

**МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«АРКТИЧЕСКИЙ ГОСУДАРСТВЕННЫЙ АГРОТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»**  
(ФГБОУ ВО Арктический ГАТУ)  
Колледж технологий и управления

Регистрационный № 24-1/25

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Дисциплина **ОП.04 Основы алгоритмизации и программирования**

Специальность **09.02.07. Информационные системы и программирование**

Квалификация **Программист**

Уровень ППСЗ **базовая**

Срок освоения ППСЗ **3 г 10 мес**

Форма обучения **очная**

Общая трудоемкость **193 ч**

Якутск 2024

Рабочая программа учебной дисциплины разработана в соответствии с:  
- Федеральным государственным образовательным стандартом среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование, утвержденный приказом Министерства образования и науки Российской Федерации от 09 декабря 2016 г. №1547.  
- Учебным планом специальности 09.02.07 Информационные системы и программирование, одобрен Ученым советом ФГБОУ ВО Арктический ГАТУ № 24 от 30.05.2024 г.

Разработчик(и) РПД Федоров Павел Иванович – преподаватель

Председатель ЦК ГиЕД \_\_\_\_\_  /Васильева Е.К./  
подпись фамилия, имя, отчество

Протокол заседания ЦК ГиЕД № 10 от « 24 » мая 2024 г.

Директор КТиУ \_\_\_\_\_  /Яковлева Н.М./  
подпись фамилия, имя, отчество

« 24 » мая 2024 г

## СОДЕРЖАНИЕ

<b>№</b>	<b>Наименование раздела</b>	<b>Стр.</b>
1	Общая характеристика рабочей программы учебной дисциплины	4
2	Структура и содержание учебной дисциплины	5
3	Условия реализации учебной дисциплины	11
4	Контроль и оценка результатов освоения учебной дисциплины	14

# 1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

## ОП.04 Основы алгоритмизации и программирования

### 1.1. Область применения программы

Рабочая программа учебной дисциплины является частью программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности (специальностям) СПО 09.02.07 Информационные системы и программирование.

### 1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена:

Учебная дисциплина ОП.04 «Основы алгоритмизации и программирования» относится к общепрофессиональному циклу.

#### **Освоение дисциплины способствует формированию компетенций:**

**ОК 01.** Выбирать способы решения задач в профессиональной деятельности, применительно к различным контекстам;

**ОК 02.** Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

**ОК 04.** Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами;

**ОК 05.** Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста;

**ОК 09.** Использовать информационные технологии профессиональной деятельности;

**ОК 10.** Пользоваться профессиональной документацией на государственном и иностранном языке;

**ПК 1.1.** Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием;

**ПК 1.2.** Разрабатывать программные модули в соответствии с техническим заданием;

**ПК 1.3.** Выполнять отладку программных модулей с использованием специализированных программных средств;

**ПК 1.4.** Выполнять тестирование программных модулей;

**ПК 1.5.** Осуществлять рефакторинг и оптимизацию программного входа;

**ПК 2.4.** Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения;

**ПК 2.5.** Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

### 1.3. Цели и задачи учебной дисциплины – требования к результатам освоения учебной дисциплины:

**Цель дисциплины** - обеспечение обучающихся теоретическими знаниями и умениями, практическими навыками, необходимыми для эффективного выполнения профессиональной деятельности.

#### **Задача дисциплины:**

Сформировать представление о программировании.

Интеллектуальное развитие.

Овладение необходимыми конкретными знаниями и умениями.

Воспитательное воздействие.

**В результате освоения учебной дисциплины обучающийся должен уметь:**

У.1. Разрабатывать алгоритмы для конкретных задач.

У.2. Использовать программы для графического отображения алгоритмов.

У.3. Определять сложность работы алгоритмов.

У.4. Работать в среде программирования.

У.5. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

У.6. Оформлять код программы в соответствии со стандартом кодирования.

У.7. Выполнять проверку, отладку кода программы.

**В результате освоения учебной дисциплины обучающийся должен знать:**

3.1. Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.

3.2. Эволюцию языков программирования, их классификацию, понятие системы программирования.

3.3. Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

3.4. Подпрограммы, составление библиотек подпрограмм

3.5. Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения

#### **1.4. Рекомендуемое количество часов на освоение программы учебной дисциплины:**

Максимальной учебной нагрузки обучающегося 193 часов, в том числе:

- обязательной аудиторной учебной нагрузки обучающегося 168 часов;

- самостоятельная работа студентов – 16 часов;

- консультация – 1 час;

- промежуточная аттестация (в форме экзамена) – 8 часов.

## **2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

### **2.1. Объем учебной дисциплины и виды учебной работы**

<b>Вид учебной работы</b>	<b>Всего часов</b>	<b>В тч в 3 семестре</b>	<b>В тч в 4 семестре</b>	<b>В тч в 5 семестре</b>
Максимальная учебная нагрузка (всего)	193	32	76	85
Обязательная аудиторная учебная нагрузка (всего)	168	32	76	60
в том числе:				
лекции	74	16	38	20
лабораторные занятия	36	16		20
практические занятия	58		38	20
консультация	1			1
Самостоятельная работа	16			16
<b>Промежуточная аттестация - в форме экзамена в пятом семестре</b>	8			8

## 2.2. Тематический план и содержание учебной дисциплины Основы алгоритмизации и программирования

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающегося, курсовая работа (проект)	Объем в часах	Уровень освоения
1	2	3	4
<b>Раздел 1.</b>	<b><i>Введение в программирование</i></b>	<b>28</b>	1,2,3
<b>Тема 1.1.</b> <b>Языки программирования</b>	<b>Содержание учебного материала</b>	<b>24</b>	
	1. Развитие языков программирования.		
	2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.		
	3. Жизненный цикл программы. Программа. Программный продукт и его характеристики.		
	4. Основные этапы решения задач на компьютере. <b>В том числе практических занятий и лабораторных работ</b>		
<b>Тема 1.2.</b> <b>Типы данных</b>	<b>Содержание учебного материала</b>	<b>4</b>	
	1. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных. <b>В том числе практических занятий и лабораторных работ</b>	<b>2</b>	
<b>Раздел 2.</b>	<b>Содержание учебного материала</b>	<b>40</b>	1,2,3
<b>Тема 2.1.</b> <b>Операторы языка программирования</b>	1. Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.		
	2. Условный оператор. Оператор выбора.		
	3. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.		
	4. Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.		
	5. Структурированный тип данных - множество. Операции над множествами.		
	6. Комбинированный тип данных - запись. Файлы последовательного доступа. Файлы прямого доступа		

	<b>В том числе практических занятий и лабораторных работ</b>	<b>18</b>	
<b>Раздел 3.</b>	<b>Содержание учебного материала</b>	<b>28</b>	
<b>Тема 3.1. Процедуры и функции</b>	1. Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.	<b>12</b>	1,2,3
	2. Рекурсия. Программирование рекурсивных алгоритмов.		
	<b>В том числе практических занятий и лабораторных работ</b>	<b>6</b>	
<b>Тема 3.2. Структуризация в программировании</b>	<b>Содержание учебного материала</b>	<b>4</b>	1,2,3
	1. Основы структурного программирования. Методы структурного программирования.	<b>2</b>	
	<b>В том числе практических занятий и лабораторных работ</b>		
<b>Тема 3.3. Модульное программирование</b>	<b>Содержание учебного материала</b>	<b>12</b>	1,2,3
	1. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.	<b>6</b>	
	2. Стандартные модули.		
	<b>В том числе практических занятий и лабораторных работ</b>		
<b>Раздел 4</b>	<b><i>Основные конструкции языков программирования</i></b>	<b>12</b>	1,2,3
<b>Тема 4.1 Указатели</b>	<b>Содержание учебного материала</b>	<b>12</b>	1,2,3
	1. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.	<b>6</b>	
	2. Структуры данных на основе указателей.		
	3. Задача о стеке.		
	<b>В том числе практических занятий и лабораторных работ</b>		
<b>Раздел 5</b>	<b>Содержание учебного материала</b>	<b>60</b>	1,2,3
<b>Тема 5.1 Основные принципы объект-ноориентированного программирования (ООП)</b>	1. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.	<b>12</b>	
	2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.		
	3. Классы объектов. Компоненты и их свойства.		
	4. Событийно-управляемая модель программирования. Компонентно ориентированный подход.	<b>8</b>	
	<b>В том числе практических занятий и лабораторных работ</b>		
	Самостоятельная работа обучающихся		
<b>Тема 5.2</b>	<b>Содержание учебного материала</b>	<b>12</b>	

<b>Интегрированная среда разработчика</b>	<b>1. Требования к аппаратным и программным средствам интегрированной среды разработчика.</b>	
	<b>2. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.</b>	
	<b>3. Панель компонентов и их свойства. Окно кода проекта.</b>	
	<b>4. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.</b>	
	<b>5. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.</b>	
	<b>6. Настройка среды и параметров проекта.</b>	
	<b>В том числе практических занятий и лабораторных работ</b>	<b>8</b>
	<b>Самостоятельная работа обучающихся</b>	<b>4</b>
<b>Тема 5.3.</b>	<b>Содержание учебного материала</b>	<b>10</b>
<b>Визуальное событийно-управляемое программирование</b>	1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.	
	2. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.	
	3. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.	
	<b>В том числе практических занятий и лабораторных работ</b>	<b>8</b>
	<b>Самостоятельная работа обучающихся</b>	<b>4</b>
	<b>Содержание учебного материала</b>	<b>12</b>
<b>Тема 5.4</b>	1. Разработка функционального интерфейса приложения. Создание интерфейса приложения.	
	2. Разработка функциональной схемы работы приложения.	
	3. Разработка игрового приложения.	
	<b>В том числе практических занятий и лабораторных работ</b>	<b>8</b>
<b>Разработка оконного приложения</b>	<b>Самостоятельная работа обучающихся</b>	<b>4</b>
<b>Тема 5.5</b>	<b>Содержание учебного материала</b>	<b>10</b>
<b>Этапы разработки приложений</b>	1. Разработка приложения.	
	2. Проектирование объектно-ориентированного приложения.	
	3. Создание интерфейса пользователя.	
	4. Тестирование, отладка приложения.	



