования Паскаль, написанные без ошибок	1	TYPE cc=array[1..5] of integer; aa=array[1..7] of cc; VAR a:aa; begin a[6,4]:=5; end.
	2	TYPE cc=array[1..5] of integer; aa=array[1..7] of cc; VAR a:aa; begin a[4,6]:=5; end.
	4	TYPE cc=array[1..5] of integer; aa=array[1..7] of cc; VAR a:aa; begin a[5,3]:=5; end.
	8	TYPE cc=array[1..5] of integer; aa=array[1..7] of cc; VAR a:aa; begin a[3,5]:=5; end.

Чему равна переменная S по окончании программы: TYPE aa=array[1..5,'a'..'j'] of integer; VAR a:aa; i,j,s:integer; begin s:=0; for i:=1 to 5 do for j:='a' to 'j' do a[i,j]:=1; for i:=1 to 5 do for j:='a' to 'j' do s:=s+a[i,j]; end.	1	15
	2	50
	4	1
	8	10
Чему равна переменная S по окончании программы: TYPE aa=array['a'..'e',1..10] of integer; VAR a:aa; i,j,s:integer; begin s:=0; for i:='a' to 'e' do for j:=1 to 10 do a[i,j]:=1; for i:='a'..'e' do for j:=1..10 do s:=s+a[i,j]; end.	1	15
	2	50
	4	1
	8	10
В описании типа массив размерность массива может быть задана типами:	1	интервал на integer
	2	интервал на char
	4	интервал на перечисляемом типе
	8	интервал на real

Вариант № 23

1	Какое значение появится на экране после выполнения программы: VAR st1,st2:string; begin st1:='паль'; st2:='топаль'; st1:=st2-st1; writeln(st); end.	1	пальто
		2	топаль
		4	сообщение об ошибке
		8	то
2	Какое значение появится на экране после выполнения программы: VAR st,st3,st4:string; k:integer; begin k:=3; st3:='клав'; st4:='иатурая'; st:=concat(st3,st4); k:=copy(st,3,7); writeln(k); end.	1	кла
		2	клавивт
		4	авиатур
		8	сообщение об ошибке
3	Какое значение появится на экране после выполнения программы: VAR st,st3,st4:string; k:integer; begin st3:='ком'; st4:='мерция'; st:=concat(st3,st4); writeln(st); end.	1	комерция
		2	коммерция
		4	м
		8	сообщение об ошибке
4	Какое значение появится на экране после выполнения программы: VAR st:string; k:integer; n:real; begin st:= '34.51'; st:=val(st::5:1,n,k); writeln('число=',k); end.	1	число=34.51
		2	число=0
		4	число= 34.5
		8	сообщение об ошибке
5	Какое значение появится на экране после выполнения программы: VAR st, st3,st4:string; begin st3:='авто'; st4:='машина'; st:=st3+st4; if (st[3]>st[4])=true then writeln('привет')else begin if (st[3]<st[4])=true then writeln('Пока') else writeln('Ок'); end ; end.	1	Привет
		2	Пока
		4	Ок
		8	сообщение об ошибке

Вариант № 15 запись

1	При выполнении какого оператора программы: TYPE aa=array[1..3]of integer; bb=record x:real; y:aa; end; cc=record z:aa; v:bb;end; dd=array['a'..'d']of cc; var t:dd; I,j:integer; begin t['a'].v.x:='d'; t['b'].v.x:=5.6; t['c'].z[1]:=32; t['a'].v['a'].y:=55; end. возникнет ошибка?	1	t['a'].v.x:='d';
		2	t['b'].v.x:=5.6;
		4	t['c'].z[1]:=32;
		8	t['a'].v['a'].y:=55;

2	При выполнении какого оператора программы: TYPE aa=array[boolean]of integer; bb= set of char; cc=record a:aa; b:bb; c:char; end; dd=array[1..3] of cc; ff=record d:aa; e:integer; f:dd; end; Var t:ff; begin t.d[true]:=34; t.e:=45; t.f[2].a['b']:=67; t.f[1].b:=['a']; end. возникнет ошибка?	1	t.d[true]:=34;
		2	t.e:=45;
		4	t.f[2].a['b']:=67;
		8	t.f[1].b:=['a'];
3	При выполнении какого оператора программы: TYPE aa=array[boolean]of integer; bb= set of char; cc=record a:aa; b:bb; c:char; end; dd=array[1..3] of cc; ff=record d:aa; e:integer; f:dd; end; Var t:ff; begin t.d:=34; t.e:=4.55; t.f[2].a[false]:=67; t.f[1].b:=['a']; end. возникнет ошибка?	1	t.d:=34;
		2	t.e:=4.55;
		4	t.f[2].a[false]:=67;
		8	t.f[1].b:=['a'];
4	При выполнении какого оператора программы: TYPE aa=array[1..3]of integer; bb=record x:real; y:aa; end; cc=record z:aa; v:bb;end; dd=array['a'..'d']of cc; var t:dd; I,j:integer; begin t['a'].v.y[2]:='a'; t['b'].x:=5.7; dd['a'].v.y[2]:=5; dd['b'].x:=5.7; end. возникнет ошибка?	1	t['a'].v.y[2]:='a';
		2	t['b'].x:=5.7;
		4	dd['a'].v.y[2]:=5;
		8	dd['b'].x:=5.7;
5	При выполнении какого оператора программы: TYPE aa=array[boolean]of integer; bb= set of char; cc=record a:aa; b:bb; c:char; end; dd=array[1..3] of cc; ff=record d:aa; e:integer; f:dd; end; Var t:ff; begin t.d[true]:=34; t[true].e:=45; t.f[2].a[false]:=67; t.f[1].b:=['a']; end. возникнет ошибка?	1	t.d[true]:=34;
		2	t[true].e:=45;
		4	t.f[2].a[false]:=67;
		8	t.f[1].b:=['a'];

Вариант № 26

1	Выполнение каких операторов программы: var a,d,s:integer m:real; function PRIMER(s,d:real):integer; var a: real; i:integer; begin a:=0; for i:=1 to 5 do a:=s/d+a;PRIMER:=a; end; begin d:=10; s:=5; m:=PRIMER(d,s); write(m); end. вызовет ошибку?	1	function PRIMER(s,d:real):integer;
		2	a:=s/d+a;
		4	PRIMER:=a;
		8	m:=PRIMER(d,s);
2	При передаче данных через список параметров функция может вернуть в головную программу	1	одно значение
		2	не более двух значений
		4	не более трех значений
		8	необходимое количество значений
3	Выполнение каких операторов программы: var a,d,s:integer; m:real; function PRIMER:real; var i:integer; begin a:=0; for i:=1 to 5 do a:=s/d+a; PRIMER:=a; end; begin d:=10; s:=5; m:=PRIMER; write(m); end. вызовет ошибку?	1	function PRIMER:real;
		2	a:=s/d+a;
		4	PRIMER:=a;
		8	m:=PRIMER;
4	Выполнение каких операторов	1	function PRIMER(s,d:integer):real;

