

МИНОБРНАУКИ РОССИИ
Воткинский филиал
Федерального государственного бюджетного образовательного
учреждения высшего образования
«Ижевский государственный технический университет имени М.Т. Калашникова»
(ВФ ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

УТВЕРЖДАЮ

Директор



И.А. Давыдов

10 апреля 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Программирование

направление 09.03.01 «Информатика и вычислительная техника»

профиль «Автоматизированные системы обработки информации и управления»

уровень образования: бакалавриат

форма обучения: очная

общая трудоемкость дисциплины составляет: 11 зачетных единиц(ы)

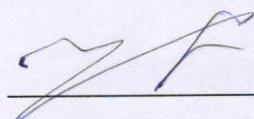
Кафедра Естественные науки и информационные технологии

Составитель _____

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования и рассмотрена на заседании кафедры

Протокол от «10» апреля 2025 г. № 3

Заведующий кафедрой

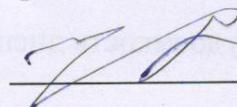
 К.Б. Сентяков

«10^я апреля 2025 г.

СОГЛАСОВАНО

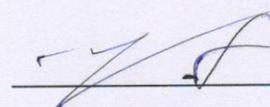
Количество часов рабочей программы и формируемые компетенции соответствуют учебному плану направления 09.03.01 «Информатика и вычислительная техника», профиль «Автоматизированные системы обработки информации и управления»

Председатель учебно-методической комиссии по направлению 09.03.01 «Информатика и вычислительная техника», профиль «Автоматизированные системы обработки информации и управления»

 К.Б. Сентяков

«10^я апреля 2025 г.

Руководитель образовательной программы

 К.Б. Сентяков

«10^я апреля 2025 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 | Целью преподавания дисциплины является 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 устанавливать 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; установка и настройка программной среды; основы ООП; |
| Форма промежуточной аттестации | Экзамен (1 сем), экзамен (2 сем) |

Целью преподавания 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 Turbo Pascal и Visual Studio;
- получение навыков оформления программной документации.

В результате изучения дисциплины студент должен

знать:

- основные этапы решения задачи, понятия алгоритма, блок-схемы алгоритма, основные виды вычислительных процессов;
- алфавит и лексику языков программирования, типы данных, правила записи выражений и операций;
- операторы ввода-вывода, операторы ветвления, операторы цикла;
- простые типы данных: множество значений, способ хранения, операции, особенности обработки;
- структурированные ссылочные типы данных: множество значений, способ хранения, операции, особенности обработки;
- основы процедурного программирования (процедуры, функции, библиотеки);
- основы объектно-ориентированного программирования (классы, наследование, инкапсуляция, полиморфизм);
- возможности современных интегрированных сред программирования;
- принципы рефакторинга, отладки, тестирования программы;
- инсталляция и параметрическая настройка интегрированной среды программирования

уметь:

- составлять алгоритмы решения задачи, оформлять блок-схемы алгоритмов;
- выбирать необходимые типы данных для решения задачи;
- реализовывать программы, реализующие линейные, разветвляющиеся и циклические вычислительные процессы;
- обрабатывать структуры данных, файлы;
- создавать программы на процедурном языке;
- создавать программы с использованием идеологии ООП;
- работать в среде программирования, реализовывать рефакторинг, отладку, тестирование программы;
- выбирать программные средства для решения задачи; - выполнять настройку среды программирования; **владеть:**
- навыками работы в интегрированной среде разработки;
- навыками настройки среды разработки;
- навыками алгоритмизации и программирования на процедурных и объектноориентированных языках;
- техникой рефакторинга, отдельной компиляции, отладчиком среды программирования, тестирования.

2 Место дисциплины в структуре ООП

Дисциплина относится к обязательной части Блока 1 «Дисциплины (модули)» ООП.

Для изучения дисциплины студент должен:

знать:

- основные понятия дисциплин: Информатика, Математика;

уметь:

- использовать полученные ранее знания в практических задачах;

владеть:

- навыками работы с персональным компьютером на высоком пользовательском уровне;
- основами работы с научно-технической литературой.

Изучение дисциплины базируется на знаниях, полученных при изучении дисциплин: Информатика, Математика (среднее(полное) общее образование).