	<b>В том числе практических занятий и лабораторных работ</b>	<b>6</b>	
<b>Тема 5.6</b> <b>Иерархия классов.</b>	<b>Содержание учебного материала</b>	<b>4</b>	
	1. Классы ООП: виды, назначение, свойства, методы, события.		
	2. Перегрузка методов.		
	3. Тестирование и отладка приложения.		
	<b>4. Решение задач</b>	<b>4</b>	
<b>В том числе практических занятий и лабораторных работ</b>			
<i>Примерная тематика практических занятий и лабораторных работ:</i>			
<p>Знакомство со средой программирования.</p> <p>Составление программ линейной структуры.</p> <p>Составление программ разветвляющейся структуры.</p> <p>Составление программ циклической структуры</p> <p>Обработка одномерных массивов.</p> <p>Обработка двумерных массивов.</p> <p>Работа со строками.</p> <p>Работа с данными типа множество.</p> <p>Файлы последовательного доступа.</p> <p>Типизированные файлы.</p> <p>Нетипизированные файлы.</p> <p>Организация процедур.</p> <p>Организация функций.</p> <p>Применение рекурсивных функций.</p> <p>Программирование модуля.</p> <p>Создание библиотеки подпрограмм.</p> <p>Использование указателей для организации связанных списков.</p> <p>Изучение интегрированной среды разработчика.</p> <p>Создание проекта с использованием компонентов для работы с текстом.</p> <p>Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени. События компонентов (элементов управления), их сущность и назначение.</p> <p>Создание процедур на основе событий.</p> <p>Создание проекта с использованием кнопочных компонентов.</p> <p>Создание проекта с использованием компонентов стандартных диалогов и системы меню. Разработка функциональной схемы работы приложения.</p> <p>Разработка оконного приложения с несколькими формами.</p> <p>Разработка игрового приложения.</p> <p>Создание процедур обработки событий. Компиляция и запуск приложения.</p>			

Разработка интерфейса приложения. Тестирование, отладка приложения. Классы ООП: виды, назначение, свойства, методы, события. Объявления класса. Создание наследованного класса. Программирование приложений. Перегрузка методов.		
<b>Всего аудиторных часов</b>	<b>168</b>	
<b>Самостоятельная работа</b>	<b>16</b>	
<b>Консультация</b>	<b>1</b>	
<b>Промежуточная аттестация – форме экзамена</b>	<b>8</b>	
<b>Всего</b>	<b>193</b>	

### 3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

#### 3.1. Требования к минимальному материально-техническому обеспечения

№ п/п	Наименование дисциплины (модуля), практик в соответствии с учебным планом	Наименование специальных* помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы
1	ОП.04 Основы алгоритмизации и программирования	<b>Ауд. № 2.405</b> <b>Компьютерный класс. Кабинет информатики,</b> учебная аудитория для занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, выполнения курсовых работ, текущего контроля и промежуточной аттестации, самостоятельной работы <b>Кабинет № 6 – 86,1 м<sup>2</sup></b> 677007, Республика Саха (Якутия), г. Якутск, ш. Сергеляхское, 3 км, д.3	<b>Оборудование:</b> Системный блок (Rusco Core-i3-7100/2*4Gb/500Gb/Win10Pro/Office - 16 шт.; Монитор (22” Benq GL2250) - 16 шт.; Интерактивная доска SMART Board 680; Проектор LGRL-JT40). <b>Учебная мебель:</b> Компьютерный стол – 32, стул подъемно-поворотный – 16, стулья – 17, стол письменный – 1. <b>Программное обеспечение:</b> Win10Pro; Microsoft Office16; Kaspersky Endpoint Security for Business; Adobe reader; PascalABC.NET (открытое лицензионное соглашение); Free Pascal (открытое лицензионное соглашение).
		<b>Ауд. № 2.406</b> <b>Компьютерный класс. Лаборатория информационных ресурсов</b> <b>Кабинет № 7 – 78,8 м<sup>2</sup></b> Учебная аудитория для занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, выполнения курсовых работ,	<b>Оборудование:</b> Автоматизированные рабочие места обучающихся: Системный блок (Rusco Core-i3-7100/2*4Gb/500Gb/Win10Pro/Office - 15 шт.; Монитор (22” Benq GL2250) - 15 шт.; Автоматизированное рабочее место преподавателя: Системный блок (Rusco Core-i3-7100/2*4Gb/500Gb/Win10Pro/Office - 1 шт.; Монитор (22” Benq GL2250) - 1 шт.; Интерактивная доска SMART Board 680; Проектор LGRL-JT40; МФУ HP LaserJet Pro MFP 127fn – 1 шт.; навесной экран, маркерная доска. <b>Учебная мебель:</b>

		<p>текущего контроля и промежуточной аттестации, самостоятельной работы</p> <p>677007, Республика Саха (Якутия), г. Якутск, ш. Сергеляхское, 3 км, д.3</p>	<p>Специализированная мебель для сервисного обслуживания ПК с заземлением и защитой от статического напряжения, компьютерный стол – 16, стул подъемно-поворотный – 16, стулья – 25.</p> <p><b>Программное обеспечение:</b>  Win10Pro;  Microsoft Office16;  Kaspersky Endpoint Security for Business;  NetBeans IDE (открытое лицензионное соглашение);  Microsoft Visual Studio (открытое лицензионное соглашение);  MySQL (открытое лицензионное соглашение);  Adobe reader. PascalABC.NET (открытое лицензионное соглашение);  Free Pascal (открытое лицензионное соглашение).</p>
--	--	--	---

### 3.2. Информационное обеспечение обучения

#### Перечень учебных изданий, интернет-ресурсов, дополнительной литературы

##### Основные источники:

№	Наименование	Авторы	Год и место издания	Используется при изучении тем	Семестр
1	2	3	4	5	6
1	<p>Основы алгоритмизации и программирования: учебник для СПО / . —. 137 с</p> <p>Режим доступа: <a href="https://www.biblio-online.ru/viewer/osnovy-algoritmizacii-i-programmirovaniya-454452#page/2">https://www.biblio-online.ru/viewer/osnovy-algoritmizacii-i-programmirovaniya-454452#page/2</a></p>	<p>В.В. Трофимов, Т. А. Павловская; под ред. В. В. Трофимова</p>	<p>М: Издательство Юрайт, 2020</p>	1-5	3-5

##### Дополнительные источники:

№	Наименование	Авторы	Год и место издания	Используется при изучении тем	Семестр
1	2	3	4	5	6
1	<p>Основы программирования на языке C#: учеб. пособие для СПО / —. 105 с</p> <p>Режим доступа: <a href="https://www.biblio-online.ru/viewer/osnovy-algoritmizacii-i-programmirovaniya-na-yazyke-c-456221#page/1">https://www.biblio-online.ru/viewer/osnovy-algoritmizacii-i-programmirovaniya-na-yazyke-c-456221#page/1</a></p>	<p>Кудрина Е.В., Огнева М.В.</p>	<p>М: Издательств о Юрайт, 2020</p>	1-5	3-5

### **Перечень электронных ресурсов:**

<b>№</b>	<b>Наименование</b>
Э1	<a href="http://www.fcior.edu.ru">www.fcior.edu.ru</a> (Федеральный центр информационно-образовательных ресурсов — ФЦИОР).
Э2	<a href="http://www.school-collection.edu.ru">www.school-collection.edu.ru</a> (Единая коллекция цифровых образовательных ресурсов).
Э3	<a href="http://www.intuit.ru/studies/courses">www.intuit.ru/studies/courses</a> (Открытые интернет-курсы «Интуит» по курсу «Информатика»)
Э4	<a href="http://www.lms.iite.unesco.org">www.lms.iite.unesco.org</a> (Открытые электронные курсы «ИИТО ЮНЕСКО» по информационным технологиям).
Э5	<a href="http://ru.iite.unesco.org/publications">http://ru.iite.unesco.org/publications</a> (Открытая электронная библиотека «ИИТО ЮНЕСКО» по ИКТ в образовании).
Э6	<a href="http://www.megabook.ru">www.megabook.ru</a> (Мегаэнциклопедия Кирилла и Мефодия, разделы «Наука / Математика. Кибернетика» и «Техника / Компьютеры и Интернет»).
Э7	<a href="http://www.ict.edu.ru">www.ict.edu.ru</a> (портал «Информационно-коммуникационные технологии в образовании»).
Э8	<a href="http://www.digital-edu.ru">www.digital-edu.ru</a> (Справочник образовательных ресурсов «Портал цифрового образования»)
Э9	<a href="http://www.window.edu.ru">www.window.edu.ru</a> (Единое окно доступа к образовательным ресурсам Российской Федерации)
Э10	<a href="http://www.freeschool.altlinux.ru">www.freeschool.altlinux.ru</a> (портал Свободного программного обеспечения)
Э11	<a href="http://www.hear.altlinux.org/issues/textbooks">www.hear.altlinux.org/issues/textbooks</a> (учебники и пособия по Linux)
Э12	<a href="http://www.books.altlinux.ru/altlibrary/openoffice">www.books.altlinux.ru/altlibrary/openoffice</a> (электронная книга «OpenOffice.org: Теория и практика»)
Э1	Учебники по программированию <a href="http://programm.ws/index.php">http://programm.ws/index.php</a>

### **Перечень информационных справочных систем:**

<b>№</b>	<b>Наименование</b>
1	Информационно-правовая система Гарант

## **3.3. Условия реализации учебной дисциплины для студентов-инвалидов и лиц с ограниченными возможностями здоровья**

### **3.3.1. Образовательные технологии.**

С целью оказания помощи в обучении студентов-инвалидов и лиц с ОВЗ применяются образовательные технологии с использованием универсальных, специальных информационных и коммуникационных средств.

Для основных видов учебной работы применяются:

Контактная работа:

- лекции – проблемная лекция, лекция-дискуссия, лекция-диалог, лекция-консультация, лекция с применением дистанционных технологий и привлечением возможностей Интернета;
- практические (семинарские) занятия - практические задания;
- групповые консультации – опрос, работа с лекционным и дополнительным материалом;
- индивидуальная работа с преподавателем - индивидуальная консультация, работа с лекционным и дополнительным материалом, беседа, морально-эмоциональная поддержка и стимулирование, дистанционные технологии.

Формы самостоятельной работы устанавливаются с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге или на компьютере).

В качестве самостоятельной подготовки в обучении используется - система дистанционного обучения Moodle, <http://sdo.agatu.ru/>.

Самостоятельная работа:

- работа с книгой и другими источниками информации, план-конспекты;
- творческие самостоятельные работы;

- дистанционные технологии.

При необходимости обучающимся предоставляется дополнительное время для консультаций и выполнения заданий.

### 3.3.2. Специальное материально-техническое и учебно-методическое обеспечение.

При обучении по дисциплине используется система, поддерживающая дистанционное образование - Moodle, <http://sdo.agatu.ru/>, ориентированная на организацию дистанционных курсов, а также на организацию взаимодействия между преподавателем и обучающимися посредством интерактивных обучающих элементов курса.

*Для обучающихся лиц с нарушением зрения предоставляются:*

- видеоувеличитель-монокуляр для просмотра Levenhuk Wise 8x25;
- электронный ручной видеоувеличитель видео оптик “wu-tv”;
- возможно также использование собственных увеличивающих устройств;
- версия сайта академии <http://www.agatu.ru/> для слабовидящих.

*Для обучающихся лиц с нарушением слуха предоставляются:*

- аудитории со звукоусиливающей аппаратурой (колонки, микрофон);
- компьютерная техника в оборудованных классах;
- учебные аудитории с мультимедийной системой с проектором;
- аудитории с интерактивными досками в аудиториях;
- учебные пособия, методические указания в форме электронного документа

*Для обучающихся лиц с нарушениями опорно-двигательного аппарата предоставляются:*

- система дистанционного обучения Moodle, <http://sdo.agatu.ru/>;
- учебные пособия, методические указания в форме электронного документа

### 3.3.3. Контроль и оценка результатов освоения учебной дисциплины.

Контроль результатов обучения осуществляется в процессе проведения практических занятий, выполнения индивидуальных самостоятельных работ.

Формы и сроки проведения рубежного контроля определяются с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.), и может проводиться в несколько этапов.

При необходимости, предоставляется дополнительное время для подготовки ответов на зачете, аттестация проводится в несколько этапов (по частям), во время аттестации может присутствовать ассистент, аттестация прерывается для приема пищи, лекарств, во время аттестации используются специальные технические средства.

## 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения учебной дисциплины осуществляется преподавателем в процессе проведения практических занятий и лабораторных работ, тестирования, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
<b>Итоговый контроль:</b>	<b>Экзамен</b>
<b>Уметь</b>	
У.1 Разрабатывать алгоритмы для конкретных задач	Компьютерное тестирование на знание терминологии по теме;
У.2 Использовать программы для графического отображения алгоритмов	•Тестирование.
У.3 Определять сложность работы алгоритмов	•Контрольная работа •Самостоятельная работа.