	программы: var a,d:real; s,m:real; function PRIMER(s,d:integer):real; var a: real; begin a:=s/d; PRIMER:=a; end; begin d:=10; s:=5; m:=PRIMER(d,s); write(m); end. вызовет ошибку?	2	a:=s/d;
		4	PRIMER:=a;
		8	m:=PRIMER(d,s);
5	Выполнение каких операторов программы: type aa=array[1..5] of integer; var k: aa;a,d,s:integer; m:real; procedure PRIMER(s:aa; var d:integer):integer; var a: real;i: integer; begin for i:=1 to 5 do d:=d+s[i]; end; begin for a:=1 to 5 do k[a]:=1; s:=5; d:=PRIMER(k,s); write(m); end. вызовет ошибку?	1	procedure PRIMER(s:aa; var d:integer):integer;
		2	for i:=1 to 5 do d:=d+s[i];
		4	for a:=1 to 5 do k[a]:=1;
		8	d:=PRIMER(k,s);

Вариант № 21

1	Чтобы описание типа было доступно и головной программе и подпрограмме, находящейся в библиотеке, оно должно быть выполнено:	1	в интерфейсной части дополнительной библиотеки, а в головной программе и в библиотеке, содержащей процедуру должно быть указано Uses имя библиотеки, где сделано описание типа
		2	после слова implementation библиотеки, где находится процедура, а в головной программе указано Uses имя библиотеки, где сделано описание типа
		4	в описательной части головной программы
		8	в интерфейсной части библиотеки содержащей процедуру, а в головной программе должно быть указано Uses имя библиотеки, где сделано описание типа
2	Программа: Program ppp; type dd=array[1..5] of real; var a,s:integer; m:real; d:dd; begin for a:=1 to 10 read(d[a]) s:=5; m:=PRIMER(s,d); write(m); end. Unit ppp2; interface function PRIMER(s:real; d:dd):real;implementation function PRIMER(s:real; d:dd):real; var a:real; i:integer; begin for i:=1 to 10 do a:=a+d[i]; PRIMER:=a/s; end; end. содержит ошибки. Какие из перечисленных утверждений корректны	1	в разделе interface не описано тело функции
		2	в описательной части головной программы нет ссылки на библиотеку ppp2
		4	в описательной части библиотеки нет ссылок на головную программу
		8	тип dd должен быть описан в библиотеке
3	Программа: program Primer11; var k: aa;a,d,s:integer; m:real; begin for a:=1 to 5 do k[a]:=1; s:=5; PRIMER(k,5); write(m); end. unit Bibl; interface type aa=array[1..5] of integer; procedure PRIMER(s:aa; var d:integer):integer; implementation procedure PRIMER(s:aa; var d:integer):integer;var a: real;i: integer; begin for i:=1 to 5 do d:=d+s[i]; end; end. содержит ошибки. Какие из перечисленных утверждений корректны	1	переменная k описана в головной программе, а должна быть описана в интерфейсной части библиотеки
		2	переменная s в списке формальных параметров процедуры должна быть описана с var
		4	в описательной части головной программы должно быть написано uses Bibl
		8	тип aa описан в интерфейсной части библиотеки Bibl, а должен быть описан в головной программе
4	Чтобы функция, находящаяся в библиотеке была доступна программе, находящейся в другом файле, необходимо:	1	чтобы ее заголовок был описан в разделе interface библиотеки, а полностью она должна быть описана в разделе implementation этой же библиотеки
		2	чтобы в головной программе в разделе uses было указано имя библиотеки
		4	чтобы в разделе uses библиотеки было указано имя головной программы

		8	чтобы в разделе uses головной программы было указано имя функции
5	Программа: program Primer11; type aa=array[1..5] of integer; var k: aa;a,d,s:integer; m:real; begin for a:=1 to 5 do k[a]:=1; s:=5; PRIMER(k,5); write(m); end. unit Bibl; interface procedure PRIMER(s:aa; var d:integer); implementation procedure PRIMER(s:aa; var d:integer); var a: real;i: integer; begin for i:=1 to 5 do d:=d+s[i]; end; end. содержит ошибки. Какие из перечисленных утверждений корректны	1	тип aa описан в головной программе, а должен быть описан в интерфейсной части библиотеки
		2	переменная d описана в головной программе, а дожна быть описана в процедуре
		4	переменная k описана в головной программе, а должна быть описана в интерфейсной части библиотеки
		8	переменная s в списке формальных параметров процедуры должна быть описана с var

Варианты заданий:
по разделу «Основы ООП»

Тест №1

1. Как в языке C# называют именованную последовательность инструкций 1 балл

- ○ Функция
- ○ Подпрограмма
- ○ Метод
- ○ Процедура

2. Какие утверждения верны? 1 балл

- ☐ Сборка — это как правило результат компиляции проекта
- ☐ Solution (решение) может содержать несколько проектов
- ☐ Сборка содержит в точности одно пространство имён
- ☐ В проекте может быть более одного кодового файла
- ☐ Разные проекты могут объявлять классы в одном и том же пространстве имён

3. Что перечисляется в секции References (ссылки) проекта в Visual Studio (или других IDE) 1 балл

- ○ Пространства имён, доступные для использования в кодовых файлах проекта
- ○ Сборки, классы которых доступны для использования в кодовых файлах проекта
- ○ Сборки, в которых классы этого проекта будут доступны
- ○ Пространства имён, определённые в этом проекте

4. Каково предназначение инструкции using в начале кодового файла? 1 балл

- ○ Подключает стороннюю библиотеку, открывая возможность пользоваться её классами
- ○ Избавляет программиста от необходимости указывать пространство имён перед именами классов данного пространства имён, сокращая код

5. Где найти exe-файл — результат компиляции моего проекта 1 балл

- ○ Скорее всего в подпапке bin/Debug папки вашего проекта
- ○ В текущей директории
- ○ Нигде, программа запускается, без создания exe-файла

Тест №2

```
1 if ([Нечто]) Console.WriteLine("OK");
```

1. Что может стоять на месте [Нечто]? Отметьте все верные утверждения.