3 Требования к результатам освоения дисциплины**3.1 Знания, приобретаемые в ходе изучения 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 |

3.2 Умения, приобретаемые в ходе изучения 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 |

3.3 Навыки, приобретаемые в ходе изучения дисциплины

| № п/п | Навыки |
|-------|--|
| 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, отдельной компиляции, отладчиком среды программирования, тестирования. |

3.4 Компетенции, приобретаемые в ходе изучения дисциплины

| Компетенции | | Знания (№№ из 3.1) | Умения (№№ из 3.2) | Навыки (№№ из 3.3) |
|---|--|--------------------------|--------------------------|--------------------------|
| 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-2.3 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности | 8 | 8 | 1 |
| ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем; | ОПК-5.1 Знать: 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 установки 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 |

4 Структура и содержание дисциплины

4.1 Разделы дисциплин и виды занятий

| № п/п | Раздел дисциплины | Семестр | Неделя семестра | Виды контактной работы, включая самостоятельную работу студентов и трудоемкость (в часах) | | | | Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам) |
|-------|--|---------|-----------------|---|------|-----|-----|---|
| | | | | лек | прак | лаб | СРС | |
| 1 | Обзор существующих программных сред, языков программирования, их классификация | 1 | 1 | 2 | 2 | | | |
| 2 | Алгоритмизация | 1 | 2 | | 2 | 4 | 4 | работа на практических занятиях: текущий контроль решения задач; выполнение лабораторной работы |
| | | 1 | 3 | 2 | 2 | | 4 | работа на практических занятиях: текущий контроль решения задач |
| | | 1 | 4 | | 2 | 4 | 4 | работа на практических занятиях: текущий контроль решения задач; защита лабораторной работы |
| 3 | Процедурный язык программирования Паскаль | 1 | 5 | 2 | 2 | | 3 | Тест, защита лабораторной работы |
| | | 1 | 6 | | 2 | 4 | 3 | Тест, выполнение лабораторной работы |
| | | 1 | 7 | 2 | 2 | | 3 | Тест |
| | | 1 | 8 | | 2 | 4 | 3 | Тест, защита лабораторной работы |
| | | 1 | 9 | 2 | 2 | | 3 | Тест |
| | | 1 | 10 | | 2 | 4 | 7 | Тест, выполнение лабораторной работы |
| | | 1 | 11 | 2 | 2 | | 5 | Тест |
| | | 1 | 12 | | 2 | 4 | 5 | Тест, защита лабораторной работы |
| | | 1 | 13 | 2 | 2 | | 5 | Тест |
| | | 1 | 14 | | 2 | 4 | 5 | Тест, выполнение лабораторной работы |
| | | 1 | 15 | 2 | 2 | | 5 | работа на практических занятиях: текущий контроль решения задач |
| | | | | | | | 36 | Экзамен |
| | Итого за 1 семестр | | | 16 | 32 | 32 | 100 | |
| | В том числе контроль самостоятельной работы | | | | 2 | | | |

| | | | | | | | | |
|---|---|---|----|-----------|-----------|-----------|------------|---|
| 4 | Инсталляция и настройка программной среды | 2 | 1 | 2 | 2 | | 8 | |
| 5 | Основы ООП | 2 | 2 | | 2 | 4 | 6 | Тест; выполнение лабораторной работы |
| | | 2 | 3 | 2 | 2 | | 6 | Тест |
| | | 2 | 4 | | 2 | 4 | 6 | работа на практических занятиях: текущий контроль решения задач; защита лабораторной работы |
| | | 2 | 5 | 2 | 2 | | 6 | Тест |
| | | 2 | 6 | | 2 | 4 | 6 | работа на практических занятиях: текущий контроль решения задач; выполнение лабораторной работы |
| | | 2 | 7 | 2 | 2 | | 6 | Тест |
| | | 2 | 8 | | 2 | 4 | 6 | Тест; защита лабораторной работы |
| | | 2 | 9 | 2 | 2 | | 6 | работа на практических занятиях: текущий контроль решения задач; |
| | | 2 | 10 | | 2 | 4 | 6 | Тест; выполнение лабораторной работы |
| | | 2 | 11 | 2 | 2 | | 6 | работа на практических занятиях: текущий контроль решения задач |
| | | 2 | 12 | | 2 | 4 | 6 | работа на практических занятиях: текущий контроль решения задач; выполнение лабораторной работы |
| | | 2 | 13 | 2 | 2 | | 6 | работа на практических занятиях: текущий контроль решения задач; |
| | | 2 | 14 | | 2 | 4 | 6 | работа на практических занятиях: текущий контроль решения задач; выполнение лабораторной работы |
| | | 2 | 15 | 2 | 2 | | 6 | Тест |
| | | 2 | 16 | | 2 | 4 | 8 | Тест; защита лабораторной работы |
| | | | | | | | 36 | Экзамен |
| | Итого за 2 семестр | | | 16 | 32 | 32 | 136 | |
| | В том числе контроль самостоятельной работы | | | | 2 | | | |