У.4 Работать в среде программирования	<ul style="list-style-type: none"> <li>• Защита реферата.</li> <li>• Семинар</li> <li>• Наблюдение за выполнением практического задания и лабораторных работ. (деятельностью студента)</li> <li>• Оценка выполнения практического задания(работы)</li> <li>• Подготовка и выступление с докладом, сообщением, презентацией.</li> <li>• Решение ситуационной задачи.</li> </ul>
У.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования	
У.6 Оформлять код программы в соответствии со стандартом кодирования	
У.7 Выполнять проверку, отладку кода программы	
<b>Знать</b>	
3.1 Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	
3.2 Эволюцию языков программирования, их классификацию, понятие системы программирования	
3.3 Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти	
3.4 Подпрограммы, составление библиотек подпрограмм	
3.5 Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения	

Лист изменений и дополнений общих компетенций  
по специальности  
09.02.07 Информационные системы и программирование

Актуализированы новые общие компетенции приказ Минпросвещения России от 03.07.2024 №464 по специальности 09.02.07 Информационные системы и программирование:

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях; (в ред. Приказа Минпросвещения России от 03.07.2024 N 464);

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения; (в ред. Приказа Минпросвещения России от 03.07.2024 N 464);

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках. (п. 3.2 в ред. Приказа Минпросвещения России от 01.09.2022 N 796).

Председатель МК КТиУ



*Ваганова В.Г.*

Ваганова В.Г.

Протокол заседания МК КТиУ от «16» сентября 2024 г. № 1.



МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Арктический государственный агротехнологический университет»  
Колледж технологий и управления

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
по учебной дисциплине

**ОП.04 Основы алгоритмизации и программирования**

09.02.07. Информационные системы и программирование

Якутск 2024 г.

Фонд оценочных средств учебной дисциплины разработан в соответствии с:

- Федеральным государственным образовательным стандартом среднего профессионального образования по специальности 09.02.07. Информационные системы и программирование, утвержденный приказом Министерства образования и науки Российской Федерации от 09 декабря 2016 г. №1547.

- Учебным планом специальности 09.02.07. Информационные системы и программирование одобрен Ученым советом ФГБОУ ВО Арктический ГАТУ Протокол №24 от 30.05.2024г.

Разработчик(и) ФОС Федоров Павел Иванович– преподаватель

Фонд оценочных средств учебной дисциплины ОП 04. Основы алгоритмизации и программирования одобрен на цикловой комиссии гуманитарных и естественных дисциплин от «24» мая 2024 г. Протокол №10

Председатель ЦК ГиЕД \_\_\_\_\_

подпись

/Васильева Е.К./  
фамилия, имя, отчество

Фонд оценочных средств учебной дисциплины рассмотрен и рекомендован к использованию в учебном процессе на заседании методической комиссии Колледжа технологий и управления по специальности 09.02.07. Информационные системы и программирование.

Председатель методической комиссии КТиУ \_\_\_\_\_

подпись

/Сивцева Е.И./  
фамилия, имя, отчество

## ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

ОП.04 Основы алгоритмизации и программирования  
09.02.07 Информационные системы и программирование

Таблица 1

Результаты обучения (освоенные умения, усвоенные знания) <sup>1</sup>	Формируемые компетенции <sup>1</sup>	Наименование темы <sup>2</sup>	Уровень освоения Темы <sup>2</sup>	Наименование контрольно-оценочного средства	
				Текущий контроль <sup>3</sup>	Промежуточная аттестация <sup>4</sup>
1	2	3	4	5	6
<p><b>Уметь:</b></p> <p>У.1. Разрабатывать алгоритмы для конкретных задач.</p> <p>У.2. Использовать программы для графического отображения алгоритмов.</p> <p>У.3. Определять сложность работы алгоритмов.</p> <p>У.4. Работать в среде программирования.</p> <p>У.5. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.</p> <p>У.6. Оформлять код программы в соответствии со стандартом кодирования.</p> <p>У.7. Выполнять проверку, отладку кода программы.</p> <p><b>Знать:</b></p> <p>3.1. Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные</p>	<p><b>ОК 01.</b> Выбирать способы решения задач в профессиональной деятельности, применительно к различным контекстам;</p> <p><b>ОК 02.</b> Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;</p> <p><b>ОК 04.</b> Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами;</p> <p><b>ОК 05.</b> Осуществлять устную и письменную коммуникацию</p>	<p>Тема 1.1. Языки программирования</p> <p>Тема 1.2. Типы данных</p> <p>Тема 2.1. Операторы языка программирования</p> <p>Тема 3.1. Процедуры и функции</p> <p>Тема 3.2. Структуризация в программировании</p> <p>Тема 3.3. Модульное программирование</p> <p>Тема 4.1. Указатели</p> <p>Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)</p> <p>Тема 5.2 Интегрированная среда разработчика</p> <p>Тема 5.3. Визуальное событийно-управляемое</p>	1,2,3	Вопросы к устному опросу, контрольные вопросы для защиты лабораторных и практических работ	Экзамен

<p>алгоритмические конструкции.</p> <p>3.2 Эволюцию языков программирования, их классификацию, понятие системы программирования.</p> <p>3.3 Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</p> <p>3.4 Подпрограммы, составление библиотек подпрограмм</p> <p>3.5 Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.</p>	<p>на государственном языке с учетом особенностей социального и культурного контекста;</p> <p><b>ОК 09.</b> Использовать информационные технологии профессиональной деятельности;</p> <p><b>ОК 10.</b> Пользоваться профессиональной на государственном и иностранном языке;</p> <p><b>ПК 1.1</b> Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием</p> <p><b>ПК 1.2.</b> Разрабатывать программные модули в соответствии с техническим заданием;</p> <p><b>ПК 1.3.</b> Выполнять отладку программных модулей с использованием специализированных программных средств;</p> <p><b>ПК 1.4.</b> Выполнять</p>	<p>программирование</p> <p>Тема 5.4 Разработка оконного приложения</p> <p>Тема 5.5 Этапы разработки приложений</p> <p>Тема 5.6 Иерархия классов.</p>			
---	---	--	--	--	--

	<p>тестирование программных модулей;  <b>ПК 1.5.</b>  Осуществлять рефакторинг и оптимизацию программного входа;  <b>ПК 2.4.</b>  Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения;  <b>ПК 2.5.</b>  Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>				
--	---	--	--	--	--

## 2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по учебной дисциплине осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих компетенций.

Таблица 2

Компетенции	Результаты обучения	Основные показатели оценки результата	Формы и методы контроля и оценки
	<b>Знает:</b>		
<b>ОК 01.</b> Выбирать способы решения задач в профессиональной деятельности, применительно к различным контекстам; <b>ОК 02.</b> Осуществлять поиск, анализ и интерпретацию	3.1 Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.	Правильные выбор и использование специализированных ПС для обработки различных видов информации грамотное применение аппаратных и программных мультимедийных средств для ввода,	Фронтальный опрос Тестирование Текущий контроль Оценка выполнения практических работ Текущий контроль Промежуточная аттестация в форме Экзамен
	3.2 Эволюцию языков программирования, их классификацию, понятие системы		

<p>информации, необходимой для выполнения задач профессиональной деятельности;  <b>ОК 04.</b> Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами;  <b>ОК 05.</b> Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста;  <b>ОК 09.</b> Использовать информационные технологии профессиональной деятельности;  <b>ОК 10.</b> Пользоваться профессиональной на государственном и иностранном языке;  <b>ПК 1.1.</b> Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием  <b>ПК 1.2.</b> Разрабатывать программные модули в соответствии с техническим</p>	<p>программирования.</p>	<p>хранения, обработки и представления информации  корректный выбор ПО и правильное его применение для обработки информации в соответствии с поставленной задачей</p>
	<p>3.3 Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</p>	
	<p>3.4 Подпрограммы, составление библиотек подпрограмм</p>	
	<p>3.5 Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения</p>	
	<p><b>Умеет:</b></p>	
	<p>У.1. Разрабатывать алгоритмы для конкретных задач.</p>	
	<p>У.2 Использовать программы для графического отображения алгоритмов.</p>	
	<p>У.3 Определять сложность работы алгоритмов.  У.4 Работать в среде программирования.</p>	

<p>заданием;  <b>ПК 1.3.</b> Выполнять отладку программных модулей с использованием специализированных программных средств;  <b>ПК 1.4.</b> Выполнять тестирование программных модулей;  <b>ПК 1.5.</b> Осуществлять рефакторинг и оптимизацию программного входа;  <b>ПК 2.4.</b> Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения;  <b>ПК 2.5.</b> Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>	<p>У.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.</p>		
	<p>У.6 Оформлять код программы в соответствии со стандартом кодирования.</p>		
	<p>У.7 Выполнять проверку, отладку кода программы.</p>		

## 2.1. Оценка освоения учебной дисциплины

### 2.1.1. Формы и методы оценивания

Предметом оценки служат умения и знания, предусмотренные ФГОС по дисциплине ОП.04 Основы алгоритмизации и программирования направленные на формирование общих компетенций.

Таблица 3

#### Перечень объектов контроля и оценки

Результаты обучения	Основные показатели оценки результата	Оценка (да/нет)
<b>Знает:</b>		
3.1 Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные	Правильная классификация информационных технологий и соответствие выбора методам обработки.	да

алгоритмические конструкции.		
3.2 Эволюцию языков программирования, их классификацию, понятие системы программирования.	Правильный выбор способов и средств сбора, накопления, обработки, передачи и распространения информации.	да
3.3 Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.	Корректное применение разных видов информационных технологий для получения оптимального результата	да
3.4 Подпрограммы, составление библиотек подпрограмм	Результативность применения инструментальных средств информационных технологий	да
3.5 Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения		
<b>Умеет</b>		
У.1. Разрабатывать алгоритмы для конкретных задач.	Осуществлять поддержку функционирования информационных систем;	да
У.2 Использовать программы для графического отображения алгоритмов.	Владение содержанием учебного материала, в котором обучающийся легко ориентируется, умеет применить теоретические знания при решении практических ситуаций	да
У.3 Определять сложность работы алгоритмов.	Разработки алгоритма поставленной задачи и реализации его средствами автоматизированного проектирования; разработки технологической документации	да
У.4 Работать в среде программирования.	Корректное применение разных видов информационных технологий для получения оптимального результата	да
У.5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.	Результативность применения инструментальных средств информационных технологий	да



У.6 Оформлять код программы в соответствии со стандартом кодирования.	Определение сложности работы алгоритмов в соответствии со стандартом кодирования	да
У.7 Выполнять проверку, отладку кода программы.	Выполнение работы в среде программирования на конкретном языке.	да

**Критерии оценивания:**

Оценка компетенции производится по интегральной оценке ОПОР. Каждый ОПОР оценивается 1 или 0, сумма этих оценок дает оценку компетенции: «да» или «нет». Уровень оценки компетенций производится суммированием количества ответов «да» в процентном соотношении от общего количества ответов.

Для перевода баллов в оценку применяется универсальная шкала оценки образовательных достижений

Таблица 3

**Универсальная шкала оценки образовательных достижений**

Процент результативности	Оценка уровня подготовки	
	оценка компетенций обучающихся	оценка уровня освоения дисциплин;
90 ÷ 100	высокий	<i>отлично</i>
70 ÷ 89	продвинутый	<i>хорошо</i>
50 ÷ 69	пороговый	<i>удовлетворительно</i>
менее 50	не освоены	<i>неудовлетворительно</i>

## 2.2. Матрица оценок образовательных достижений обучающихся

### 2.2.1. Оценка достижений обучающихся по результатам Экзамена учебной дисциплины ОП.04 Основы алгоритмизации и программирования

Группа ИСП-9-19

Ф.И.О. обучающихся	Компетенции ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.5 ПК 2.4 ПК 2.5															max балл	% выпол- нения	Оценка компетенции***		
	У1	У2	У3	У4	У5	У6	У7		31	32	33	34	35							
Умения и знания*																				
Величина баллов **	5	5	5	5	5	5	5		3	3	3	3	3					50	100 %	отлично

\*- включаете все умения и знания, которые указаны в ФГОС СПО специальности

\*\* - величину баллов за одно умение и знание определяете самостоятельно. Сумму баллов пересчитываете в проценты.