- [ ] Сравнение чисел или строк
- [ ] Любое выражение
- [ ] Любое выражение типа bool
- [ ] Вызов метода, возвращающего bool
- [ ] Переменная, поле или свойство класса типа bool
- [ ] Выражение типа int
- [ ] Выражение типа string

```
1 if (!IsNotPrimeNumber(p)) Console.WriteLine(p);
```

2. Что не так в этом коде? 1 балл

- ( ) IsNotPrimeNumber слишком длинное название
- ( ) Лучше написать `if (IsNotPrimeNumber(p) == false) ...`
- ( ) Лучше вместо метода IsNotPrimeNumber написать метод IsPrimeNumber, тогда не будет двойного отрицания в условии

Известный анекдот про индусский код звучит так:

```
1 bool F(bool a)
2 {
3     if (a == true) return true;
4     else if (a == false) return false;
5     else return !true && !false;
6 }
```

Что же тут смешного?

3. Отметьте все верные утверждения про этот код 1 балл

- [ ] Выражение в последней ветви всегда равно false
- [ ] Если заменить выражение в последней else-ветви на true, то ничего не изменится
- [ ] Последняя else-ветвь никогда не выполняется
- [ ] Второй if можно заменить на `return false`
- [ ] Весь этот код можно заменить на `return a`

Тест №4

```
static void Main()
{
    var arr = new int[4];
    for (var i = 0; i <= arr.Length; i++)
        arr[i] = 0;
    DoSomething(arr);
}
```

1. Что не так в приведенном выше коде? Отметьте все подходящие варианты.

- [ ] Ничего, код великолепен!
- [ ] Нет смысла заполнять массив нулями в цикле
- [ ] Произойдет выход за границы массива
- [ ] Лучше было использовать foreach

2. В массив типа int[] можно положить элемент типа string

- ( ) Верно
- ( ) Неверно

3. Массивы отлично использовать для...

- [ ] хранения набора однотипных данных
- [ ] хранения набора разнотипных данных
- [ ] хранения разных свойств одного объекта
- [ ] поиска значения по его номеру

```
var a = new int[2, 3];
var b = new int[2][];
```

4. Отметьте все верные факты про массивы a и b

- [ ] Все ячейки массива `a` содержат значение 0
- [ ] Все ячейки массива `a` содержат значение null
- [ ] Все ячейки массива `b` содержат значение 0
- [ ] Все ячейки массива `b` содержат значение null
- [ ] Массив `a` двумерный
- [ ] Массив `b` двумерный
- [ ] Тип элементов `a` — int
- [ ] Тип элементов `b` — int[]

Тест №6

```
string result="";
for (int i=0; i<3; i++)
{
    result+=i.ToString() + ", ";
}
```

1. Этот код может быть существенно оптимизирован по производительности

- ○ Верно
- ○ Неверно

2. Выберите все верные утверждения: 1 балл

- ☐ При передаче строки в метод эта строка копируется
- ☐ Строка (string) эквивалентна массиву символов (char[])
- ☐ Строки являются типами-значениями (Value type)
- ☐ Строки являются типами-ссылками (Reference Type)
- ☐ Строки можно изменять во время выполнения

3. Что напечатает код Console.WriteLine("12345\n321");

- ○ одну строку: 12345321
- ○ одну строку: 12345n321
- ○ две строки: 12345 и 321
- ○ одну строку: 12345\n321
- ○ одну строку: 32145

4. Спецсимволы 1 балл

Перетащите блоки из правого столбца на свободные места.

\n		Слэш
\t		Перевод строки
\\		Возврат каретки
\r		Табуляция
\"		Кавычка

Тест №7

Изучите код ниже:

```
class Sample1 {
    static void F() {
        i = 1;
    }
    static int i = 0;
}
```

1. Корректно ли использование поля класса ДО его определения как в примере выше?

- Да, корректно
- Нет, это не скомпилируется
- Это скомпилируется, но при выполнении метода F возникнет ошибка

Локальные и глобальные переменные могут иметь одинаковые имена, а неоднозначности разрешаются в пользу локальных переменных. Область видимости какого-либо объекта почти всегда определяется фигурными скобками. Используя эти знания, изучите код ниже и ответьте на вопросы:

```
class Sample2 {
    static string who = "class";

    static void F() {
        string who = "F";
    }

    static void G() {
        F();
        Console.WriteLine(who);
    }

    static void H() {
        string who = "H";
        F();
        Console.Write(who);
    }
}
```

2. Что будет выведено на консоль при вызове метода G()? 1 балл

- class
- F
- H
- Ничего. Будет ошибка

Изучите следующий код:

```
class ClassA {
  public string FieldA;
}
class ClassB : ClassA {
  public string FieldB;
}
class ClassC : ClassA {
  public string FieldC;
}
class ClassD : ClassC {
  public string FieldD;
}
```

1. Какие поля содержатся в классе ClassA?

- [ ] FieldA
- [ ] FieldB
- [ ] FieldC
- [ ] FieldD

2. Какие поля содержатся в классе ClassB?

- [ ] FieldA
- [ ] FieldB
- [ ] FieldC
- [ ] FieldD

3. Какие поля содержатся в классе ClassC?

- [ ] FieldA
- [ ] FieldB
- [ ] FieldC
- [ ] FieldD

4. Какие поля содержатся в классе ClassD?

- [ ] FieldA
- [ ] FieldB
- [ ] FieldC
- [ ] FieldD

Локальные и глобальные переменные могут иметь одинаковые имена, а неоднозначности разрешаются в пользу локальных переменных. Область видимости какого-либо объекта почти всегда определяется фигурными скобками. Используя эти знания, изучите код ниже и ответьте на вопросы:

```
class Sample2 {
    static string who = "class";

    static void F() {
        string who = "F";
    }

    static void G() {
        F();
        Console.WriteLine(who);
    }

    static void H() {
        string who = "H";
        F();
        Console.Write(who);
    }
}
```

3. Что будет выведено при вызове метода H()? 1 балл

- ( ) class
- ( ) F
- ( ) H
- ( ) Ничего. Будет ошибка

```
class Sample3 {
    static string who = "class";

    static void Mixed() {
        Console.Write(who + " ");
        string who = "Mixed";
        Console.Write(who);
    }
}
```

4. Что будет выведено при вызове метода Mixed()?