4.2 Содержание разделов курса

| № п/п | Раздел дисциплины | Знания (номер из 3.1) | Умения (номер из 3.2) | Навыки (номер из 3.3) |
|----------|---|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| 1 | 1. Классификация языков программирования | 8 | 8 | 1 |
| 2 | 1. Основные этапы решения задачи 2. Основные виды вычислительных процессов (разветвления, циклы) 3. Базовые алгоритмы (сумма, максимум, минимум) и методы решения | 1 | 1 | 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) | 2-6 | 2-5 | 3 |
| 4 | 1. Инсталляция и настройка программной среды | 10 | 9 | 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) | 2-5, 7,9 | 2-4 6-7 | 3-4 |

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

4.3.1. Первый семестр

| № | № раздела дисциплины | Название практических работ | Объем в часах |
|-------|----------------------|--|---------------|
| 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 |

Второй семестр

| № | № раздела дисциплины | Название практических работ | Объем в часах |
|-------|----------------------|--|---------------|
| 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 |

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

4.4.1. 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 |

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 Содержание самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

5.1 Содержание самостоятельной работы 5.1.1. Первый семестр

| № п/п | № раздела дисциплины | Наименование тем | Трудоёмкость (час) |
|-------|----------------------|--|--------------------|
| 1 | 2 | Основные виды вычислительных процессов, виды блоков блок-схемы | 3 |
| 2 | 2 | Разветвляющиеся вычислительные процессы | 3 |
| 3 | 2 | Циклические вычислительные процессы | 3 |
| 4 | 3 | Целочисленный тип данных языка программирования Паскаль | 3 |
| 5 | 3 | Символьные тип данных языка программирования Паскаль | 4 |
| 6 | 3 | Логический тип данных языка программирования Паскаль | 4 |
| 7 | 3 | Вещественный тип данных языка программирования Паскаль | 4 |
| 8 | 3 | Тип Множество языка программирования Паскаль | 4 |
| 9 | 3 | Тип Массив языка программирования Паскаль | 4 |
| 10 | 3 | Тип Строка языка программирования Паскаль | 4 |
| 11 | 3 | Тип Запись языка программирования Паскаль | 4 |
| 12 | 3 | Функции процедуры языка программирования Паскаль | 5 |
| 13 | 3 | Библиотеки языка программирования Паскаль | 4 |
| 14 | 3 | Типизированные файлы (последовательный и прямой доступ) | 4 |
| 15 | 3 | Текстовые файлы | 4 |
| 16 | 3 | Динамические связные списки (стеки, очереди, деревья) | 4 |
| 17 | 1-3 | Экзамен | 36 |
| | ВСЕГО | | 100 |

Второй семестр

| № п/п | № раздела дисциплины | Наименование тем | Трудоёмкость (час) |
|-------|----------------------|---|--------------------|
| 1 | 4 | Система программирования Visual Studio – инсталляция и настройка | 8 |
| 2 | 5 | С# Основные понятия. Базовые структуры данных. Разветвляющиеся вычислительные процессы. Циклы | 6 |
| 3 | 5 | С# Массивы | 7 |
| 4 | 5 | С# Коллекции | 7 |
| 5 | 5 | С# Строки | 6 |
| 6 | 5 | С# Классы (поля и методы) | 7 |
| 7 | 5 | С# Статические и динамические поля | 6 |
| 8 | 5 | С# Статические и динамические методы | 6 |
| 9 | 5 | С# Инкапсуляция (основы) | 6 |
| 10 | 5 | С# Наследование (основы) | 6 |
| 11 | 5 | С# Полиморфизм (основы) | 6 |
| 12 | 5 | С# Интерфейсы (основы) | 7 |

| | | | |
|----|-------|--------------------------|-----|
| 13 | 5 | C# Свойства Конструкторы | 7 |
| 14 | 5 | C# Структуры | 7 |
| 15 | 5 | Тестирование (основы) | 8 |
| 16 | 1-5 | Экзамен | 36 |
| | ВСЕГО | | 136 |

5.2 Оценочные средства, используемые для текущего контроля успеваемости и промежуточной аттестации обучающихся по итогам освоения дисциплины, их виды и формы, требования к ним и шкалы оценивания приведены в приложении к рабочей программе дисциплины «Фонд оценочных средств по дисциплине Программирование», которое оформляется в виде отдельного документа.