\*\*\*- при оценке компетенций необходимо воспользоваться «Универсальной шкалой оценки»:

90 – 100 %	высокий	отлично
70 – 89 %	продвинутый	хорошо
50 – 69 %	пороговый	удовлетворительно
менее 50 %	не освоены	неудовлетворительно

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Арктический государственный агротехнологический университет»  
Колледж технологий и управления  
Цикловая комиссия гуманитарных и естественных дисциплин

**Комплект  
контрольно-измерительных материалов  
для текущего контроля**

ОП.04 Основы алгоритмизации и программирования  
09.02.07 Информационные системы и программирование

## Материалы текущего контроля знаний и умений КОМПЛЕКТ ЗАДАНИЙ ДЛЯ ПРОВЕРОЧНОЙ РАБОТЫ

Максимальное время выполнения задания: 40 мин.

**Оборудование и оснащение:** *нет*

### Вариант № 1

Задание 1.

Цепочка из трех бусин, помеченных латинскими буквами, формируется по следующим правилам. В конце цепочки стоит одна из бусин А, В, С. На первом месте одна из бусин В, D, С, которой нет на третьем месте. В середине — одна из бусин А, С, Е, В, не стоящая на первом месте. Какая из перечисленных цепочек создана по этим правилам?

1)СВВ 2)ЕАС 3)BCD 4)BCB

Задание 2.

Определить значения переменных а и в после выполнения данного алгоритма, построив трассировочную таблицу.

$a=11$

$a=112-a*9$

$v=51-33+a*2$

$a=a-2$

$a=a-v/a+39$

где / - операция целочисленного деления.

Задание 3.

Сколько раз выполнится тело цикла, чему равны значения переменных а и в после выполнения фрагмента алгоритма?

Задание 4.

У исполнителя Утроитель две команды, которым присвоены номера:

1. Вычти 1

2. Умножь на3

Первая из них уменьшает число на экране на 1, вторая - утраивает его. Запишите порядок команд в алгоритме получения из числа 5 числа 26, содержащем не более 5 команд, указывая лишь номера команд.

Задание 5.

Алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала записывается буква, следующая в алфавите после первой буквы исходной цепочки, затем две последние буквы исходной цепочки в обратном порядке и, наконец, первая буква исходной цепочки. Получившаяся цепочка является результатом работы алгоритма. Например, если исходная цепочка символов была СОН, то результатом алгоритма будет цепочка ТНОС.

Дана цепочка символов КОШ. Какая цепочка символов получится, если к данной цепочке применить алгоритм трижды (т.е. алгоритм применяется к данной цепочке, затем к результату его работы, а затем ко второму результату работы алгоритма)?

Русский алфавит:

АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

### Вариант 2.

Задание 1.

В формировании цепочки из четырех бусин используются некоторые правила. В конце цепочки стоит одна из бусин Р, N, Т, О. На первом — одна из бусин Р, R, Т, О; которой нет на третьем месте. На третьем месте — одна из бусин О, Р, Т, не стоящая в цепочке последней.

Какая из перечисленных цепочек могла быть создана с учетом этих правил?

1)PORT 2)TTTO 3)TTOO 4)OORO

Задание 2.

Определить значения переменных  $a$  и  $v$  после выполнения данного алгоритма, построив трассировочную таблицу.

$$a=24$$

$$v=18$$

$$a=a+v/3+16$$

$$v=a+2*v$$

$$a=v/a+v/2$$

где / - операция целочисленного деления.

Задание 3.

Сколько раз выполнится тело цикла, чему равны значения переменных  $a$  и  $v$  после выполнения фрагмента алгоритма?

Задание 4.

У исполнителя Утроитель две команды, которым присвоены номера:

1. Вычти 2

2. Умножь на 3

Первая из НИХ уменьшает число на экране на 2, вторая - утраивает его. Запишите порядок команд в программе получения из 11 числа 13, содержащей не более 5 команд, указывая лишь номера команд.

Задание 5.

Алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала записываются первые две буквы исходной цепочки в обратном порядке, затем буква, следующая в алфавите за последней буквой исходной цепочки, и, наконец, исходная цепочка символов, записанная в обратном порядке. Получившаяся цепочка является результатом работы алгоритма. Например, если исходная цепочка символов была СОН, то результатом алгоритма будет цепочка ОСОНОС.

Дана цепочка символов НИЛ. Какая цепочка символов получится, если к данной цепочке применить алгоритм дважды (т.е. алгоритм применяется к данной цепочке, затем к результату его работы)?

Русский алфавит:

АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

**Вариант 3.**

Задание 1.

Для составления цепочек используются бусины, помеченные буквами А, В, С, D, Е. На первом месте в цепочке стоит одна из бусин А, С, Е. На втором — любая гласная, если первая буква согласная, и любая согласная, если первая гласная. На третьем месте — одна из бусин С, D, Е, не стоящая в цепочке на первом месте. Какая из перечисленных цепочек создана по этим правилам?

1)СВЕ 2)ADD 3)ECE 4)EAD

Задание 2.

Определить значения переменных  $a$  и  $v$  после выполнения данного алгоритма, построив трассировочную таблицу.

$$a=15$$

$$v=2*a+24$$

$$a= v/2+a$$

$$v=2*a+2*v+9$$

$$a=v/a+2*a$$

где / - операция целочисленного деления.

Задание 3.

Сколько раз выполнится тело цикла, чему равны значения переменных  $I$  и  $A$  после выполнения фрагмента алгоритма?

Задание 4.

У исполнителя Калькулятор две команды, которым присвоены номера:

1. Прибавь 2

2. Умножь на 3

Выполняя первую из них, Калькулятор прибавляет к числу на экране 2, а выполняя вторую, утраивает его. Запишите порядок команд в программе получения из 0 числа 28, содержащей не более 6 команд, указывая лишь номера команд.

Задание 5.

Алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала записывается буква, предшествующая в алфавите первой букве исходной цепочки, затем исходная цепочка символов, записанная в обратном порядке. Получившаяся цепочка является результатом работы алгоритма. Например, если исходная цепочка символов была СОН, то результатом алгоритма будет цепочка РНОС.

Дана цепочка символов РОК. Какая цепочка символов получится, если к данной цепочке применить алгоритм трижды (т.е. алгоритм применяется к данной цепочке, затем к результату его работы, а затем ко второму результату работы)?

Русский алфавит:

АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

**Вариант 4.**

Задание 1.

Цепочка из трех бусин формируется по следующим правилам. На первом месте в цепочке стоит одна из бусин А, Б, В. На втором — одна из бусин Б, В, Г. На третьем месте — одна из бусин А, В, Г, не стоящая в цепочке на первом или втором месте. Какая из следующих цепочек создана по этим правилам?

1)АГБ 2)ВАГ 3)БГГ 4)ББГ

Задание 2.

Определить значения переменных а и в после исполнения данного алгоритма, построив трассировочную таблицу.

$a=48$

$v=a/4-5$

$a= a-v$

$v=2*a+37$

$a=v/a+2*a$

где / - операция целочисленного деления.

Задание 3.

Сколько раз выполнится тело цикла, чему равны значения переменных I, A и R после выполнения фрагмента алгоритма?

Задание 4.

Исполнитель Калькулятор имеет только две команды, которым присвоены номера:

1. Вычти 3

2. Умножь на 2

Выполняя команду номер 1, Калькулятор вычитает из числа на экране 3, а выполняя команду номер 2, умножает число на экране на 2. Напишите программу, содержащую не более 5 команд, которая из числа 5 получает число 25. Укажите лишь номера команд.

Задание 5.

Алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала записывается буква, предшествующая в алфавите последней букве исходной цепочки, затем первые две буквы исходной цепочки, и наконец, буква, следующая в алфавите за первой буквой исходной цепочки символов. Получившаяся цепочка является результатом работы алгоритма. Например, если исходная цепочка символов была СОН, то результатом алгоритма будет цепочка МСОТ.

Дана цепочка символов АРТ. Какая цепочка символов получится, если к данной цепочке применить алгоритм дважды (т.е. алгоритм применяется к данной цепочке, затем к результату его работы)?

Русский алфавит:  
АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

### КОМПЛЕКТ ЗАДАНИЙ ДЛЯ подготовки докладов

Форма контроля	Поисковое задание
Вид контроля	оперативный

Условия выполнения задания  
Аудитория, 20 минут.

#### Инструкция для студентов

1. Последовательность и условия выполнения задания задания выполняются в произвольном порядке
2. Вы можете воспользоваться: нет
3. Максимальное время выполнения задания: 20 мин.

Оборудование и оснащение: ПК, Интернет

#### Темы докладов:

История и классификация языков программирования.  
Системы программирования.  
Процедурные и непроцедурные языки программирования.  
Объектно-ориентированное программирование.  
Язык программирования Паскаль. История создания. Использование в современности.  
Биография Никлауса Вирта.

### КОМПЛЕКТ ЗАДАНИЙ ДЛЯ ПРОВЕРОЧНОЙ РАБОТЫ

Форма контроля	Проверочная работа
Вид контроля	оперативный

Условия выполнения задания  
Аудитория, 15 минут.

#### Инструкция для студентов

1. Последовательность и условия выполнения задания задания выполняются в произвольном порядке
2. Вы можете воспользоваться: нет
3. Максимальное время выполнения задания: 15 мин.

Оборудование и оснащение: нет

#### ВАРИАНТ 1

Составьте программу вычисления значения функции, используя полное ветвление:

$$y = \begin{cases} \frac{4x^2 + 1}{x - 5}, & \text{если } x < 5, \\ 3x^2 - 2, & \text{если } x \geq 5. \end{cases}$$

## ВАРИАНТ 2

Составьте программу вычисления значения функции, используя полное ветвление:

$$y = \begin{cases} \frac{5x^2 + 2}{x + 4}, & \text{если } x > -4, \\ 3x^2 + 7, & \text{если } x \leq -4 \end{cases}$$

## ВАРИАНТ 3

Составьте программу вычисления значения функции, используя полное ветвление:

$$y = \begin{cases} \frac{7x^2 - 1}{2x + 6}, & \text{если } x < -3, \\ 4x^2 - 5, & \text{если } x \geq -3 \end{cases}$$

## ВАРИАНТ 4

Составьте программу вычисления значения функции, используя полное ветвление:

$$y = \begin{cases} \frac{9x^2 + 5}{3x + 12}, & \text{если } x < -4, \\ 4x^2 - 7, & \text{если } x \geq -4 \end{cases}$$

## КОМПЛЕКТ ЗАДАНИЙ ДЛЯ ПРОВЕРОЧНОЙ РАБОТЫ

Форма контроля	Проверочная работа
Вид контроля	оперативный

Условия выполнения задания

Аудитория, 15 минут.

Инструкция для студентов

1. Последовательность и условия выполнения задания задания выполняются в произвольном порядке
2. Вы можете воспользоваться: нет
3. Максимальное время выполнения задания: 15 мин.

Оборудование и оснащение: нет

Источники

Семакин И.Г., Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования – М.: Издательский центр «Академия», 2014 – 304с.

Вариант 1

1. Составить программу вычисления суммы и произведения четных чисел из промежутка от 1 до 10.

Вариант 2

2. Составьте программу вывода на экран всех нечетных трехзначных чисел.

Вариант 3

3. Дано целое число N(больше 0). Найти сумму  $1+1/2+1/3+...+1/N$ (вещественное число)

Пакет преподавателя

Вариант 1

program jjj;

var a: array [1..10] of integer; i,summa,proizv:integer;



```

begin
for i:=1 to 10 do begin
  Writeln('напишите элемент массива номер ',i); read(a[i]);
  end;
For i:=1 to 10 do begin
  if a[i] mod 2=0 then summa:=summa+a[i] else
  if (a[i]<=10) and (a[i]>=1) then proizv:=proizv*a[i];
  end;
Writeln(summa,' ',proizv); readln;
end.

```

#### Вариант 2

```

var i,s: integer;
begin
i := 100;
s := 0;
repeat
s := s + i;
i := i + 5;
until i <= 1000;
writeln(s);
end.

```

#### Вариант 3

```

var H: real;
  N, i: integer;
begin
  H := 0;
  readln(N);
  for i := 1 to N do
    H := H + 1/i;
  writeln(H:6:4)
end.