- ○ class class
- ○ class Mixed
- ○ Mixed Mixed
- ○ Mixed class
- ○ Ничего. Будет ошибка компиляции

Локальные переменные и параметры метода с пересекающимися областями видимости не могут называться одинаково. Если это правило нарушается, то компиляция закончится с ошибкой. Например, если в методе F(int x) объявить локальную переменную x, то такой код не скомпилируется.

```
class Sample4 {
    void M1() {
        int i = 0;
        {
            int i = 1;
        }
    }
    void M2(int i) {
        int i = 0;
    }
    void M3() {
        // Фигурными скобками можно группировать операторы (правда почти никогда этого делать не стоит)
    // и каждая пара фигурных скобок создает свою область видимости.
        {
            int i = 0;
        }
        {
            int i = 1;
        }
    }
}
```

5. В каких методах компилятор сгенерирует ошибки компиляции? 1 балл

- ☐ M1
- ☐ M2
- ☐ M3

Тест №9

1. Динамический метод можно вызвать только в контексте объекта (экземпляра класса)

- ○ Верно
- ○ Неверно

2. Чем больше в классе динамических методов, тем больше места в памяти занимает каждый экземпляр этого класса 1 балл

- ○ Верно
- ○ Неверно

Изучите код ниже:

```
class SomeClass {
    public static int s = 1;
    public int d = 1;

    public void Run() {
        Console.Write(s + " " + d + " ");
        s++; d++;
    }

    public static void Main() {
        var object1 = new SomeClass();
        var object2 = new SomeClass();
        object1.Run();
        object2.Run();
        object1.Run();
    }
}
```

3. Что напечатает вызов метода Main в листинге выше?

4. Метод Run динамический?

- Верно
- Неверно

5. Метод Main статический?

- Верно
- Неверно

Тест №14

```
class A
{
    private int a;
    public int B;
    public void PublicMethod(){
        a = 1; // строка 1
        B = 2; // строка 2
    }
    private void PrivateMethod(){
        a = 3; // строка 3
        B = 4; // строка 4
    }
}

class B
{
    private A privateA = new A();
    public A PublicA = new A();

    public void M(){
        privateA.a = 5; // строка 5
        PublicA.a = 7;  // строка 6
        privateA.B = 6; // строка 7
        PublicA.B = 8;  // строка 8
        PublicA.PublicMethod();   // строка 9
        privateA.PublicMethod();  // строка 10
        PublicA.PrivateMethod();  // строка 11
        privateA.PrivateMethod(); // строка 12
    }
}
```

Тест №15

Изучите следующий код:

```
X[] xs = new X[10];
xs[0].A = 1;
```

1. Если этот код компилируется и работает, то чем может быть X?

- [ ] X может быть структурой
- [ ] X может быть классом
- [ ] X не может быть ни структурой, ни классом

Изучите следующий код:

```
X[] xs = new X[10];
xs[0] = new X();
```

2. Если этот код компилируется и работает, то чем может быть X?

- [ ] X может быть структурой
- [ ] X может быть классом
- [ ] X не может быть ни структурой, ни классом

Изучите следующий код:

```
1 X x;
2 x = new X();
```

3. Если этот код компилируется и работает, то чем может быть X?

- [x] X может быть структурой
- [x] X может быть классом
- [ ] X не может быть ни структурой, ни классом

Изучите следующий код:

```
1 X x;
2 x.A = 5;
```

4. Если этот код компилируется и работает, то чем может быть X?

- [x] X может быть структурой
- [ ] X может быть классом
- [ ] X не может быть ни структурой, ни классом

Тест №16

Вася написал такой код, но что-то опять пошло не так.

```
1  namespace Slide01
2  {
3      class Program
4      {
5          static void Main()
6          {
7              System.Console.WriteLine(Min(4, 2, 3));
8          }
9
10         static int Min(int a, int b, int c)
11         {
12             return Math.Min(a, Math.Min(b, c));
13         }
14     }
15 }
```

1. Как можно дополнить код, чтобы он начал компилироваться? Выберите все возможные варианты. 1 балл

- [ ] Дописать "using System.Console;" в начало
- [ ] Это скомпилируется, но при выполнении метода возникнет ошибка
- [ ] Дописать "using System;" в начало
- [ ] Дописать "using System.Math;" в начало
- [ ] Написать "System.Math.Min" вместо "Math.Min"

Компилируя *другую* программу Вася получил следующее сообщение при компиляции:

```
Controller.cs(9,4,9,15): error CS1501: No overload for method 'Min' takes 1 arguments
```

2. Что это может значить? Отметьте все корректные варианты. 1 балл

- [ ] Есть ошибка в девятой строке
- [ ] Вася снова забыл написать using
- [ ] Вася привел компилятору всего один аргумент, почему стоит компилировать эту программу. Этого явно мало!
- [ ] Вася попытался вызвать функцию Min с одним аргументом
- [ ] Ошибка в файле Controller.cs
- [ ] Ничего страшного, это сообщение можно просто игнорировать
- [ ] Ошибка в файле Min.cs

1. Чем автоматизированные тесты лучше ручного тестирования? 1 балл

- [ ] Автотесты выполняются быстрее, поэтому их можно запускать часто, но тратить время программиста только на непрошедшие тесты
- [ ] Работа автоматических тестов не зависит от концентрации, усидчивости и других личных качеств тестировщика
- [ ] Для больших программ полное ручное тестирование предполагает огромное количество ручной работы, поэтому проводится крайне редко

2. Почему тестирование важно? 1 балл

- [ ] При разработке веб-сервисов (например, заказа билетов) ошибки могут мгновенно приводить к финансовым убыткам разработчиков сервиса
- [ ] Тестирование ускоряет процесс разработки на начальном этапе
- [ ] Ошибки совершают даже очень опытные разработчики
- [ ] Некоторые ошибки могут привести к значительному ущербу, а мероприятия по их поиску и исправлению выгоднее исправления последствий
- [ ] Тестирование позволяет гарантировать, что в создаваемом ПО не будет ни одной ошибки

Критерии оценки:
Приведены в разделе 2

Наименование: защита лабораторных работ
***Представление в ФОС*:** задания и требования к выполнению представлены в методических указаниях по дисциплине
Варианты заданий:
Процедура SUM1, и функция SUM2 для внешнего использования
Процедура SUM1,для внешнего использования, функция SUM2 - для внутреннего
Подготовить сводную ведомость по результатам
Одномерные массивы
Многомерные массивы
Операции над данными типа множество
Глобальные переменные в подпрограмме
Организация вычислительного процесса. Процедуре PROCEDURE
Процедура. Передача данных через общую область
Описание переменной строкового типа
Типизированные файлы

Критерии оценки:
Приведены в разделе 2.