6 Учебно-методическое и информационное обеспечение дисциплины

2 а) Основная литература

| Номер | Наименование книги | Год издания |
|-------|--|-------------|
| 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 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/65427.html | 2015 |
| 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 | 2017 |
| 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 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73713.html | 2016 |

б) Дополнительная литература

| Номер | Наименование книги | Год издания |
|-------|--|-------------|
| 1 | Ермина, М. А. Информатика. Алгоритмизация и программирование вычислительных задач : учебное пособие / М. А. Ермина, Д. А. Ермин. — Санкт-Петербург : Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2017. — 103 с. — ISBN 978-5-7937-1479-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/102422.html | 2017 |
| 2 | Тюльпинова, Н. В. Алгоритмизация и программирование [Электронный ресурс] : учебное пособие / Н. В. Тюльпинова. — Электрон. текстовые данные. — Саратов : Вузовское образование, 2019. — 200 с. — 978-5-4487-0470-3. — Режим доступа: http://www.iprbookshop.ru/80539.html | 2019 |
| 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 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73714.html | 2016 |

| | | |
|---|---|------|
| 4 | Павловская, Т. А. Программирование на языке высокого уровня С# [Электронный ресурс] / Т. А. Павловская. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 245 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73713.html | 2016 |
|---|---|------|

в) 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>

г) программное обеспечение

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

д) методические указания к лабораторным и практическим работам:

1. Программирование на языке высокого уровня [Электронный ресурс] : методические указания и варианты заданий для студентов 1-го курса направления подготовки 09.03.01 Информатика и вычислительная техника / сост. С. П. Зоткин. — Электрон. текстовые данные. — М. : Московский государственный строительный университет, Ай Пи Эр Медиа, ЭБС АСВ, 2016. — 89 с. — 978-5-7264-1277-1. — Режим доступа: <http://www.iprbookshop.ru/46060.html>
2. Береснев Н.В. Методические указания к выполнению лабораторных работ по курсу «Программирование». Воткинск. Воткинский филиал ИжГТУ имени М.Т. Калашникова, 2018
3. Программирование. Учебно-методическое пособие для выполнения курсовой работы по дисциплине «Программирование» для студентов направления: 09.03.01 Информатика и вычислительная техника (квалификация «бакалавр») / сост. К. Б. Сентяков, В. А. Смирнов. – Воткинск: ВФ ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова», 2018. – 10 с.

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).

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 используются аудитории:

№ 220 адрес: 427430, Удмуртская Республика, г. Воткинск, ул. П.И. Шувалова, д. 1, оснащенная следующим оборудованием: столы лабораторные, стулья, компьютерная техника с возможностью подключения к сети «Интернет».

№ 221 адрес: 427430, Удмуртская Республика, г. Воткинск, ул. П.И. Шувалова, д. 1, оснащенная следующим оборудованием: столы лабораторные, стулья, компьютерная техника с возможностью подключения к сети «Интернет».

4. Самостоятельная работа.

Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и доступом к электронной информационно-

образовательной среде ВФ ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»: помещения для самостоятельной работы обучающихся (ауд.№ 224, адрес: 427430, Удмуртская Республика, г. Воткинск, ул. П.И. Шувалова, д. 1).

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).