```

### КОМПЛЕКТ ЗАДАНИЙ ДЛЯ ОПРОСА

Форма контроля	Проверочная работа
Вид контроля	оперативный

Условия выполнения задания

Аудитория, 10 минут.

#### Инструкция для студентов

1. Последовательность и условия выполнения задания  
студенты отвечают на вопросы
2. Вы можете воспользоваться: нет
3. Максимальное время выполнения задания: 10 мин.

Оборудование и оснащение: нет

#### Источники

Семакин И.Г., Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования – М.: Издательский центр «Академия», 2014 – 304с.

- 1) Оператор цикла с предусловием, формат команды.
  - 2) Заголовок и тело цикла с предусловием.
  - 3) Оператор цикла с параметром, формат команды.
  - 4) Заголовок и тело цикла с параметром.
  - 5) Шаг изменения счетчика цикла с параметром.
  - 6) Использование служебных слов to и downto в заголовке цикла с параметром.
  - 7) Оператор цикла с постусловием, формат команды, условие и тело цикла.
  - 8) Организация повтора программы с использованием цикла с постусловием.
  - 9) Использование составного оператора в теле цикла.
- Максимальное время выполнения задания: 4 часа.

Оборудование и оснащение: ПК, Интернет

### КОМПЛЕКТ ЗАДАНИЙ ДЛЯ практического задания

1. Презентация
2. Вы можете воспользоваться конспектами, учебником
3. Максимальное время выполнения задания: 2 часа.
4. Перечень раздаточных и дополнительных материалов методические указания к лабораторным работам

Оборудование и оснащение: ПК

**Цель работы:** Научиться составлять циклические алгоритмы, создавать программы, используя полученные знания.

**Содержание работы.**

**Основные понятия.**

1. Графический способ представления циклических алгоритмов (метод блок-схемы).

Проверка условий	
Цикл с параметром	

2. Алгоритм, в котором вычисления повторяются по одной и той же совокупности формул, называется циклическим

3. Существуют следующие конструкции для организации циклов:

- цикл с предусловием;
- цикл с постусловием;
- цикл со счетчиком.

4. Графическое представление циклических алгоритмов



5. Циклы с предусловием используются тогда, когда выполнение цикла связано с некоторым логическим условием. Оператор цикла с предусловием имеет две части: условие выполнения цикла и тело цикла. При выполнении оператора цикла определенная группа операторов выполняется до тех пор, пока

Источники

Семакин И.Г., Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования – М.: Издательский центр «Академия», 2014 – 304с.

П

## Практическое задание

определенное в операторе условие истинно. Если условие сразу ложно, то оператор не выполнится ни разу.

Составить и записать алгоритм решения задачи в графическом и словесно-формульном виде

**Исходные данные:**

Дано целое число  $N$ . Вычислить значение выражения  $1 + \frac{2}{2!} + \frac{3}{3!} + \dots + \frac{N}{N!}$ , результат вывести как действительное число.

**Решение:**

$$1 \quad N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot N$$

Значение выражения будем рассчитывать поэтапно.

На первом шаге к значению суммы  $S$  добавим  $1 = \frac{1}{1!} = \frac{1}{1}$

На втором  $\frac{2}{2!} = \frac{2}{1 \cdot 2}$

На третьем  $\frac{3}{3!} = \frac{3}{1 \cdot 2 \cdot 3}$

...

На  $N$ -ном  $\frac{N}{N!} = \frac{N}{1 \cdot 2 \cdot \dots \cdot N}$

В зависимости от заданного пользователем значения  $N$  количество шагов в алгоритме может быть разным, поэтому для определения шага добавим промежуточную переменную  $I$ , которая будет меняться от 1 до  $N$

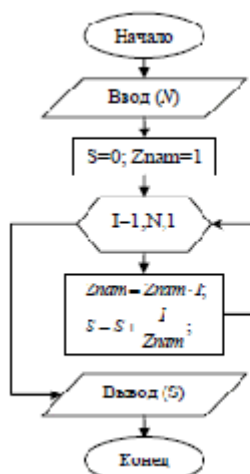
Так как к значению суммы на каждом шаге мы добавляем какое-то число, необходимо определить начальное значение, не влияющее на результат. Ясно, что это  $S = 0$ .

На каждом шаге считать факториал трудно, поэтому обратим внимание, что на любом этапе вычислений значение факториала отличается от предыдущего на число, равное значению шага, значит введем переменную для факториала:  $Z_{\text{факт}} = 1$ .

Получили общие формулы:

$$Z_{\text{факт}} = Z_{\text{факт}} \cdot I;$$

$$S = S + \frac{I}{Z_{\text{факт}}};$$



1. Начало;
2. Ввод ( $N$ );
3.  $S=0$ ;
4.  $Z_{\text{факт}}=1$ ;
5. Для  $I = 1$  до  $N$  выполнить  
    кц  
     $Z_{\text{факт}} = Z_{\text{факт}} \cdot I$ ;  
     $S = S + \frac{I}{Z_{\text{факт}}}$ ;  
    кц
6. Вывод ( $S$ );
7. Конец.

Дополнительно на оценку

### **Задания к практической работе.**

1 Даны действительные числа  $x$ ,  $y$ . Вывести в порядке возрастания все целые числа, расположенные между  $x$  и  $y$ , а также количество этих чисел.

2 Даны действительные числа  $x$ ,  $y$ . Вывести в порядке убывания все целые числа, расположенные между  $x$  и  $y$ , а также количество этих чисел.

3 Дано действительное число – цена 1 кг конфет. Вывести стоимость 1, 2 ... 15 кг конфет.

### **КОМПЛЕКТ ЗАДАНИЙ ДЛЯ практического задания**

2. Вы можете воспользоваться конспектами, учебником

3. Максимальное время выполнения задания: 2 часа.

4 Перечень раздаточных и дополнительных материалов *методические указания к лабораторным работам*

**Оборудование и оснащение: ПК**

### **Источники**

*Семакин И.Г., Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования – М.: Издательский центр «Академия», 2014 – 304с.*

### **Практическое задание**

**Массив** - это структура данных, что представляет собой совокупность фиксированного размера и конфигурации упорядоченных однородных независимых переменных.

Массив относится к так называемым структурированным данным, то есть таких, что имеют фиксированную внутреннюю структуру (организацию).

Массив характеризуется:

1. Количеством размерностей (количеством координат, необходимых для определения местонахождения нужного элемента массива).

2. Общим идентификатором (именем) для всех элементов массива.

3. Индексом или совокупностью индексов, которые определяют каждый отдельный элемент массива.

**Одномерный массив (вектор)** - имеет одну размерность

При обращении к отдельному элементу массива необходимо указать его индекс (местонахождение в массиве):

$A[7]$   $i:=7$ ;  $A[i]$

Здесь  $i$  - индекс элемента массива, может быть только целого или натурального типа

**Двумерные массивы (матричные)** - имеют две размерности,  $m*n$ . Доступ к отдельному элементу массива осуществляется путем определения двух его координат: номера строки  $i=1..m$  и столбца  $j=1..n$

Операции предоставления выполняются аналогично:

$a[3,9]=8$ ;  $B:=A[1,1]$

**В трехмерном массиве** для доступа к элементу необходимо указать три индекса  $A[i,j,k]$ .

Можно создавать массивы с большей размерностью, но работа с массивами, размерность которых превышает 3, существенным образом усложняет алгоритм, поэтому, по возможности, необходимо избегать организации подобных структур данных.

### **Сортировка массивов**

Методы сортировки можно разбить в соответствии с определяющими их принципами на три основные группы:

1. Сортировка с помощью вставки (by Insertion) или с помощью включения

2. Сортировка с помощью выбора (by Selection) или с помощью выделения

3. Сортировка с помощью обмена (by Exchange) или пузырьковая.

Каждая группа имеет прямой метод (самый простой) и улучшенный(усложненный) методы сортировки

### **I. Сортировка с помощью вставки**

Принцип сортировки: массив распределяется на отсортированную и неотсортированную части. На первом шаге за отсортированную часть (последовательность) принимается первый элемент массива. Каждый следующий элемент из неотсортированной части вставляем в заранее отсортированную последовательность так, чтобы эта последовательность оставалась отсортированной.

При этом надо:

1. Найти место, куда нужно вставить этот элемент

2. Сдвинуть элементы, которые стоят справа в отсортированной части на одну позицию вправо.

3. На освобожденное место поставить элемент, который анализируется (вставляется).

Два способа выполнения этих действий:

1) каждый следующий элемент сравнивается с элементами в отсортированной части, находится место вставки, все следующие элементы сдвигаются на одну позицию вправо и после этого вставляется элемент;

2) элемент, который вставляется, последовательно, слева направо, сравнивается с любым из элементов в отсортированной части. Если нужно, элемент в отсортированной части сразу сдвигается на одну позицию вправо. Как только найдено нужное место вставки, элемент, который анализируется, вставляется на нужную позицию.

### **II. Сортировка с помощью прямого выбора**

Принцип сортировки: массив также делится на отсортированную и неотсортированную части. На первом шаге весь массив - неотсортированный. В неотсортированной части находится минимальный (или максимальный) элемент и меняется местами с первым элементом неотсортированной части. Граница отсортированной части сдвигается вправо на 1. Процедура выполняется циклически, n-1 раз (последний элемент передвигать не надо)

### **III. Сортировка с помощью прямого обмена (пузырьковая)**

Принцип сортировки: слева направо поочередно направляется сравнение двух соседних элементов. Если они не упорядочены между собою, то меняются местами. В базовом алгоритме прохождения массива и очередное приведение в порядок повторяются n-1 раз.

### **ПРИМЕР ПРОГРАММЫ**

#### **1. Нахождение среднего арифметического ста чисел**

Program N1;

var M: Array [1..100] of Integer; A: Real; I: Byte;

begin Randomize;

For I := 1 to 100 do M[I] := Random(500);

For I := 1 to 100 do

A := A + M[I];

A := A / 100;

Write('Среднее арифметическое всех чисел массива: ', A);

end.

#### **2. Найдите максимальное число среди введенных чисел и выведите его на экран.**

Program N2;

var M: Array [1..10] of Integer; Max: Integer; I: Byte;

begin Writeln('Введите 10 чисел: ');

For I := 1 to 10 do

begin Write('N', i, ': ');

Readln(M[i]);

end;

Max := M[1];

```
For I := 1 to 10 do
if Max < M[i] then Max := M[i];
Write('Максимальное число: ', Max);
Readln;
end
```

### **Задание №1**

Найдите среднее арифметическое первых двухсот чисел.

### **Дополнительные задания на оценку**

#### **1. Задание № 2**

Найдите минимальное число среди введенных 10 чисел.

#### **2. Задание № 3**

Найдите максимальное число среди введенных 15 чисел.

## **КОМПЛЕКТ ЗАДАНИЙ для практического задания**

2. Вы можете воспользоваться конспектами, учебником

3. Максимальное время выполнения задания: 2 часа.

4 Перечень раздаточных и дополнительных материалов *методические указания к лабораторным работам*

**Оборудование и оснащение: ПК**

### **Источники**

*Семакин И.Г., Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования – М.: Издательский центр «Академия», 2014 – 304с.*

### **Практическое задание**

Для формирования графических изображений в Турбо Паскале предназначен стандартный модуль GRAPH. В нем содержится 79 графических процедур, функций, десятки стандартных констант и типов данных. В состав модуля GRAPH входит ряд программ драйверов для наиболее распространенных видеоадаптеров: CGA, MCGA, EGA, VGA, HERCULES, AT&T, 3270 PC. Эти драйверы хранятся в файлах с расширением BGI.