Наименование: работа на практических занятиях: решение задач**.**
***Представление в ФОС*:** сборник задач:

Целые числа
1. Вводится 100 чисел, в том числе число 0. Найти сумму всех положительных чисел, введенных после 0.
2. Вводятся числа. Конец ввода - 100. Найти сумму всех чисел, начиная с шестого.
3. Вводится 100 чисел. Найти среднее арифметическое отрицательных чисел
4. Вводятся числа, в том числе два числа 10. Конец ввода 0. Найти максимальное число, введенных между двумя 10.
5. Вводятся числа. Конец ввода - неубывающая последовательность из двух чисел. Найти среднее арифметическое трехзначных чисел.
6. Вводится 100 чисел. Найти максимальное отрицательное число

Символьные данные
1. Вводится текст конец ввода F4. Определить количество имен собственных в тексте (только первая буква большая). Слова разделяются одним пробелом.
2. Вводится текст конец ввода F5. Определить количество слов, начинающихся на букву А и состоящих из 3 букв. Слова раздеяются пробелами(может быть больше одного)
3. Вводится текст конец ввода F6. Определить количество слов, состоящих из 4 укв. Слова разделяются пробелом.

Множество:

1. Вводятся двузначные числа. Конец ввода – 0. Напечатать все введенные числа без повторений.
2. Вводится 10 двузначных чисел. Напечатать количество введенных чисел без повторений.
3. Вводиятся 15 двузначных чисел. Напечатать каждое число введенное впервые
4. Вводится 2 множеств. Каждое из 5 цифр. Напечатать числа второго множества, входящие в первое множество.
5. Вводятся два слова. Напечатать буквы первого слова, которых нет во втором.
6. Вводятся два слова. В каком слове больше разных букв.

Массив

1. Вводится массив 5*6. Найти среднее арифметическое элементов, лежащих выше главной диагонали.
2. Вводится массив 8*8. Сумму элементов над главной диагональю поделить на сумму элементов под побочной диагональю. Побочную диагональ отсортировать по убыванию.
3. Вводится массив 5*9. Упорядочить элементы столбца, содержащего наибольшее количество отрицательных чисел, по убыванию.
4. Вводится массив 5*5. Построить вектор длиной 9, элементы которого - максимумы элементов диагоналей, параллельных главной диагонали.

Строка

1. Вводится текст, конец ввода F5. Расставить слова по возрастанию количества букв.
2. Вводится текст, конец ввода F3. Расставить слова по убыванию количества букв.
3. Вводится текст, конец ввода F1. Заменить все пробелы на # и найти количество точек.

Запись (Паскаль)

1. Вводится информация об итогах зимней сессии на 1 курсе. Сведения о каждом студенте (всего их 25) заданы в виде следующего текста : «фамилия», «имя», «отчество», «год рождения», «номер группы», «оценка 1», «оценка 2», «оценка 3», причем первая оценка - за экзамен по высшей математике, вторая - по физике, третья - по программированию), «форма обучения (бюджетная, договорная)» Сведения о студентах отделены друг от друга точкой с запятой. В группе определить средний балл после зимней сессии, абсолютную успеваемость. Распечатать ФИО студентов по возрастанию среднего балла.
2. Вводится информация об итогах зимней сессии на 1 курсе. Сведения о каждом студенте (всего их 25) заданы в виде следующего текста : «фамилия», «имя», «отчество», «год рождения», «номер группы», «оценка 1», «оценка 2», «оценка 3», причем первая оценка - за экзамен по высшей математике, вторая - по физике, третья - по программированию), «форма обучения (бюджетная, договорная)» Сведения о студентах отделены друг от друга точкой с запятой. В группе студентов определить средний балл каждого, рассортировать по убыванию среднийбалла. Вывести ФИО студентов, у которых больше одной тройки.

Подпрограммы (Паскаль)

1. Вводится информация об итогах зимней сессии на 1 курсе. Сведения о каждом студенте (всего их 25) заданы в виде следующего текста : «фамилия», «имя», «отчество», «год рождения», «номер группы», «оценка 1», «оценка 2», «оценка 3», причем первая оценка - за экзамен по высшей математике, вторая - по физике, третья - по программированию), «форма обучения (бюджетная, договорная)» Сведения о студентах отделены друг от друга точкой с запятой. В группе **студентов рассортировать в алфавитном порядке по имени (процедура)**. Составить список студентов договорников и **определить у них средний балл (функция)**. Ввод - вывод в головной программе. Передача данных через список параметров.
2. Вводится информация об итогах зимней сессии на 1 курсе. Сведения о каждом студенте (всего их 25) заданы в виде следующего текста : «фамилия», «имя», «отчество», «год рождения», «номер группы», «оценка 1», «оценка 2», «оценка 3», причем первая оценка - за экзамен по высшей математике, вторая - по физике, третья - по программированию), «форма обучения (бюджетная,

договорная)» Сведения о студентах отделены друг от друга точкой с запятой. В группе студентов определить **средний балл по каждому предмету (функция)**. **Рассортировать ФИО в алфавитном порядке (процедура)**. Вывести на печать название предмета с лучшим средним баллом и список студентов. Ввод - вывод в головной программе. Передача данных через список параметров.

Классы (C#)

1. Вводится информация об итогах зимней сессии на 1 курсе. Сведения о каждом студенте (всего N) заданы в виде следующего текста: «фамилия», «имя», «отчество», «год рождения», «номер группы», «оценка 1», «оценка 2», «оценка 3», причем первая оценка - за экзамен по высшей математике, вторая - по физике, третья - по программированию), «форма обучения (бюджетная, договорная)». В группе определить а) средний балл после зимней сессии , абсолютную успеваемость. Распечатать ФИО студентов по возрастанию среднего балла. (Создать два класса Student и Group. Весь функционал распределить по классам. Методы и поля классов динамические)
2. Вводится информация об итогах зимней сессии на 1 курсе. Сведения о каждом студенте (всего N) заданы в виде следующего текста: «фамилия», «имя», «отчество», «год рождения», «номер группы», «оценка 1», «оценка 2», «оценка 3», причем первая оценка - за экзамен по высшей математике, вторая - по физике, третья - по программированию), «форма обучения (бюджетная, договорная)». В группе а)студентов рассортировать в алфавитном порядке по имени . Составить список студентов-договорников и б) определить у них средний балл. (Создать два класса Student и Group. Весь функционал распределить по классам. Методы и поля классов динамические)