**Приложение к рабочей программе
дисциплины**

МИНОБРНАУКИ РОССИИ
Воткинский филиал
Федерального государственного бюджетного образовательного
учреждения высшего образования
«Ижевский государственный технический университет имени М.Т. Калашникова»
(ВФ ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

**Оценочные средства
по дисциплине**

Программирование

направление 09.03.01 «Информатика и вычислительная техника»

профиль «Автоматизированные системы обработки информации и управления»

уровень образования: бакалавриат

форма обучения: очная

общая трудоемкость дисциплины составляет: 11 зачетных единиц(ы)

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 | Раздел Дисциплины | Код контролируемой компетенции (или ее части) | Наименование оценочного средства |
|-------|---|--|---|
| 1 | Алгоритмизация | 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; | вопросы для проведения промежуточного контроля (к зачету) работа на практических занятиях: текущий контроль решения задач |
| 2 | Процедурный язык программирования Паскаль | ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем; | вопросы для проведения промежуточного контроля (к зачету) работа на практических занятиях: текущий контроль решения задач тест защита лабораторных работ (отчет по лабораторной работе) |
| 3 | Основы ООП | ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения; | работана практических занятиях: текущий контроль решения задач тест защита лабораторных работ (отчет по лабораторной работе) |
| | Все разделы курса | | Экзамен |

Описания элементов ФОС

1. 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) С#. Базовые структуры данных.
 - 28) С#. Класс TObject.
 - 29) С#. Жизнь классов и объектов.
 - 30) С# Поля и свойства
 - 31) С# .
- Статические методы.
- 32) С# . Виртуальные методы.
 - 33) С# . Динамические методы.
 - 34) С# . Инкапсуляция.
 - 35) С# . Наследование.
 - 36) С# . Полиморфизм.
 - 37) С#. Структуры.
 - 38) С#. Интерфейсы.
 - 39) С#. Массивы.
 - 40) С#. Коллекции.

Критерии оценки:

Приведены в разделе 2

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 | Выберите 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 считаются 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 | | 1 | 78,5 |

| | | | |
|---|--|---|------------|
| | Действительное положительное число в программе на Паскале может быть записано следующим образом: | 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 результате 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 считать корректным | 1 | var PER:char; |
| | | 2 | var PER:(red,dark, magenta); |
| | | 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; beginFOR 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 | Выберите 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:=st2st1; 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 | клавишт |
| | | 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'; | 1 | комерция |
| | | 2 | коммерция |
| | | 4 | м |
| | st:=concat(st3,st4); writeln(st); end. | 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= recordd: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; | 1 | t.d[true]:=34; |
| | | 2 | t[true].e:=45; |
| | | 4 | t.f[2].a[false]:=67; |
| | begin t.d[true]:=34; t[true].e:=45; t.f[2].a[false]:=67; t.f[1].b:=['a']; end. возникнет ошибка? | 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. | 1 | function PRIMER(s,d:real):integer; |
| | | 2 | a:=s/d+a; |
| | | 4 | PRIMER:=a; |

| | | | |
|---|--|---|--|
| | вызовет ошибку? | 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 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? | 1 | function PRIMER(s,d:integer):real; |
| | | 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 | 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 |

| | | | |
|---|---|---|---|
| | 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 | 8 | тип dd должен быть описан в библиотеке |
| 3 | Программа: program Primer1 1; 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 Primer1 1; 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 должна 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

```
1 static void Main()
2 {
3     var arr = new int[4];
4     for (var i = 0; i <= arr.Length; i++)
5         arr[i] = 0;
6     DoSomething(arr);
7 }
```

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

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

```
1 string result="";
2 for (int i=0; i<3; i++)
3 {
4     result+=i.ToString() + ", ";
5 }
```

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.

| | |
|-----------------|--|
| <code>\n</code> | |
| <code>\t</code> | |
| <code>\\</code> | |
| <code>\r</code> | |
| <code>\"</code> | |

Слэш

Перевод строки

Возврат каретки

Табуляция

Кавычка

Тест №7

Изучите код ниже:

```
1 class Sample1 {
2     static void F() {
3         i = 1;
4     }
5     static int i = 0;
6 }
```

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:

```
1 class Sample2 {
2     static string who = "class";
3
4     static void F() {
5         string who = "F";
6     }
7
8     static void G() {
9         F();
10        Console.WriteLine(who);
11    }
12
13    static void H() {
14        string who = "H";
15        F();
16        Console.Write(who);
17    }
18 }
```

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

- class
- F
- H
- Ничего. Будет ошибка

Изучите следующий код:

```
1 class ClassA {
2     public string FieldA;
3 }
4 class ClassB : ClassA {
5     public string FieldB;
6 }
7 class ClassC : ClassA {
8     public string FieldC;
9 }
10 class ClassD : ClassC {
11     public string FieldD;
12 }
```

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:

```
1 class Sample2 {
2     static string who = "class";
3
4     static void F() {
5         string who = "F";
6     }
7
8     static void G() {
9         F();
10        Console.WriteLine(who);
11    }
12
13    static void H() {
14        string who = "H";
15        F();
16        Console.Write(who);
17    }
18 }
```

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

- class
- F
- H
- Ничего. Будет ошибка

```

1 class Sample3 {
2     static string who = "class";
3
4     static void Mixed() {
5         Console.Write(who + " ");
6         string who = "Mixed";
7         Console.Write(who);
8     }
9 }

```

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.

```

1 class Sample4 {
2     void M1() {
3         int i = 0;
4         {
5             int i = 1;
6         }
7     }
8     void M2(int i) {
9         int i = 0;
10    }
11    void M3() {
12        // Фигурными скобками можно группировать операторы (правда почти никогда этого делать не стоит)
13        // и каждая пара фигурных скобок создает свою область видимости.
14        {
15            int i = 0;
16        }
17        {
18            int i = 1;
19        }
20    }
21 }

```

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:

```
1 class SomeClass {
2     public static int s = 1;
3     public int d = 1;
4
5     public void Run() {
6         Console.Write(s + " " + d + " ");
7         s++; d++;
8     }
9
10    public static void Main() {
11        var object1 = new SomeClass();
12        var object2 = new SomeClass();
13        object1.Run();
14        object2.Run();
15        object1.Run();
16    }
17 }
```

3. Что напечатает вызов метода Main в листинге выше?

4. Метод Run динамический?

- Верно
 Неверно

5. Метод Main статический?

- Верно
 Неверно

Тест №14

```
1 class A
2 {
3     private int a;
4     public int B;
5     public void PublicMethod(){
6         a = 1; // строка 1
7         B = 2; // строка 2
8     }
9     private void PrivateMethod(){
10        a = 3; // строка 3
11        B = 4; // строка 4
12    }
13 }
14
15 class B
16 {
17     private A privateA = new A();
18     public A PublicA = new A();
19
20     public void M(){
21         privateA.a = 5; // строка 5
22         PublicA.a = 7; // строка 6
23         privateA.B = 6; // строка 7
24         PublicA.B = 8; // строка 8
25         PublicA.PublicMethod(); // строка 9
26         privateA.PublicMethod(); // строка 10
27         PublicA.PrivateMethod(); // строка 11
28         privateA.PrivateMethod(); // строка 12
29     }
30 }
```

Тест №15

Изучите следующий код:

```
1 X[] xs = new X[10];
2 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:

```
1 X[] xs = new X[10];
2 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 может быть структурой
- 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 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

3 Наименование: защита лабораторных работ

Представление в ФОС: задания и требования к выполнению представлены в методических указаниях по дисциплине

Варианты заданий: задания и требования к выполнению представлены в методических указаниях по дисциплине **Критерии оценки:**

Приведены в разделе 2.

4 Наименование: работа на практических занятиях: текущий контроль решения 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 разделяются пробелами (может быть больше одного)

3. Вводится текст конец ввода F6. Определить количество слов, состоящих из 4 букв. 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. Вводятся 15 двузначных чисел. Напечатать каждое число введенное впервые

4. Вводится 2 множества. 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 среднего балла. 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 средним 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

6. Наименование: дифференцированный зачет 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

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.

| Разделы дисциплины | Форма контроля | Количество баллов | |
|---------------------------|-------------------------|--------------------------|------------|
| | | <i>min</i> | <i>max</i> |
| 6 | Лабораторная работа № 1 | 10 | 20 |
| 6 | Лабораторная работа № 2 | 20 | 40 |
| 6 | Лабораторная работа № 3 | 20 | 40 |
| | 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. |

Промежуточная аттестация по дисциплине проводится в форме экзамена.

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

| Оценка | Набрано баллов |
|-----------------------|-----------------------|
| «отлично» | 90-100 |
| «хорошо» | 80-89 |
| «удовлетворительно» | 55-79 |
| «неудовлетворительно» | 0-54 |

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.

Промежуточная аттестация проводится в письменной форме. 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.

Билет к экзамену включает 2 теоретических вопроса.

Время на подготовку: 40 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» | <p>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</p> |
| «неудовлетворительно» | <p>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</p> |