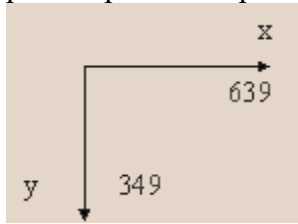
Скелет графической программы:

```
program имя;
USES GRAPH; {подключение модуля}
var GR,GM:integer;
begin
GR:=detect;
initgraph(gr,gm,'); {вызов процедуры GRAPH}
графические операторы
readln;
closegraph;
end.
```

INITGRAPH(драйвер, режим, 'путь') - подключает графический режим. Драйвер - это переменная целого типа, определяющая тип графического дисплея. Режим - это переменная типа integer, определяющая режим работы графического адаптера(GM). Путь - указывает путь программам с TPU.

DETECT - режим автоматического определения типа.

CLOSEGRAPH - прекращает работу графического адаптера и восстанавливает текстовый режим работы экрана.



### Цвет

- 0 - черный
- 1 - синий
- 2 - зеленый
- 3 - голубой
- 4 - красный
- 5 - лиловый
- 6 - коричневый
- 7 - светло-серый
- 8 - темно-серый
- 9 - светло-синий
- 10 - светло-зеленый
- 11 - светло-голубой
- 12 - светло-красный
- 13 - светло-лиловый
- 14 - желтый
- 15 - белый

### ГРАФИЧЕСКИЕ ОПЕРАТОРЫ:

- 3. SETCOLOR(цвет) - устанавливает текущий цвет для выводимых линий и символов;
- 4. SETBKCOLOR(цвет) - устанавливает цвет фона;
- 5. MOVETO(x,y) - переводит курсор в позицию x,y, точка при этом не высвечивается;
- 6. LINETO(x,y) - рисует линию от текущей позиции курсора до x,y;
- 7. LINEREL(Dx,Dy) - рисует линию от текущей позиции на относительное расстояние Dx,Dy;
- 8. PUTPIXEL(x,y,цвет) - выводит точку;
- 9. LINE(x1,y1,x2,y2) - выводит линию;
- 10. CIRCLE(x,y,r) - выводит окружность;
- 11. ARC(x,y,начальный угол,конечный угол,r) - выводит дугу (углы измеряются в градусах против часовой стрелки);
- 12. ELLIPSE(x,y,нач.угол,кон.угол,Rx,Ry) - эллипс;
- 13. RECTANGLE(x1,y1,x2,y2) - прямоугольник, где x1,y1 - координаты левого верхнего угла, x2,y2 - правого нижнего угла;
- 14. BAR(x1,y1,x2,y2) - закрашенный прямоугольник;
- 15. BAR3D(x1,y1,x2,y2,глубина,false или true) - параллелепипед; где x1,y1 - координаты левого верхнего угла; x2,y2 - координаты правого нижнего угла, true - верхняя грань есть, false - верхняя грань отсутствует.
- 16. FILLELLIPSE(x,y,Rx,Ry) - закрашенный эллипс;
- 17. SECTOR(x,y,нач.угол,кон.угол,Rx,Ry) - сектор закрашенный;
- 18. SETFILLESTYLE(штриховка,цвет) - каким цветом или какими штрихами будет закрашена замкнутая область;

### Штриховка:

- 0 - фоном
- 1 - сплошная
- 2 - \_ \_ \_ \_
- 3 - ///
- 4 - толстыми///
- 5 - \\\
- 6 - толстыми \\\
- 7 - ++++
- 8 - х х х
- 9- квадратами
- 10 - редкими точками
- 11 - частыми точками
- 12 - пользовательские

3. FLOODFILL(х,у,цвет,цвет границы) - штрихует произвольную замкнутую область;

**Пример графических программ:**

*Программа №1.*

```
program g1;  
uses graph;  
var gr,gm:integer;  
begin  
gr:=detect;  
initgraph(gr,gm,' ');  
setcolor(3);  
setbkcolor(14);  
setfillstyle(8,4);  
fillellipse(100,100,30,50);  
setfillstyle(2,2);  
circle(200,200,50);  
floodfill(200,200,3);  
readln;  
closegraph;  
end.
```



МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Арктический государственный агротехнологический университет»  
Колледж технологий и управления  
Цикловая комиссия гуманитарных и естественных дисциплин

**Комплект материалов  
для проведения контрольных работ**

ОП.04 Основы алгоритмизации и программирования  
09.02.07 Информационные системы и программирование

Якутск – 2024 г.

## КОМПЛЕКТ ЗАДАНИЙ ДЛЯ ПРОВЕРОЧНОЙ РАБОТЫ

### Тема 3.3. Массивы

3. Максимальное время выполнения задания: 20 мин.

**Оборудование и оснащение:** *нет*

#### Источники

*Программирование на языке C++: Учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 512 с.*

#### Вариант 1

Написать программу на языке программирования С. В массиве из 20 целочисленных элементов, заданных случайным образом из отрезка  $[-15;15]$  найти максимальный элемент и его номер.

#### Вариант 2

Написать программу на языке программирования С. В массиве из  $N$  (вводится с клавиатуры) целочисленных элементов, заданных случайным образом из отрезка  $[-5;5]$  подсчитать количество нечетных положительных элементов, вывести найденные элементы на печать.

#### Вариант 3

Написать программу на языке программирования С, которая определяет в целочисленной матрице, состоящей из элементов из отрезка  $[-5;5]$ , номер строки, которая содержит наибольшее количество элементов, равных нулю.

## КОМПЛЕКТ ЗАДАНИЙ ДЛЯ КОНТРОЛЬНОЙ РАБОТЫ

#### Условия выполнения задания

Аудитория, 15 минут.

#### Инструкция для студентов

1. Последовательность и условия выполнения задания задания выполняются в произвольном порядке
2. Вы можете воспользоваться: *нет*
3. Максимальное время выполнения задания: 15 мин.

**Оборудование и оснащение:** *нет*

#### Вариант 1

1. Что означает команда языка Pascal `Ellipse(50,100,150,200)`
2. Что означает команда языка Pascal `SetBrushColor(clRed)`
3. Написать команду рисования прямоугольника, заливка зеленым цветом

#### Вариант 2

1. Что означает команда языка Pascal `SetPixel(100,200,clRed)`
2. Что означает команда языка Pascal `SetPenColor(clGreen)`
3. Написать команду рисования эллипса, заливка синим цветом

#### Вариант 3

1. Что означает команда языка Pascal `Rectangle(50,100,250,300)`
2. Что означает команда языка Pascal `Uses GraphABC`
3. Написать команду рисования круга, заливка красным цветом

Вариант 4

1. Что означает команда языка Pascal FloodFill(100,150,clBlue)
2. Что означает команда языка Pascal SetPenWidth(5)
3. Написать команду рисования линии красным цветом

### КОМПЛЕКТ ЗАДАНИЙ ДЛЯ СОЗДАНИЯ И ЗАЩИТЫ ПРЕЗЕНТАЦИИ

<b>Форма контроля</b>	Исследовательское групповое задание
<b>Вид контроля</b>	оперативный

**Условия выполнения задания**

Аудитория, 2 часа.

#### **Инструкция для студентов**

1. Последовательность и условия выполнения задания  
*Студенты работают в группе по 3 человека, создают презентацию, затем защищают ее.*
2. Вы можете воспользоваться: *Конспект лекции, Интернет*
3. Максимальное время выполнения задания: 2 часа.

**Оборудование и оснащение:** *ПК, Интернет*

#### **Вопросы, которые необходимо раскрыть при создании презентации:**

- 1) Описание файловых переменных.
- 2) Установление связи между программой и файлом на диске.
- 3) Открытие файла для чтения, записи, добавления данных.
- 4) Запись данных в файл, чтение данных из файла.
- 5) Завершение работы с файлом.

### КОМПЛЕКТ ЗАДАНИЙ ДЛЯ ОПРОСА

<b>Форма контроля</b>	Фронтальный опрос
<b>Вид контроля</b>	оперативный

**Условия выполнения задания**

Аудитория, 7 минут.

#### **Инструкция для студентов**

1. Последовательность и условия выполнения задания  
*студенты отвечают на вопросы*
2. Вы можете воспользоваться: *нет*
3. Максимальное время выполнения задания: 7 мин.

**Оборудование и оснащение:** *ПК*

- 1) Структурированный тип запись, определение, назначение.
- 2) Описание типа запись, используемые служебные слова.
- 3) Компоненты записи и их типы.
- 4) Массив записей, описание, назначение.
- 5) Составные имена.
- 6) Оператор присоединения и его область действия.

### КОМПЛЕКТ ЗАДАНИЙ ДЛЯ ОПРОСА

<b>Форма контроля</b>	Фронтальный опрос
<b>Вид контроля</b>	оперативный

**Условия выполнения задания**

Аудитория, 10 минут.

**Инструкция для студентов**

1. Последовательность и условия выполнения задания  
*студенты отвечают на вопросы*
2. Вы можете воспользоваться: *нет*
3. Максимальное время выполнения задания: *10 мин.*

**Оборудование и оснащение: ПК**

**Источники**

*Семакин И.Г., Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования – М.: Издательский центр «Академия», 2014 – 304с.*

- 1) Описание одномерного массива.
- 2) Использование операторов цикла при решении задач обработки массивов.
- 3) Выделение отдельного элемента одномерного массива.
- 4) Поиск элементов в массиве, кратных заданному числу (условие кратности). Привести фрагмент программы.
- 5) Описание двумерного массива.
- 6) Использование операторов цикла при решении задач обработки массивов.
- 7) Выделение отдельного элемента двумерного массива.
- 8) Свойства диагональных элементов квадратной матрицы.

**ОЦЕНОЧНОЕ СРЕДСТВО № 10**

**КОМПЛЕКТ ЗАДАНИЙ ДЛЯ ОПРОСА**

<b>Форма контроля</b>	Фронтальный опрос
<b>Вид контроля</b>	оперативный

**Объекты оценки:**

**Условия выполнения задания**

Аудитория, 10 минут.

**Инструкция для студентов**

1. Последовательность и условия выполнения задания  
*студенты отвечают на вопросы*
2. Вы можете воспользоваться: *нет*
3. Максимальное время выполнения задания: *10 мин.*

**Оборудование и оснащение: ПК**

**Источники**

*Семакин И.Г., Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования – М.: Издательский центр «Академия», 2014 – 304с.*

- 1) Понятие подпрограммы, назначение подпрограмм.
- 2) Виды подпрограмм в языке программирования Pascal.

- 3) Состав подпрограммы.
- 4) Вызов подпрограммы.
- 5) Функции, описание функции, заголовок функции, тело функции.
- 6) Процедуры, описание процедуры, заголовок процедуры, тело процедуры.
- 7) Параметры-значение и параметры-переменные процедуры.
- 8) Формальные и фактические параметры, работа механизма передачи параметров.

#### **Критерии оценивания:**

*Оценки "отлично"* заслуживает студент, обнаруживший всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной программой. Как правило, оценка "отлично" выставляется студентам, усвоившим взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.

*Оценки "хорошо"* заслуживает студент обнаруживший полное знание учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка "хорошо" выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.

*Оценки "удовлетворительно"* заслуживает студент, обнаруживший знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по специальности, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка "удовлетворительно" выставляется студентам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.

*Оценка "неудовлетворительно"* выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий. Как правило, оценка "неудовлетворительно" ставится студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине.

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Арктический государственный агротехнологический университет»  
Колледж технологий и управления  
Цикловая комиссия гуманитарных и естественных дисциплин

**Комплект  
контрольно-оценочных средств  
для промежуточной аттестации по результатам освоения дисциплины**

ОП.04 Основы алгоритмизации и программирования  
09.02.07 Информационные системы и программирование

Якутск – 2019 г.