Библиотеки

1. Вводится информация об итогах зимней сессии на 1 курсе. Сведения о каждом студенте (всего их 25) заданы в виде следующего текста: «фамилия», «имя», «отчество», «год рождения», «номер группы», «оценка 1», «оценка 2», «оценка 3», причем первая оценка - за экзамен по высшей математике, вторая - по физике, третья - по программированию), «форма обучения (бюджетная, договорная)». Сведения о студентах отделены друг от друга точкой с запятой. В группе студентов определить а) **средний балл по каждому предмету (функция)**. б) **Рассортировать ФИО в алфавитном порядке (процедура)**. Вывести на печать название предмета с лучшим средний баллом и список студентов. а)+б) - первая библиотека
Передача данных через список параметров. Ввод - вывод **реализован функциями, расположенными во второй библиотеке.**
2. Вводится информация об итогах зимней сессии на 1 курсе. Сведения о каждом студенте (всего их 25) заданы в виде следующего текста: «фамилия», «имя», «отчество», «год рождения», «номер группы», «оценка 1», «оценка 2», «оценка 3», причем первая оценка - за экзамен по высшей математике, вторая - по физике, третья - по программированию), «форма обучения (бюджетная, договорная)». Сведения о студентах отделены друг от друга точкой с запятой. а) **Рассортировать по возрасту (процедура)** и вывести на печать список студентов.
б) **Определить количество студентов бюджетной формы, у которых по математике 4 или 5 баллов (функция)**. а)+б) - первая библиотека
Передача данных через список параметров. Ввод - вывод **реализован функциями, расположенными во второй библиотеке.**

Файлы

1. Вводится информация об итогах зимней сессии на 1 курсе. Сведения о каждом студенте (всего их 25) заданы в виде следующего текста: «фамилия», «имя», «отчество», «год рождения», «номер группы», «оценка 1», «оценка 2», «оценка 3», причем первая оценка - за экзамен по высшей математике, вторая - по физике, третья - по программированию), «форма обучения (бюджетная, договорная)» Ввод каждого значения завершается нажатием <ENTER>. **Массив записей не использовать!!!** В группе определить средний балл после зимней сессии, абсолютную успеваемость. Распечатать ФИО студентов по возрастанию среднего балла.

2. Вводится информация об итогах зимней сессии на 1 курсе. Сведения о каждом студенте (всего их 25) заданы в виде следующего текста: «фамилия», «имя», «отчество», «год рождения», «номер группы», «оценка 1», «оценка 2», «оценка 3», причем первая оценка - за экзамен по высшей математике, вторая - по физике, третья - по программированию), «форма обучения (бюджетная, договорная)» Ввод каждого значения завершается нажатием <ENTER>. **Массив записей не использовать!!!** В группе студентов определить средний балл каждого. Распечатать список по убыванию среднего балла. Вывести ФИО студентов, у которых больше одной тройки.

Критерии оценки:
Приведены в разделе 2

Наименование: экзамен
***Представление в ФОС*:** перечень вопросов
Перечень вопросов для проведения экзамена:
1) Паскаль. Типы данных в языках программирования.
2) Паскаль. Базовые типы данных.
3) Паскаль. Оператор присваивания.
4) Паскаль. Оператор условный.
5) Паскаль. Оператор выбора.
6) Паскаль. Оператор цикла с порядком.
7) Паскаль. Оператор цикла с предусловием.
8) Паскаль. Оператор цикла с постусловием.
9) Паскаль. Конструирование типов данных.
10) Паскаль. Перечислимый тип данных.
11) Паскаль. Ограниченный тип данных (диапазон).
12) Паскаль. Регулярный тип данных (массив).
13) Паскаль. Комбинированный тип данных (запись).
14) Паскаль. Множественный тип данных.
15) Паскаль. Модель ввода-вывода.
16) Паскаль. Двоичные (типизированные) файлы.
17) Паскаль. Текстовые файлы.
18) Паскаль. Расположение данных в текстовом файле.
19) Паскаль. Процедуры и функции. Назначение, принципы построения и использования.
20) Паскаль. Процедуры и функции. Передача параметров.
21) Библиотеки
22) Паскаль. Динамические структуры данных.
23) Паскаль. Обработка линейных списков.
24) Паскаль. Обработка очередей.
25) Паскаль. Обработка стеков.
26) Паскаль. Обработка деревьев.

Критерии оценки:
Приведены в разделе 2

При проведении диагностики освоения компетенций и оценки минимального уровня знаний могут быть использованы тестовые материалы:

1. **Система программирования - то совокупность средств разработки программ, включающая (выбрать все правильны варианты):**

1) языки программирования;
2) текстовые редакторы;
3) трансляторы (компилятор/интерпретатор);

4) библиотеки подпрограмм;
5) утилиты и обслуживающие программы.

2. **По парадигме программирования языки программирования подразделяются на (указать все правильные варианты):**

1) функциональные;
2) логические;
3) операторные;
4) универсальные.

3. **По направлению ориентации языка программирования выделяются следующие классификационные группы (выбрать все существующие):**
1) машинно-ориентированные;
2) процедурно-ориентированные (универсальные);
3) проблемно-ориентированные;
4) объектно-ориентированные;
5) сверхуниверсальные.

4. **По способу выполнения программ C# относится к языкам программирования:**
1) компилируемым,
2) интерпретируемым,
3) компилируемые на основе псевдокода,
4) совмещенным, в которых исходный код проходит две стадии обработки.

5. **По факту созданию процесса языки программирования делятся на создающие процесс и сценарные.**
Python относится (выбрать правильный вариант):
1) к языкам, создающим процесс;
2) к сценарным языкам.

6. **К языкам высокого уровня относятся (выбрать правильные варианты):**
1) Java
2) C++
3) Ассемблер
4) Python
5) C#

7. **Декларативное программирование – это**:
1) это программирования, реализующее такой способ кодирования алгоритма, в результате которого получается понятная для компьютера запись алгоритма – программа, состоящая из операторов;
2) это парадигма программирования, в которой задаётся спецификация решения задачи, то есть описывается, что представляет собой проблема и ожидаемый результат.

Ключ теста:

Вопрос	1	2	3	4	5	6	7
Ответ	1,2,3,4,5	1,2,3	1,2,3,4,5	4	2	1,2,4,5	2

1. Как в языке C# называют именованную последовательность инструкций?
1) функция;
2) подпрограмма;

3) метод;
4) процедура

2. **Какие утверждения верны?**

1) Сборка – это, как правило, результат компиляции проекта.
2) Solution (решение) может содержать несколько проектов.
3) Сборка содержит в точности одно пространство имен.
4) В проекте может быть более одного кодового файла.
5) Разные проекты могут объявлять классы в одном и том же пространстве имён

3. Что перечисляется в секции References (ссылки) проекта в Visual Studio (или других IDE)?

1) Пространства имён, доступные для использования в кодовых файлах проекта.
2) Сборки, классы которых доступны для использования в кодовых файлах проекта.
3) Сборки, в которых классы этого проекта будут доступны.
4) Пространства имён, определённые в этом проекте.

4. **Каково предназначение инструкции using в начале кодового файла, написанного на языке программирования C#?**

1) Подключает стороннюю библиотеку, открывая возможность пользоваться её классами.
2) Избавляет программиста от необходимости указывать пространство имён перед именами классов данного пространства имён, сокращая код

5. **Где найти exe-файл — результат компиляции вашего проекта в Visual Studio?**

1) Скорее всего в подпапке bin/Debug папки вашего проекта.
2) В текущей директории.
3) Нигде, так как программа запускается без создания exe- файла.

6. **Зачем нужно переносить часть кода в библиотеки — отдельные проекты, компилирующиеся в dll-файлы?**

1) Это ускоряет выполнение программы.
2) Это необходимо для написания модульных тестов.
3) Это ускоряет компиляцию.
4) Чтобы можно было повторно использовать код из библиотек в других проектах.

7. **Почему стоит отделять в своем коде логику от ввода и вывода?**

1) Это незыблемое правило, которому нужно следовать всегда без обсуждений и раздумий.
2) Логику, отделенную от ввода-вывода проще тестировать.
3) Тестировать программу через ввод-вывод невозможно
4) Логика может пригодиться в других обстоятельствах, а если она будет объединена с вводом-выводом, ее не получится использовать повторно.

8. **Чтобы создать модульный тест с помощью встроенной в Visual Studio системы тестирования, нужно:**

1) Создать новый проект с типом Test Project.
2) Добавить в этом проекте ссылку на сборку с тестируемым кодом.
3) Создать класс с тестами и пометить его атрибутом [TestClass].
4) Создать в классе нестатический public void-метод без параметров.
5) Пометить этот метод атрибутом [TestMethod].
6) Написать код подготовки исходных данных, код вызова тестируемого метода и код проверки результата вызова

9. **Достоинства платформы .NET:**

1) Цельная объектно-ориентированная модель программирования, упрощающая разработку программ.
2) Многоплатформенность приложений.
3) Автоматическое управление ресурсами.
4) Упрощение развертывания приложений.
5) Современная модель безопасности развертывания и сопровождения кода.
6) Полный отказ от реестра.

10. **Укажите порядковый номер каждого этапа в процессе установки и настройки Visual Studio для C#.**

1) Настройка нового проекта, назначение ему имени и местоположение для хранения файлов проекта - _____
2) Выбор цветовой темы, и запуск кнопкой "Start Visual Studio"- _____
3) Выбор опции "NET desktop development" и нажатие "Install" - _____
4) Выбор опции Create a new project и настройка среды C# - _____
5) Начало загрузки Visual Studio, все версии которой доступны на visualstudio.microsoft.com - _____
6) Вход в свою учетную запись Microsoft или ее создание - _____
7) Выбор опции Console App for C# для работы с консольными приложениями для C# - _____

Ключ теста:

Вопрос	1	2	3	4	5	6	7	8	9	10
Ответ	3	1,2,4,5	2	2	1	4	2,4	1,2,3,4,5,6	1,2,3,4,5,6	7-4-2-5-1-3-6

1. **Какие типы ошибок могут возникать в процессе разработки программы?**

1) Ошибки компиляции.
2) Ошибки выполнения.
3) Стилистические ошибки.
4) Алгоритмические ошибки.
5) Ошибки проектирования.

2. **Что может стоять на месте [Нечто] в выражении (отметить верные выриантты): if ([Нечто]) Console.WriteLine("OK");**

1) Сравнение чисел или строк.
2) Любое выражение.
3) Любое выражение типа bool.
4) Вызов метода, возвращающего bool.
5) Переменная, поле или свойство класса типа bool.
6) Выражение типа int.
7) Выражение типа string.

3. **Какие типы данных являются ссылочными?**

1) int;
2) double;
3) int[];
4) double[];
5) string;
6) string[].

4. **Какие из ниже приведенных операторов являются операторами цикла?**

1) if;
2) while;
3) for;
4) foreach;
5) switch

5. **Выберите все верные утверждения:**

1) При передаче строки в метод эта строка копируется.
2) Строка (string) эквивалентна массиву символов (char[]).
3) Строки являются типами-значениями (Value type).
4) Строки являются типами-ссылками (Reference Type).
5) Строки можно изменять во время выполнения.

6. **Отметьте все верные факты про кодировки**:

1) Кодировка — это способ преобразования символов в байты и обратно.
2) Кодировка — это только лишь сопоставление номера каждому символу.
3) В кодировке UTF-8 все символы кодируются более, чем одним байтом.
4) Формат записи латинских букв в кодировке UTF-8 совпадает с форматом в старой кодировке ASCII.

7. **Уровень покрытия кода тестами показывает ...**

1) ... долю корректных строк кода тестируемой программы.
2) ... долю потенциальных ошибок, найденных комплектом тестов.
3) ... долю строк кода тестируемой программы, выполнившихся при запуске комплекта тестов.

8. **Отметьте верные утверждения про анализ покрытия кода тестами:**

1) Он позволяет найти участки кода, которые еще не тестируются.
2) Он не позволяет найти ошибки, связанные с тем, что некоторые особые случаи не рассмотрены.
3) В простых случаях этот анализ можно провести с помощью точек останова и отладчика.
4) Чем выше процент кода, покрытого тестами, тем корректнее работает код.

9. **Из каких этапов состоит теоретический анализ алгоритма?**

1) Доказательство того, что алгоритм остановится.
2) Доказательство того, что алгоритм вычисляет верный результат.
3) Оценка сложности алгоритма.
4) Оптимизация алгоритма.
5) Тестирование алгоритма.
6) Отладка алгоритма.

10. **Изучите код ниже:**

class Sample1 { static void F() {i = 1;} static int i = 0;}

Корректно ли использование поля класса ДО его определения как в примере выше?

1) Да, корректно.
2) Нет, это не скомпилируется.
3) Это скомпилируется, но при выполнении метода F возникнет ошибка.

11. **Изучите код ниже:**

```
class Sample2 {
  static string who = "class";

  static void F() {
        string who = "F";
  }

  static void G() {
        F();
        Console.WriteLine(who);
  }

  static void H() {
        string who = "H";
        F();
        Console.Write(who);
  }
}
```

А) Что будет выведено на консоль при вызове метода G()?

1) class;
2) F;
3) H;
4) Ничего. Будет ошибка.

Б) **Что будет выведено на консоль при вызове метода H()?**

1) class;
2) F;
3) H;
4) Ничего. Будет ошибка.

12. **Изучите код ниже:**

```
class Sample3 {
  static string who = "class";

  static void Mixed() {
        Console.Write(who + " ");
        string who = "Mixed";
        Console.Write(who);
  }
}
```

Что будет выведено при вызове метода Mixed()?

1) class class
2) class Mixed
3) Mixed Mixed
4) Mixed class
5) Ничего. Будет ошибка компиляции

Ключ теста:

Вопрос	1	2	3	4	5	6	7	8	9	10	11		12
											А	Б	
Ответ	1,2,3,4,5	1,3,4,5	3,4,5,6	2,3,4	4	1,4	3	1,2,3	1,2,3	1	1	3	5

2. Критерии и шкалы оценивания

Для контрольных мероприятий (текущего контроля) устанавливается минимальное и максимальное количество баллов в соответствии с таблицей. Контрольное мероприятие считается пройденным успешно при условии набора количества баллов не ниже минимального.

Результат обучения по дисциплине считается достигнутым при успешном прохождении обучающимся всех контрольных мероприятий, относящихся к данному результату обучения.

1 семестр:

Разделы дисциплины	***Форма контроля***	***Количество баллов***	
		min	***max***
2	Лабораторная работа № 1	10	20
3	Лабораторная работа № 2	10	20
3	Лабораторная работа № 3	15	30
3	Лабораторная работа № 4	15	30
	Итого:	**50**	**100**

При оценивании результатов обучения по дисциплине в ходе текущего контроля успеваемости используются следующие критерии. Минимальное количество баллов выставляется обучающемуся при выполнении всех показателей, допускаются несущественные неточности в изложении и оформлении материала.

Наименование, назначение	***Показатели выставления минимального количества баллов***
Лабораторная работа	Лабораторная работа выполнена в полном объеме; Представлен отчет, содержащий необходимые этапы, выводы, оформленный в соответствии с установленными требованиями; Продемонстрирован удовлетворительный уровень владения материалом при защите лабораторной работы, даны правильные ответы не менее чем на 50% заданных вопросов.
Практическая работа	Задания выполнены более чем наполовину. Присутствуют серьёзные ошибки. Продемонстрирован удовлетворительный уровень владения материалом. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий. На защите практической работы даны правильные ответы не менее чем на 50% заданных вопросов
Тест	Правильно решено не менее 50% тестовых заданий

Промежуточная аттестация в 1 семестре проводится в форме экзамена.

Итоговая оценка по дисциплине может быть выставлена на основе результатов текущего контроля с использованием следующей шкалы:

Оценка	***Набрано баллов***
«неудовлетворительно»	менее 39
«удовлетворительно»	39-50
«хорошо»	51-84
«отлично»	85-100

Если сумма набранных баллов менее 39 – обучающийся не допускается до промежуточной аттестации.

Если сумма баллов составляет от 39 до 84 баллов, обучающийся допускается до экзамена.

По сумме набранных баллов студенту может быть выставлена оценка за промежуточную аттестацию, согласно приведенной шкале. Обучающийся имеет право сдать экзамен в устной форме для изменения балла.

Промежуточная аттестация проводится в устной форме.

Билет к экзамену включает 2 теоретических вопроса.

Время на подготовку: 30 минут.

Промежуточная аттестация в 2 семестре проводится в форме экзамена

Оценка	*Набрано баллов*
«отлично»	73-90
«хорошо»	64-72
«удовлетворительно»	55-63
«неудовлетворительно»	45-55

Если сумма набранных баллов менее 54 – обучающийся не допускается до промежуточной аттестации.

Если сумма баллов более 55, обучающийся допускается до экзамена, при условии что выполнены и защищены лабораторные работы.

По сумме набранных баллов студенту может быть выставлена оценка за промежуточную аттестацию, согласно приведенной шкале. Обучающийся имеет право сдать экзамен в письменной форме для изменения балла.

Промежуточная аттестация проводится в письменной форме.

Билет к экзамену включает 2 теоретических вопроса.

Время на подготовку: 45 минут.

При оценивании результатов обучения по дисциплине в ходе промежуточной аттестации используются следующие критерии и шкала оценки:

Оценка	*Критерии оценки*
«отлично»	Обучающийся показал всестороннее, систематическое и глубокое знание учебного материала, предусмотренного программой, умение уверенно применять на их практике при решении задач (выполнении заданий), способность полно, правильно и аргументировано отвечать на вопросы и делать необходимые выводы. Свободно использует основную литературу и знаком с дополнительной литературой, рекомендованной программой
«хорошо»	Обучающийся показал полное знание теоретического материала, владение основной литературой, рекомендованной в программе, умение самостоятельно решать задачи (выполнять задания), способность аргументировано отвечать на вопросы и делать необходимые выводы, допускает единичные ошибки, исправляемые после замечания преподавателя. Способен к самостоятельному пополнению и обновлению знаний в ходе дальнейшей учебной работы и профессиональной деятельности
«удовлетворительно»	Обучающийся демонстрирует неполное или фрагментарное знание основного учебного материала, допускает существенные ошибки в его изложении, испытывает затруднения и допускает ошибки при выполнении заданий (решении задач), выполняет задание при подсказке преподавателя, затрудняется в формулировке выводов. Владеет знанием основных разделов, необходимых для дальнейшего обучения, знаком с основной и дополнительной литературой, рекомендованной программой
«неудовлетворительно»	Обучающийся при ответе демонстрирует существенные пробелы в знаниях основного учебного материала, допускает грубые ошибки в формулировании основных понятий и при решении типовых задач (при выполнении типовых заданий), не способен ответить на наводящие вопросы преподавателя. Оценка ставится обучающимся, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании образовательного учреждения без дополнительных занятий по рассматриваемой дисциплине