**ПЕРЕЧЕНЬ ВОПРОСОВ для проведения экзамена  
ПО РАЗДЕЛУ 1**

1. Дайте определение алгоритма и приведите примеры.
2. Что такое алгоритмизация?
3. Приведите пример математического выражения и составьте алгоритм его вычисления.
4. Поясните на примерах свойства алгоритма.
5. Как вы понимаете свойство конечности алгоритма? Приведите примеры.
6. Как вы понимаете свойство массовости алгоритма? Приведите примеры.
7. Что такое линейный алгоритм? Приведите примеры.
8. Что такое циклический алгоритм? Приведите примеры.
9. Напишите циклический алгоритм и укажите в нем тело цикла.
10. Как происходит окончание циклического алгоритма?
11. Что такое разветвляющийся алгоритм? Приведите примеры.
12. Как в алгоритме записывается условие?
13. Как записывается полная форма разветвляющегося алгоритма? Приведите примеры.
14. Как записывается неполная форма разветвляющегося алгоритма? Приведите примеры.
15. Что такое вспомогательный алгоритм? Приведите примеры.
16. Зачем нужна блок-схема алгоритма?
17. Придумайте пример алгоритма и представьте его в виде блок-схемы.
18. Какие стадии разработки алгоритма вы знаете и в чем их суть?
19. Приведите пример разработки алгоритма по стадиям в виде двух блок-схем.
20. Порядок выполнения логических операций.
21. Установить, какие из предложений являются логическими высказываниями, а какие — нет (объясните почему)
22. Указать, какие из высказываний истинны, какие — ложны, а какие относятся к числу тех, истинность которых трудно или невозможно установить.
23. Составите таблицы истинности для логических выражений
24. Составить логическую функцию для заданной таблицы истинности
25. Алфавит языка программирования
26. Оператор. Переменная. Модуль
27. Синтаксис, семантика, функция языка программирования
28. Языки программирования высокого уровня
29. Величины в языках программирования
30. Файл. Выражение. Рекурсия

## ПЕРЕЧЕНЬ ВОПРОСОВ ПРОВЕРОЧНОЙ РАБОТЫ по теме «ОСНОВНЫЕ ЭЛЕМЕНТЫ ЯЗЫКА»

1. Служебные слова
2. Структура программы
3. Расширение файлов программ языка программирования
4. Запись выражений на языке программирования
5. Элементов алгоритмический язык.
6. Что входит в набор основных символов языка программирования?
7. Стандартные типы
8. Что такое массив?
9. Выражения-константы
10. Описания переменных
11. Приоритет операций
12. Операторы ввода-вывода
13. Определение в записи операторов синтаксических ошибок
14. Какие типы циклов существуют в языке программирования
15. Структурированная программа
16. Типы данных
17. Языки с полиморфным типом данных
18. Стандартизация –

### ПЕРЕЧЕНЬ ВОПРОСОВ К ЗАЧЕТУ

(вариант № 2)

1. Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов.
2. Общие принципы построения алгоритмов. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические.
3. Данные: понятие и типы. Основные базовые и структурированные типы данных, их характеристика.
4. Основы алгебры логики. Логические операции с высказываниями: конъюнкция, дизъюнкция, инверсия.
5. Законы логических операций.
6. Таблицы истинности.
7. Эволюция языков программирования.
8. Классификация языков программирования.
9. Понятие системы программирования.
10. Исходный, объектный и загрузочный модуль.
11. Интегрированная среда программирования
12. Методы программирования: структурный, модульный, объектно-ориентированный. Достоинства и недостатки методов программирования.
13. Общие принципы разработки программного обеспечения.
14. Жизненный цикл программного обеспечения.
15. Структурная схема программы на алгоритмическом языке.
16. Лексика языка. Переменные и константы. Типы данных. Выражения и операции.
17. Синтаксис операторов: присваивания, ввода-вывода, безусловного и условного переходов, циклов.
18. Составной оператор. Вложенные условные операторы.
19. Циклические конструкции. Циклы с предусловием и постусловием.
20. Структурированные типы данных. Массивы, строки, множества.
21. Объявление массива. Ввод и вывод одномерных массивов и двумерных массивов. Обработка массивов. Стандартные функции для массива целых и вещественных чисел.
22. Объявление строковых типов данных. Операции со строками. Стандартные функции и процедуры для работы со строками.
23. Объявление множества. Операции над множествами.
24. Процедуры и функции, их сущность, назначение, различие.



25. Организация процедур, стандартные процедуры. Процедуры, определенные пользователем: синтаксис, передача аргументов.

26. Формальные и фактические параметры. Процедуры с параметрами, описание  
Процедур.

26. Функции: способы организации и описание. Вызов функций, рекурсия. Стандартные функции.

27. Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа. Запись в файл и чтение из файла последовательного доступа.

28. Составление блок-схем алгоритмов.

29. Составление программ линейной структуры.

30. Составление программ разветвляющейся структуры.

31. Составление программ циклической структуры.

32. Обработка одномерных массивов.

33. Обработка двумерных массивов.

34. Работа со строковыми переменными.

35. Работа с данными типа множество.

36. Организация процедур.

37. Организация функций.

38. Использование процедур.

39. Использование функций.

40. Работа с файлом последовательного доступа.

41. Работа с файлом произвольного доступа.

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Арктический государственный агротехнологический университет»  
Колледж технологий и управления  
Цикловая комиссия гуманитарных и естественных дисциплин

**Комплект  
контрольно-оценочных средств  
для проведения тестового контроля**

ОП.04 Основы алгоритмизации и программирования  
09.02.07 Информационные системы и программирование

Якутск – 2019 г.

## Тест 1

1. Какой оператор в СИ находит остаток от деления?

mod  
%  
«&&»

2. Описание строки s из 10 символов в СИ нужно выполнить:

string s(10);  
char s[10];  
char s[11];  
string s[10];

3. n в СИ обозначает:

горизонтальная табуляция  
перевод курсора на новую строку  
спецификатор целочисленного формата  
спецификатор вещественного формата

4. Оператор соотношения "равно" записывается:

=  
:=  
!=  
==

5. Функция strlen() ...

вычисляет длину строки  
выполняет сцепление строк  
проверяет вхождение одной строки в другую  
выполняет преобразование числа в строку

6. Как записывается логическое "ИЛИ" в СИ?

||  
OR  
|  
||

7. Как записывается логическое "НЕ" в СИ?

!  
not  
&&

8. Язык СИ разработал...

Том Куртц  
Никлаус Вирт  
Деннис Ритчи  
Билл Гейтс

9. Как находится модуль вещественного числа в СИ?

mod  
fabs  
abs  
||

10. Как записать на СИ "корень квадратный из x"?

$x^{0.5}$   
power(x,2)  
pow(x,0.5)  
pow(x,y,2)

11. Оператор форматированного вывода:

scanf()  
print()  
printf()  
write()

12. Оператор ввода:

scanf()  
read()  
printf()  
input()

13. Укажите правильно заданную команду для ввода вещественной переменной a.

printf("%f",a);  
scanf("%d",&a);  
scanf("%f",&a);  
scanf("%-f",a);

14. Укажите правильно заданную команду для вывода вещественной переменной a.

printf("%f",a);  
scanf("%d",&a);  
scanf("%f",&a);  
printf("%d",a);

15. Значение вещественной переменной нужно вывести следующим образом: выравнивание по левому краю, три знака после запятой. Выберите соответствующую команду.

```
printf("%f5.3",x);  
printf("%d3",x);  
printf("%f-8.3",x);  
printf("%-3f",x);
```

16. Служебное слово switch используется в конструкции...

условия  
множественного выбора  
цикла с параметром  
форматированного вывода

17. Выберите правильно записанное условие проверки числа x на четность на языке СИ:

```
if (x%2=0)  
if (x%2==2)  
if (x mod 2 =0)  
if (x%2==0
```

18. Переход на новую строку в языке Си осуществляется:

```
\t  
%d  
\n  
&&
```

19. Укажите количество ошибок при использовании условного оператора на языке Си:

20.

```
if (a==5)  
{ printf("Good");  
b=a; }  
нет ошибок
```

20. Укажите неправильный вариант для обозначения имен переменных:

A: VAZ  
B: \_sww  
C: 5a  
D: re4  
E: kl\*s

*C и E*  
*Только E*  
*Все неправильные*  
*Все правильные*

21. Каким словом обозначается «главная» функция в программе на языке Си:

*mail*  
*main*  
*#include*  
*return*

21. Укажите какое значение будет выведено на экран:

```
h=4.3767;  
j=0;  
printf(“%1.1f”,h);
```

4.0  
0.3767  
0.0  
4.4

22. Укажите значение элемента массива K[2] после завершения цикла:

```
int K[78];  
for(i=0;i<=34;i+=2)  
K[i]=i-10;
```

-8  
2  
34  
-10

23. Целый тип в языке Си обозначается:

*integer*  
*float*  
*int*  
*Такого типа не существует*

24. Укажите количество ошибок при использовании условного оператора на языке Си: if

```
(a<-3);  
{printf(“%d\n”,a);  
a++;}  
нет ошибок.
```

25. Укажите количество ошибок при использовании оператора цикла на языке Си: for(t=-

```
4,t<4;  
t=t+0.5{ printf(“%f\n”,t);  
}
```

26. Укажите неправильный вариант для обозначения имен переменных:

A:  
a45B:  
y\_22C:  
pycD:  
f% E: \_1  
B

C  
D  
C и D

28. Подключение какой библиотеки необходимо для использования математических функций:

*math.h*  
*stdio.h*  
*#include*  
*conio.h*

29. Укажите какое значение будет выведено на экран:

```
h=-6.278;  
j=6;  
printf(“%4.2f”,j);
```

6.00  
-6.28  
-6.00  
6.28

30. Укажите значение элемента массива K[13] после завершения цикла:

```
int K[13];  
for(i=0;i<13;i=i++)  
K[i]=i;
```

13.  
0.

Массив задан неправильно  
Элемент K[13] не определится

31. Укажите количество ошибок в строке:

```
print(“%d %d”,y,&y);  
Ошибок нет.
```

32. Для подключения модулей используется:

*#include*  
*stdio.h*  
*main*  
*getch()*

33. Тип переменной указывается:

*через : после переменной*  
*после переменной*  
*указывать не обязательно*  
*до ее написания*

34. Укажите количество ошибок при использовании условного оператора на языке Си:  
if (c<>3) {G=sin(c); printf(“%f\n”,G); c--;};  
Ошибок нет.

35. Укажите количество ошибок при использовании оператора цикла на языке Си:  
for (g=0,g<13,g++);  
{printf(“%f\n”,g); }

36. Подключение какой библиотеки необходимо для использования функции getch()?

*stdio.h*  
*stdlib.h*  
*getch.h*  
*conio.h*

37. Укажите значение элемента массива K[0] после завершения цикла:  
int K[4];  
for(i=3;i>0;i=i-1) K[i]=i\*i; K[0]=K[2];

38. Укажите количество ошибок при использовании условного оператора:

if e=12  
printf(“%d”,e);  
Нет ошибок.

39. Для использования функции strlen() необходимо подключить библиотеку:

*stdio.h*  
*conio.h*  
*string.h*  
*stdlib.h*

40. Укажите диапазон случайных чисел для random(12)-6:

*от 0 до 6*  
*от -6 до 0*  
*от -6 до 5*  
*от -6 до -1*

42. Укажите количество ошибок при использовании условного оператора на языке Си:  
while (d!=13) {G=cos(sqrt(d));  
printf(“%f\n”,G); d--;}  
Нет условного оператора.  
while (d!=13) {G=cos(sqrt(d)); printf(“%f\n”,G); d--;}  
42. Какой оператор используется в коде программы?

оператор условия  
оператор цикла с параметром  
оператор цикла с предусловием  
оператор цикла с постусловием



43. Укажите значение элемента массива K[4] после завершения цикла:

```
int K[100];
for(i=3;i<100;i++){ if(i==4) K[i]=10;
else K[i]=2*i; }
```

44. Каким символом завершается программа на языке Си:

«}»  
«getch»  
«.»  
«return»

45. Можно ли в операторе условия if не ставить скобки {}

*Всегда надо ставить*  
*Можно, если нет else*  
*Нельзя*  
*Можно, если блок состоит только из одного оператора*

46. Целочисленный массив обозначается:

*float A[4]*  
*int A[13]*  
*int A[1][3]*  
*2 или 3 вариант*

47. Укажите количество ошибок при использовании условного оператора на языке Си:

```
if (x<10) {printf(“%f\n”,x);
x++;
}else printf(“end”);
```

48. Какой модуль необходимо подключить для функции random?

*stdio.h*  
*stdlib.h*  
*conio.h*  
*randomize.h*

49. Какой из операторов не является оператором цикла?

*for*  
*while*  
*if*  
*2 и 3 варианты*

50. Двумерный массив обозначается:

*W[3][5]*  
*W[3;5]*  
*W[3,5]*  
*W[3],[5]*

51. В каком порядке следует подключать модули?

*В порядке следования необходимых функций  
В строго определённом  
Не имеет значения  
Их можно вообще не подключать*

52. a=5, b=7

После выполнения операции a+=b значение переменной a равно:

12.

5.

35.

операция не имеет смысла

```
int a,b; float c; printf("Vvedite a\n");
scanf("%d",&a);
printf("Vvedite b\n");
scanf("%d",&b);
c=a/b;
printf("c=%3.2f\n",c);
```

53. С клавиатуры вводятся 12 и 5. Какой результат выведется на экран?

c=0.05

c=2.00

c=2.40

c=3.20

54. В каком выражении использован оператор инкремента?

4\*a--

a\*=b

x\*y++

ceil(x)

55. Оператор, уменьшающий значение переменной на единицу - это ...

fabs

инкремента

floor

декремента

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Арктический государственный агротехнологический университет»  
Колледж технологий и управления  
Цикловая комиссия гуманитарных и естественных дисциплин

**Комплект  
контрольно-оценочных средств  
для промежуточной аттестации  
по результатам освоения дисциплины**

ОП.04 Основы алгоритмизации и программирования  
09.02.07 Информационные системы и программирование

## **ПЕРЕЧЕНЬ ТЕОРЕТИЧЕСКИХ ВОПРОСОВ К ЭКЗАМЕНУ**

1. Алгоритм. Базовые алгоритмические конструкции.
2. Исполнители алгоритма. СКИ. Пошаговое выполнение алгоритма.
3. История и классификация языков программирования.
4. Pascal. Алфавит. Идентификаторы, переменные, константы. Общая структура программы. Комментарии.
5. Pascal. Стандартные типы данных. Арифметические и логические выражения. Стандартные функции.
6. Pascal. Простые операторы. Операторы присваивания, составной и простой операторы, операторы ввода-вывода.
7. Pascal. Условный оператор.
8. Pascal. Оператор цикла с предусловием.
9. Pascal. Оператор цикла с постусловием.
10. Pascal. Оператор цикла с параметром.
11. Pascal. Оператор множественного выбора.
12. Pascal. Строки. Действия над строками.
13. Pascal. Множества. Операции над множествами.
14. Pascal. Одномерные массивы.
15. Pascal. Двумерные массивы. Свойства квадратных матриц.
16. Pascal. Записи. Оператор присоединения.
17. Pascal. Файлы. Функции для работы с файлами.
18. Pascal. Процедуры и функции.
19. Pascal. Работа с графикой.
20. Pascal. Анимация.
21. C. Основные типы данных. Стандартные функции. Математические функции. Структура программы.
22. C. Операторы форматного ввода и вывода. Спецификаторы и модификаторы формата.
23. C. Оператор присваивания. Арифметические выражения. Операторы инкремента и декремента. Сокращенный вид записи арифметических выражений.
24. C. Условные выражения. Оператор условия.
25. C. Оператор множественного выбора.
26. C. Циклы с предусловием.
27. C. Циклы с постусловием.
28. C. Циклы с параметром.
29. C. Массивы.
30. C. Строки.
31. C. Функции.
32. C. Структуры.
33. C. Файлы.

## ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ВОПРОСОВ К ЭКЗАМЕНУ

1) Составить программу для вычисления значения выражения:

$$y = \cos^2 2x + \left| \sin \frac{x}{2} \right|$$

Входные данные –  $x$  (вводится с клавиатуры). Выходные данные –  $y$ .

2) Составьте блок-схему алгоритма и программу, определяющую является ли введенное число двузначным. Результатом работы программы должно быть сообщение 'является' или 'не является'.

3) Составьте блок-схему и программу вычисления значения функции, используя алгоритм полного ветвления.

4) Составьте блок-схему алгоритма и программу вычисления площади и периметра прямоугольника, если известны его стороны (вводятся с клавиатуры).

5) Составьте блок-схему алгоритма и программу вычисления площади и периметра треугольника, если известны его стороны и высота (вводятся с клавиатуры).

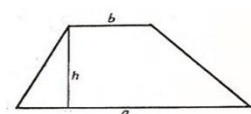
6) Составьте блок-схему алгоритма и программу, определяющую является ли введенное целое число положительным. Результатом работы программы должно быть сообщение 'является' или 'не является'.

7) Составьте блок-схему алгоритма и программу, определяющую является ли введенное целое число нечетным. Результатом работы программы должно быть сообщение 'является' или 'не является'.

8) Составьте блок-схему алгоритма и программу вычисления количества букв в строке. Строка вводится с клавиатуры.

9) Составьте блок-схему алгоритма и программу вычисления площади ромба по его диагоналям (вводятся с клавиатуры).

$S = \frac{1}{2} d_1 \cdot d_2$  10) Составьте блок-схему алгоритма и программу вычисления площади трапеции по ее основаниям и высоте (вводятся с клавиатуры).



$$S = \frac{a + b}{2} h$$

11) Составить блок-схему алгоритма и программу для вычисления значения выражения:

Входные данные –  $c, b$  (вводятся с клавиатуры). Выходные данные –  $a$ .

$$a) y = \frac{5x^2 - 4}{7x + 2}; \quad 6) a = \sqrt{\frac{b+c}{3bc} - \frac{b^2}{2c}};$$

$$e) y = \sin^2 \frac{x}{2} \cdot |\cos 2x|; \quad d) x^2 + y^2 \geq (z + 2)^2;$$

$$a) * a_1 = \sqrt{\frac{bx^2 - 1}{2xy} - \frac{(x+y)^2}{2xy}}$$

12) Составьте блок-схему алгоритма и программу, которая по введенному с клавиатуры целому числу в диапазоне 0 – 9 выводит строку — название соответствующей цифры на русском языке (0 — "ноль", 1 — "один", 2 — "два", ...).

13) Составьте блок-схему алгоритма и программу вычисления среднего арифметического трех целых чисел, введенных с клавиатуры, с точностью до трех знаков после запятой.

14) Составить блок-схему алгоритма и программу для вычисления значения выражения:

Входные данные –  $x, y$  (вводятся с клавиатуры). Выходные данные –  $s$ .

$$s = \frac{|x + \sqrt{y}|}{|x^2 + y^2|}$$

- 15) Составить блок-схему алгоритма и программу, которая по введенному номеру дня недели (вводится с клавиатуры), выдает его название.
- 16) Составьте блок-схему алгоритма и программу вычисления площади круга и длины окружности по введенному радиусу (вводится с клавиатуры).
- 17) Составьте блок-схему алгоритма и программу вычисления площади и периметра квадрата, если известны его стороны (вводятся с клавиатуры).
- 18) Составьте блок-схему алгоритма и программу вывода на экран всех чисел, которые кратны числу 5, в интервале от  $a$  до  $b$  (значения  $a$  и  $b$  вводятся,  $a$
- 19) Составьте блок-схему алгоритма и программу вычисления произведения целых чисел из промежутка  $[-6; 5)$ .
- 20) Составьте блок-схему алгоритма и программу вывода на экран всех трехзначных чисел, кратных трем. Решите данную задачу, используя цикл с параметром, цикл с предусловием и цикл с постусловием.
- 21) Составьте блок-схему алгоритма и программу расположения трех чисел в порядке возрастания.
- 22) Составьте блок-схему алгоритма и программу вывода на экран таблицы степеней  $2^n$ , где  $0 \leq n \leq 10$ .
- 23) Составьте блок-схему алгоритма и программу вывода таблицы значений функции  $y = 4x^2 + 5x - 10$  на отрезке  $[-9; 9]$  с шагом  $h = 3$ . Вывод результатов оформите в виде таблицы с границами.
- 24) Составьте блок-схему алгоритма и программу вычисления среднего арифметического целых чисел из отрезка  $[-4; 15]$  с точностью до трех знаков после запятой.
- 25) Составьте блок-схему алгоритма и программу вывода таблицы значений функции  $y = x^2 - 5x - 2$  на отрезке  $[1; 20]$  с шагом  $h = 2$ . Вывод результатов оформите в виде таблицы с границами.
- 26) Составьте блок-схему алгоритма и программу вычисления значения выражения для данного натурального числа  $N$ :  $1/1^2 + 1/2^2 + 1/3^2 + \dots + 1/N^2$ . Значение  $N$  вводится с клавиатуры.
- 27) Составьте блок-схему алгоритма и программу заполнения массива, не используя клавиатуру, числами: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20. Найти сумму элементов полученного массива.
- 28) Составьте блок-схему алгоритма и программу, определяющую, на какую букву начинается второе слово в строке, введенной с клавиатуры.
- 29) Составить программу и блок-схему алгоритма. Сгенерировать массив  $A$  из 10 целых чисел, которые берутся из промежутка  $[-15, 15]$ . Подсчитать количество нечетных элементов массива.
- 30) Составить программу и блок-схему алгоритма. Сгенерировать массив  $C$  из 15 целых чисел, которые берутся из промежутка  $[-10, 10]$ . Подсчитать сумму четных элементов массива.

Экзамен, завершающий изучение учебной дисциплины, – это форма промежуточного контроля, целью которой является оценка теоретических знаний и практических навыков, способности студента к мышлению, приобретение навыков самостоятельной работы, умение синтезировать полученные знания и применять их при решении практических. При проведении промежуточной аттестации в форме экзамена / комплексного экзамена уровень освоения оценивается оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Промежуточная аттестация в форме экзамена / комплексного экзамена проводится в дни, освобожденные от других форм учебной нагрузки, по отдельному расписанию за счет времени, отведенного учебным планом на промежуточную аттестацию.

### **Условия проведения экзамена**

К экзамену допускается студент прошедший обучение по дисциплине и освоивший текущий контроль не менее чем на 70% от общего количества.

Экзамен проводится согласно расписанию экзаменационной сессии, аудиторно. Билет с теоретической и практической частями предоставляется студенту в день экзамена. На подготовку по теоретическому вопросу студенту отводится не более 20 минут выполнение практической части студенту отводится не более 30 минут. На ответ по билету отводится не более 10 минут.

### **Инструкция для студентов**

#### **1. Последовательность и условия выполнения заданий**

*В день экзамена, согласно билету необходимо выполнить практическое задание и ответить на теоретический вопрос. Защита практического задания проводится студентом непосредственно после завершения работы. Преподаватель вправе задавать уточняющие или дополнительные вопросы в количестве не более трех, после чего должна быть выставлена оценка.*

#### **2. Вы можете воспользоваться нет**

#### **3. Максимальное время выполнения заданий: 60 минут**

#### **4. Перечень раздаточных и дополнительных материалов экзаменационные билеты**

### **Критерии оценивания:**

*Оценки "отлично" заслуживает студент, обнаруживший всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной программой. Как правило, оценка "отлично" выставляется студентам, усвоившим взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.*

*Оценки "хорошо" заслуживает студент обнаруживший полное знание учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка "хорошо" выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.*

*Оценки "удовлетворительно" заслуживает студент, обнаруживший знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по специальности, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка "удовлетворительно" выставляется студентам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.*

*Оценка "неудовлетворительно" выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий. Как правило, оценка "неудовлетворительно" ставится*

студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине.