

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО Уральский государственный лесотехнический университет
Уральский лесотехнический колледж

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
ОП. 06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ
специальность
09.02.11 РАЗРАБОТКА И УПРАВЛЕНИЕ ПРОГРАММНЫМ
ОБЕСПЕЧЕНИЕМ
Направленность: РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ

Составитель: преподаватель

А.В. Токарь

Рабочая программа рекомендована к использованию в учебном процессе
методическим советом Уральского лесотехнического колледжа
(протокол № 1 от 30 августа 2025 года)

Председатель методического совета

(подпись)

М.В. Чапаева

г. Екатеринбург, 2026

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	3
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	5
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ	12
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	13
ПРИЛОЖЕНИЕ	
Фонд оценочных средств для промежуточной аттестации	15

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

1.1. Место дисциплины в структуре основной профессиональной образовательной программы

Учебная дисциплина «Основы алгоритмизации и программирования» принадлежит к общепрофессиональному циклу.

1.2. Цель и планируемые результаты освоения дисциплины:

Код ПК, ОК	Умения	Знания
ПК 2.4.	Разрабатывать алгоритмы для конкретных задач. Использовать программы для графического отображения алгоритмов. Определять сложность работы алгоритмов. Работать в среде программирования. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. Оформлять код программы в соответствии со стандартом кодирования. Выполнять проверку, отладку кода программы. .	Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. Эволюцию языков программирования, их классификацию, понятие системы программирования. Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. Подпрограммы, составление библиотек подпрограмм 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
ЛР14 ЛР15 ЛР18 ЛР20	Демонстрирующий навыки анализа и интерпретации информации из различных источников с учетом нормативно-правовых норм Демонстрирующий готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности. Выполняющий трудовые функции и демонстрирующий профессиональные навыки в профессиональной деятельности.	

	Владеющий современными методами обслуживания потребителя на основе уважения, толерантности, готовности к диалогу	
--	--	--

2. СТРУКТУРА УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем в часах
Объем образовательной программы	176
в том числе:	
теоретическое обучение	54
практические занятия	98
Самостоятельная работа ¹	12
Промежуточная аттестация	12

¹Самостоятельная работа в рамках образовательной программы планируется образовательной организацией в соответствии с требованиями ФГОС СПО в пределах объема учебной дисциплины в количестве часов, необходимом для выполнения заданий самостоятельной работы обучающихся, предусмотренных тематическим планом и содержанием учебной дисциплины.

1.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающегося, курсовая работа (проект)	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Раздел 1.	Введение в программирование		ПК 2.4
Тема 1.1. Языки программирования	Содержание учебного материала Лекции	8	
	1. Развитие языков программирования.	2	
	2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.	2	
	3. Жизненный цикл программы. Программа. Программный продукт и его характеристики.	2	
	4. Основные этапы решения задач на компьютере.	2	
	Практические занятия	10	
	Самостоятельная работа обучающихся	1	
Тема 1.2. Типы данных	Содержание учебного материала Лекции	4	
	1. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.	4	
	Практические занятия	10	
	Самостоятельная работа обучающихся	1	
Раздел 2.	Содержание учебного материала Лекции	6	ПК 2.4
Тема 2.1. Операторы языка программирования	1. Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.	1	
	2. Условный оператор. Оператор выбора.	1	
	3. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные	1	

	циклы.		
	4. Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.	1	
	5. Структурированный тип данных – множество. Операции над множествами.	1	
	6. Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа	1	
	Практические занятия	5	
	Самостоятельная работа обучающихся	1	
Раздел 3.	Содержание учебного материала Лекции	10	ПК 2.4.
Тема 3.1. Процедуры и функции	1. Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.	5	
	2. Рекурсия. Программирование рекурсивных алгоритмов.	5	
	Практические занятия	10	
	Самостоятельная работа обучающихся	1	
Тема 3.2. Структуризация в программировании	Содержание учебного материала Лекции	4	
	1. Основы структурного программирования. Методы структурного программирования.	4	
	Практические занятия	5	
	Самостоятельная работа обучающихся	1	
Тема 3.3. Модульное программирование	Содержание учебного материала Лекции	2	
	1. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.	1	
	2. Стандартные модули.	1	
	Практические занятия	10	
	Самостоятельная работа обучающихся	1	
Раздел 4	Основные конструкции языков программирования		ПК 2.4.
Тема 4.1 Указатели.	Содержание учебного материала Лекции	3	

	1. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.	1	
	2. Структуры данных на основе указателей.	1	
	3. Задача о стеке.	1	
	Практические занятия	10	
	Самостоятельная работа обучающихся	1	
Раздел 5	Содержание учебного материала Лекции	4	ПК 2.4
Тема 5.1 Основные принципы объектно-ориентированного программирования (ОПОП)	1. История развития ОПОП. Базовые понятия ОПОП: объект, его свойства и методы, класс, интерфейс.	1	
	2. Основные принципы ОПОП: инкапсуляция, наследование, полиморфизм.	1	
	3. Классы объектов. Компоненты и их свойства.	1	
	4. Событийно-управляемая модель программирования. Компонентно-ориентированный подход.	1	
	Практические занятия	10	
	Самостоятельная работа обучающихся	1	
Тема 5.2 Интегрированная среда разработчика.	Содержание учебного материала Лекции	6	
	1. Требования к аппаратным и программным средствам интегрированной среды разработчика.	1	
	2. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.	1	
	3. Панель компонентов и их свойства. Окно кода проекта.	1	
	4. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.	1	
	5. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.	1	
	6. Настройка среды и параметров проекта.	1	
	Практические занятия	10	
	Самостоятельная работа обучающихся	2	
Тема 5.3. Визуальное	Содержание учебного материала Лекции	4	

событийно-управляемое программирование	1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.	1	
	2. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.	1	
	3. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.	2	
	Практические занятия	5	
	Самостоятельная работа обучающихся	2	
Тема 5.4 Разработка оконного приложения	Содержание учебного материала Лекции	2	
	1. Разработка функционального интерфейса приложения. Создание интерфейса приложения.		
	2. Разработка функциональной схемы работы приложения.		
	3. Разработка игрового приложения.		
	Практические занятия	5	
Тема 5.5 Этапы разработки приложений	Самостоятельная работа обучающихся	1	
	Содержание учебного материала Лекции	1	
	1. Разработка приложения.		
	2. Проектирование объектно-ориентированного приложения.		
	3. Создание интерфейса пользователя.		
	4. Тестирование, отладка приложения.		
	Практические занятия	3	
	Самостоятельная работа	1	
	Тематика практических занятий и лабораторных работ: Знакомство со средой программирования. Составление программ линейной структуры. Составление программ разветвляющейся структуры. Составление программ циклической структуры Обработка одномерных массивов. Обработка двумерных массивов.		

Работа со строками. Работа с данными типа множество. Файлы последовательного доступа. Типизированные файлы. Нетипизированные файлы. Организация процедур. Организация функций. Применение рекурсивных функций. Программирование модуля. Создание библиотеки подпрограмм. Использование указателей для организации связанных списков. Изучение интегрированной среды разработчика. Создание проекта с использованием компонентов для работы с текстом. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Создание проекта с использованием кнопочных компонентов. Создание проекта с использованием компонентов стандартных диалогов и системы меню. Разработка функциональной схемы работы приложения. Разработка оконного приложения с несколькими формами. Разработка игрового приложения. Создание процедур обработки событий. Компиляция и запуск приложения. Разработка интерфейса приложения. Тестирование, отладка приложения. Классы ОПОП: виды, назначение, свойства, методы, события. Объявления класса. Создание наследованного класса. Программирование приложений. Перегрузка методов.		
Промежуточная аттестация		
Всего:		

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.04. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

3.1. Материально-техническое обеспечение

Реализация программы предполагает наличие лаборатории программирования и баз данных.

Лаборатория программирования и баз данных – это учебная аудитория для проведения лекционных, практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, имеющая следующее оснащение: столы и стулья с посадочными местами на количество обучающихся, рабочее место преподавателя, персональные компьютеры с возможностью подключения к сети "Интернет"., Интерактивная доска Smart Board480i со встроенным проектором SMART V25.

В качестве помещений для самостоятельной работы обучающихся используется:

- кабинет информатики, имеющее следующее оснащение: столы и стулья с посадочными местами на количество обучающихся, рабочее место преподавателя, персональные компьютеры с возможностью подключения к сети "Интернет"., интерактивная доска, проектор, экран проекционный.

- читальный зал с посадочными местами на количество обучающихся, автоматизированные рабочие места для читателей с выходом в сеть Интернет и электронную информационную образовательную среду УГЛТУ, программное обеспечение общего назначения. Технология беспроводной локальной сети Wi-Fi.

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации имеет печатные и электронные образовательные, а также информационные ресурсы, рекомендуемые для использования в образовательном процессе

3.2.1. Основные печатные издания

1. Семакин И.Г., Шестаков А.П. Основы алгоритмизации и программирования. –М,: ОИЦ «Академия», 2019

3.2.2. Основные электронные издания

1. Андрианова, А. А. Алгоритмизация и программирование. Практикум : учебное пособие для спо / А. А. Андрианова, Л. Н. Исмагилов, Т. М. Мухтарова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 240 с. — ISBN 978-5-8114-8948-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/186390> — Режим доступа: для авториз. пользователей.

2. Основы алгоритмизации и программирования. Деревья : методические указания / составители И. Ф. Астахова [и др.]. — Воронеж : ВГУ, 2015. — 58 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/356639> — Режим доступа: для авториз. пользователей.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.04. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

Результаты обучения	Критерии оценки	Формы и методы оценки
Перечень умений, осваиваемых в рамках дисциплины: • Разрабатывать алгоритмы для конкретных задач. • Использовать программы для графического отображения алгоритмов. • Определять сложность работы алгоритмов. • Работать в среде программирования. • Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. • Оформлять код программы в соответствии со стандартом кодирования. • Выполнять проверку, отладку кода программы.	«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко. «Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.	Примеры форм и методов контроля и оценки • Компьютерное тестирование на знание терминологии по теме; • Тестирование. • Контрольная работа. • Самостоятельная работа. • Оценка выполнения практического задания(работы) • Решение ситуационной задачи.
Перечень знаний, осваиваемых в рамках дисциплины: • Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. • Эволюцию языков программирования, их классификацию, понятие системы программирования. • Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. • Подпрограммы, составление библиотек подпрограмм • 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.	«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки. «Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.	

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения промежуточной аттестации

ОП.06 ОСНОВЫ АЛГОРИТМИЗАЦИИ ПРОГРАММИРОВАНИЯ

для студентов специальности

09.02.11 РАЗРАБОТКА И УПРАВЛЕНИЕ ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ

Направленность: **РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ**

1. ПАСПОРТ ФОС ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Комплект оценочных средств предназначен для проверки результатов освоения учебной дисциплины «Основы алгоритмизации и программирования» и входит в состав фонда оценочных средств программы подготовки специалистов среднего звена (далее - ППССЗ) по специальности 09.02.11 Разработка и управление программным обеспечением, направленность Разработка информационных систем.

Настоящий комплект оценочных средств предназначен для проведения аттестационных испытаний по учебной дисциплине в форме устного экзамена с выполнением практических заданий.

Экзамен проводится по билетам, содержащим два теоретических вопроса и одно практическое задание. На подготовку ответа по билету каждому студенту отводится не более 45 минут.

Полный комплект оценочных средств включает 75 теоретических вопроса и 41 практических задания, направленных на проверку сформированности всей совокупности образовательных результатов, заявленных во ФГОС СПО и рабочей программе «Основы алгоритмизации и программирования».

Используемые термины и определения, сокращения

УД	–	учебная дисциплина;
МДК	–	междисциплинарный курс;
ППССЗ	–	программы подготовки специалистов среднего звена;
КОС	–	контрольно-оценочные средства;
ФГОС СПО	–	Федеральный государственный образовательный стандарт среднего профессионального образования;
ОК	–	общие компетенции;
ПК	–	профессиональные компетенции

1. Общие положения

В результате освоения учебной дисциплины ОП.06. «Основы алгоритмизации и программирования» обучающийся **должен обладать** предусмотренными ФГОС СПО по специальности Разработка и управление программным обеспечением следующими умениями и знаниями:

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;
Зн 1	понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
Зн 2	эволюцию языков программирования, их классификацию, понятие системы программирования;
Зн 3	основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
Зн 4	подпрограммы, составление библиотек подпрограмм;
Зн 5	объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.

Знания и умения, формируемые в рамках УД Основы алгоритмизации и программирования, направлены на формирование общих и профессиональных компетенций:

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

**Матрица соответствия оценочных материалов образовательным результатам
УД**

Образов а тел. результата	Формулировка умения/знания	Код ОК, ПК	№ во- проса	№ прак- тически- го зада- ния
У 1	разрабатывать алгоритмы для конкретных задач;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4.		1-41
У 2	использовать программы для графического отображения алгоритмов;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4.		1-41
У 3	определять сложность работы алгоритмов;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4.		1-41
У 4	работать в среде программирования;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4.		1-41
У 5	реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4.		1-41
У 6	оформлять код программы в соответствии со стандартом кодирования;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4., ПК 1.6, ПК 2.5		1-41
У 7	выполнять проверку, отладку кода программы;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4., ПК 2.4.		1-41
З 1	понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4.	1-17	
З 2	эволюцию языков программирования, их классификацию, понятие системы программирования;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4.	1, 5, 8, 56,57	
З 3	основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4.	18-34	
З 4	подпрограммы, составление библиотек подпрограмм;	ОК 1, ОК 2, ОК 4, ОК 9, ПК 1.1. - ПК 1.4., ПК 1.6	35-37	
З 5	объектно-	ОК 1, ОК 2, ОК 4, ОК 9,	41-75	

	ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.	ПК 1.1. - ПК 1.4.		
--	--	--------------------------	--	--

2. СТРУКТУРА И ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ЭКЗАМЕНУ

Раздел 1 Введение в программирование

Теоретические вопросы:

1. Алгоритмы. Правила записи схем алгоритмов.
Определить значение переменной s после выполнения следующих операторов: $\text{int } s=0;$
 $\text{int } n=4;$ for (int $i=2$; $i < n$; $i++$) $s+=10/i$;
2. Структура программы на языке C#.
Чему равны значения переменных a и b после выполнения последовательности действий: $a = 15 / (16 \% 7); b = 34 / a * 5 - 29 \% 5 * 2$
3. Арифметические операции, приоритет операций. Выражения. Примеры.
Чему равны значения переменных a и b после выполнения последовательности действий: $a = 4 * 5 / 3 \% 2; b = 4 * 5 / (3 \% 2)$
4. Типы данных (Integer, Double, String, массивы, файлы) – назначение, описание.
Чему равны значения переменных a и b после выполнения последовательности действий: $a = 15 / (16 \% 7); b = 34 / a * 5 - 29 \% 5 * 2;$
5. *Трассировка программы, использование точек останова, просмотр и изменение значений элементов данных.
6. Строковый тип данных. Основные функции работы со строками.
Определите значение переменной p после выполнения следующего фрагмента программы: $\text{int } m = 13; \text{int } n = 21; n = 2 * m - n; \text{if } (m \leq n) p = m + n; \text{else } p = 4 - m * n;$
7. Операторы ввода/вывода. Форматы вывода данных.
Чему равны значения переменных a и b после выполнения последовательности действий: $\text{int } a = 4 * 5 / 3 \% 2; \text{int } b = 4 * 5 / (3 \% 2);$
8. *Компиляция и отладка программы: Использование точек останова. Просмотр и изменение значений элементов данных.
9. Арифметические функции и процедуры. Пример.
Чему равны значения переменных a и b после выполнения последовательности действий: $\text{int } a = 15 / (16 \% 7); \text{int } b = 34 \% a * 5 - 29 \% 5 * 2;$
10. *Локальная обработка исключительных ситуаций. Объявление и вызов.
11. Основные символы языка C#. Примеры записи комментариев.
Чему равны значения переменных a и b после выполнения последовательности действий: $\text{int } a = 4 * 5 / 3 \% 2; \text{int } b = 4 * 5 / (3 \% 2);$
12. Алгоритм нахождения количества и суммы натуральных чисел.
Определить значение переменной s после выполнения следующих операторов: $\text{int } s=0;$
 $\text{int } n=5;$ for (int $i=2$; $i \leq n$; $i++$) $s+=100 / i$;
13. Переменные (назначение, описание в программе).
Чему равны значения переменных a и b после выполнения последовательности действий: $\text{int } a = 15 / (16 \% 7); \text{int } b = 34 / a * 5 - 29 \% 5 * 2;$
14. *Компиляция и отладка программы: Типы сообщений компилятора.
Чему равны значения переменных a и b после выполнения последовательности действий: $\text{int } a = 4 * 5 / 3 \% 2; b = 4 * 5 / (3 \% 2);$
15. Типы данных.
Чему равны значения переменных a и b после выполнения последовательности действий: $\text{int } a = 15 / (16 \% 7); \text{int } b = 34 \% a * 5 - 29 \% 5 * 2;$
16. Алгоритм обмена местами элементов массива.
Чему равны значения переменных a и b после выполнения последовательности действий: $\text{int } a = 4 * 5 / 3 \% 2; \text{int } b = 4 * 5 / (3 \% 2);$

17. Имена переменных (правила задания, примеры).

Чему равны значения переменных *a* и *b* после выполнения последовательности действий: `int a= 15 /(16 % 7); int b=34 % a*5 – 29 % 5*2;`

Раздел 2 Базовые конструкции языка

программирования Теоретические вопросы:

18. Оператор цикла с постусловием Do While.

Какими будут значения переменных *s* и *i* после выполнения следующих операторов: `int s=20; int i= 0; while (i<=10) {s+= 5; i+= 1;}`

19. Оператор цикла с предусловием While.

Определить значения переменных *a* и *b* после выполнения следующих операторов: `a= 1; b= 2; while (b<20) {a= a + 2; b= b +2;}`

20. Оператор цикла с параметром For.

Определить значение переменной *s* после выполнения следующих операторов: `int s=0; int n=5; for (int i=2; i< n; i++) s+= 100 / i;`

Определить значение переменной *s* после выполнения следующих операторов: `int s=0; int n=5; for (int i=2; i<= n; i++) s+= 100 / i;`

21. Оператор выбора switch.

Определить значение переменной *p* после выполнения следующего фрагмента программы: `int m= 13; int n= 21; n= 2*m – n;`

`if (m<=n) p = m + n ; else p= 4 – m*n;`

22. Операторы switch и If, назначение и отличие.

Определить значение переменной *p* после выполнения следующего фрагмента программы: `int m = 13; int n= 21; n= 2*m – n;`

`if (m<=n) p= m + n; else p= 4 – m*n;`

23. Операторы цикла For , While и Do While, блок-схемы.

Какими будут значения переменных *s* после выполнения следующих операторов: `int s=2; int i= 1; while (i<10) {s+= i; i+= 1;}`

24. Условный оператор IF (IF-....., IF-.....-ELSE).

Определить значение переменной *p* после выполнения следующего фрагмента программы: `int m= 13; int n= 21; n= 2*m – n;`

`if (m<=n ) p= m + n; else p= 4 – m*n;`

25. Операторы цикла While и Do While, их отличие.

Какими будут значения переменных *a* и *b* после выполнения следующих операторов: `int a= 1; int b= 2; while (b<20 ) { a+= 2; b+= 2;}`

26. Оператор многовероятностного выбора. Форма записи и пример.

Определить значение переменной *p* после выполнения следующего фрагмента программы: `int m= 13; int n= 21; n= 2*m – n;`

`if (m<=n) p= m + n; else p= 4 – m*n`

27. Ввод одномерных массивов случайным образом. Пример.

Определить значение переменной *s* после выполнения следующих операторов: `int s=0; int n=5; for (int i=2 ; i< n; i++) s+= 100 / i;`

28. Алгоритм сортировки массива.

Какими будут значения переменных *s* и *i* после выполнения следующих операторов: `int s=20; int i= 0; while (i<=10) { s+= 5; i+= 1;}`

29. Ввод двумерных массивов.

Какими будут значения переменных *a* и *b* после выполнения следующих операторов:

`int a= 1; int b= 2; while (b<20) { a+= 2; b+=2;}`

30. Вложенные условные операторы. Пример алгоритма.

Определить значение переменной p после выполнения следующего фрагмента программы: `int m= 13; int n= 21; n= 2*m – n; if (m<=n ) p= m + n; else p= 4 – m*n;`

31. Вывод двумерных массивов.

Определить значение переменной s после выполнения следующих операторов: `int s =0; int n =4; for (int i=2; i<= n; i++) s+= 10 / i;`

32. Алгоритм нахождения максимального (минимального) элемента в массиве (одномерный массив, матрица).

33. Алгоритм нахождения порядкового номера максимального (минимального) элемента в массиве (одномерный массив, матрица).

34. Функции деления с остатком. Определить значения переменных a и b после выполнения последовательности действий:

`a= 4*5 / 3 %2; b= 4*5 /(3 %2);`

Практические задания

1. Разработать и произвести отладку программы: Найти сумму бесконечного ряда. Суммировать до тех пор, пока сумма не станет больше заданного $p > 0$. Вывести эти числа.

2. Разработать и произвести отладку программы для определения $N! - M!$. $N! = 1*2*3*4*... *n$

3. Разработать и произвести отладку программы: Вычислить сумму квадратов всех целых чисел, пока сумма квадратов меньше заданного числа A . Вывести эти числа.

4. Разработать и произвести отладку программы: Произведение первых четных чисел равно P , сколько сомножителей взято.

5. Разработать и произвести отладку программы: Определить все двузначные числа, сумма квадратов цифр которых кратны числу 15.

6. Разработать и произвести отладку программы: Даны два одномерных массива одинаковой длины. Получить третий массив такой же размерности, каждый элемент которого равен сумме соответствующих элементов данных массивов.

7. Разработать и произвести отладку программы: ан одномерный массив чисел. Определите сумму элементов, принадлежащих промежутку от A до B (A и B водить с клавиатуры).

8. Разработать и произвести отладку программы определения количества элементов массива, больших среднего арифметического всех его элементов.

9. Разработать и произвести отладку программы: Дан массив P целых чисел из n элементов, заполненный случайным образом числами из промежутка $[-10,10]$. Из элементов массива P сформировать массив M той же размерности по правилу: если номер четный, то $M_i = i * P_i$, если нечетный, то $M_i = -P_i$. Исходный и скорректированный массив вывести на экран.

10. Разработать и произвести отладку программы: ан массив P целых чисел из n элементов, заполненный случайным образом числами из промежутка $[-30,30]$. Из элементов массива P сформировать массив M из четных чисел. Исходный и скорректированный массивы вывести на экран.

11. Разработать и произвести отладку программы: ан массив P целых чисел из n элементов, заполненный случайным образом числами из промежутка $[-10,10]$. Из элементов массива P сформировать массив M той же размерности по возрастанию. Исходный и скорректированный массивы вывести на экран.

12. Разработать и произвести отладку программы, печатающей все делители целого числа в порядке убывания.

13. Разработать и произвести отладку программы, печатающей все делители целого числа в порядке возрастания.

Раздел 3 Структурное и модульное программирование

Теоретические вопросы:

35. Назначение процедуры. Описание и вызов процедуры.
Какими будут значения переменных s и i после выполнения следующих операторов: `int s=20; int i= 0; while (i<=10) { s+= 5; i+= 1;}`
36. Назначение функции. Описание и вызов функции. Отличие процедуры от функции. Чему равны значения переменных a и b после выполнения последовательности действий: `int a= 4*5 /3 % 2; b= 4*5/ (3 %2)`
37. Подпрограммы (структура подпрограммы, схема алгоритма).
Чему равны значения переменных a и b после выполнения последовательности действий: `int a= 4*5 / 3 % 2; int b= 4*5 /(3 %2);`

Раздел 4 Основные конструкции языков программирования

Теоретические вопросы:

38. Файлы (общие сведения, характеристики, описание файловых переменных).
Определить значение переменной s после выполнения следующих операторов: `int s=0; int n=5; for (int i=2 ; i< n; i++) s+= 100 / i;`
39. Основные процедуры работы с файлами.
Какими будут значения переменных a и b после выполнения следующих операторов: `int a= 1; int b= 2; while (b<20) { a+= 2; b+= 2;}`
40. Процедуры и функции работы с текстовыми файлами.
Какими будут значения переменных a и b после выполнения следующих операторов: `int a= 1; int b= 1; while (a<=3) { a+= 1; b+= 1;}`
41. Основные команды работы с графикой.
Чему равны значения переменных a и b после выполнения последовательности действий: `int a= 15 / (16 % 7); int b=34 % a*5 – 29 % 5*2;`
42. Текстовые файлы (назначение, описание в программе).
Какими будут значения переменных s и i после выполнения следующих операторов: `int s=20; int i= 0; while (i<=10) { s+= 5; i+= 1;}`

Практические задания:

14. Разработать и произвести отладку программы: Пользователь угадывает число задуманное компьютером, при помощи подсказок больше или меньше, компьютер выдаёт количество шагов, за которые пользователь угадал число.
15. Разработать и произвести отладку программы: Произведение N первых нечетных чисел равно p . Сколько сомножителей взято?
16. Дана последовательность из N целых чисел. Определить произведение максимального и минимального элементов этой последовательности.
17. Составьте программу- "перевертыш" (так называются слова, читающиеся одинаково слева направо и справа налево, например: ПОТОП, КАЗАК).
18. Дан текст со скобками, удалить текст в скобках вместе со скобками и вывести его отдельно.
19. Вывести все двухзначные числа, сумма цифр которых равна N .
20. В тексте имеется одна точка с запятой «;». Подсчитать количество символов до точки с запятой и после нее.

21. В тексте имеются запятые. Подсчитать количество запятых.
22. Вывести все четырехзначные числа, у которых сумма первых двух равна сумме двух последних.
23. В тексте имеются «;». Подсчитать их количество.
24. Вывести все шестизначные числа, у которых сумма первых трех равна сумме трех последних.
25. В тексте имеются скобки. Подсчитать количество скобок открывающихся и закрывающихся.
26. Вывести все четырехзначные числа, сумма цифр которых равна N.
27. Поменять порядок следования цифр в натуральном числе N на обратный.

Раздел 5. Объектно-ориентированная модель программирования

Теоретические вопросы:

41. *Интегрированная среда разработки Visual Studio.
42. Принципы объектно-ориентированного программирования: инкапсуляция.
43. Принципы объектно-ориентированного программирования: Понятие класса и объекта.
44. Назначение и свойства компонента CheckBox.
45. *Организация проекта. Структура проекта: файл проекта, файлы настроек проекта.
46. Обработка исключительных ситуаций. Классификация
47. Принципы объектно-ориентированного программирования: наследование.
48. *Организация проекта. Интерфейс приложения: форма как главный элемент приложения, основные свойства формы.
49. Структура компонентов: Свойства. Методы.
50. *Компиляция и отладка программы: Типы сообщений компилятора. Использование встроенного отладчика. Трассировка программы.
51. *Управление проектом: создание, сохранение, открытие и настройка проектов.
52. Принципы объектно-ориентированного программирования: полиморфизм.
53. Принципы объектно-ориентированного программирования: Методы. Вызов методов.
54. Свойства и обработчик событий.
55. Принципы объектно-ориентированного программирования: Свойства. Описание свойства.
56. *Организация проекта. Управление проектом: создание, сохранение и открытие, настройки.
57. *Организация проекта. Компиляция и отладка программы: типы сообщений компилятора, использование встроенного отладчика.
58. Управляющие элементы: списки, кнопки, меню, переключатели выбора, таблицы, панель инструментов.
59. Интерфейс окна Visual Studio.
60. *Интерфейс приложения: Форма как главный элемент приложения. Основные свойства формы. Размещение компонентов на форме.
61. *Структура проекта: файл ресурсов, файла исходного модуля и формы, файлы создаваемые пользователем.
62. *Структура проекта: Файл проекта. Файлы настроек проекта. Файл ресурсов.
63. Назначение команды меню File.
64. Организация проекта. Управление проектом: создание, сохранение и открытие, настройки.
65. Дизайнер форм. Редактор кода.
66. Назначение компонента Timer.
67. Назначение и свойства компонентов TextBox и Label.

68. Основные принципы ООП.
69. Назначение вкладки Standard палитры компонентов.
70. Свойства. Методы. События и обработчики событий.
71. Разработка интерфейса приложения, обработка событий.
72. Назначение компонента RadioButton.
73. Назначение компонента ListBox.
74. Компоненты вывода данных.
75. Назначение и свойства компонента Button.

***Практические задания:**

28. Разработать и произвести отладку программы для решения квадратного уравнения.
 29. Создать и отладить приложение – конвертор перевода суммы денег из долларов в рубли.
 30. Разработать и произвести отладку программы для вычисления делителей натурального числа
- N. Вывести сами делители, их количество.
31. Разработать и произвести отладку программы, вычисляющей сумму 1-й и последней цифр натурального числа N. Вывести эти цифры и сумму.
 32. Создать и отладить приложение для решения квадратного уравнения.
 33. Разработать и произвести отладку программы, находящей все простые числа в заданном диапазоне.
 34. Разработать и произвести отладку программы, находящей все нечетные числа в заданном диапазоне и их количество.
 35. Разработать и произвести отладку программы, находящей все четные числа в заданном диапазоне и их количество.
 36. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; заменить отрицательные числа на 0, положительные – на 1.
 37. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; отсортировать массив по убыванию.
 38. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; отсортировать массив по возрастанию
 39. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; поменять местами два элемента массива с номерами k1 и k2.
 40. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива, определяет минимальный и максимальный элементы массива.
 41. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива, определяет сумму всех элементов и количество положительных элементов.

Примеры заданий для проверочных, самостоятельных и контрольных работ

Задание № 1

Дайте определение понятие алгоритма. Опишите историю появления данного понятия.

Задание № 2

Составьте блок-схему вычисления и вывода среднего арифметического значения заданных чисел a,b,c.

Задание № 3

Составить блок-схему решения данной задачи и определить степень сложности. Даны сведения о соревновании фигуристов в виде: фамилия, название спортивного общества, девять оценок за выступление. Требуется по каждому из N спортивных обществ выдать справку о лучшем результате фигуриста. (Балл, полученный фигуристом, считается следующим образом: максимальная и минимальная оценки (по одной, если их несколько) отбрасываются, из остальных ищется среднее). Названия обществ заданы.

Задание № 4

Перечислите и раскройте свойства алгоритма.

Задание № 5

Составьте блок-схему алгоритма нахождения значения и вывода y, если известно, что $y=x*2$ при $x>10$, $y=x/2+1$ при $x\leq 10$.

Задание № 6

Составьте блок-схему вычисления суммы (S) всех чисел в диапазоне от 150 до 1000. Результат S вывести.

Задание № 7

Перечислите и опишите основные способы записи алгоритмов.

Задание № 8

Составьте блок-схему алгоритма нахождения значения выражения $y = \begin{cases} z * z, & x < 0 \\ z * 5, & x > 0 \end{cases}$

Задание № 9

Составьте блок-схему вычисления факториала числа 1500. Результат вывести.

Задание № 10

Перечислите (зарисуйте) блочные символы (блоки), которые используются для графического описания алгоритмов (блок-схем).

Задание № 11

Составьте блок-схему алгоритма решения следующей задачи (нахождения результата R): Даны числа a,b. Вывести квадраты чисел, если оба числа положительные и в противном случае модуль чисел a, b.

Задание № 12

Составьте блок-схему вычисления мин элемента в одномерном массиве A (I), состоящем из 70 чисел.

Задание № 13

Перечислите и опишите базовые алгоритмические конструкции.

Примеры экзаменационных билетов

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО Уральский государственный лесотехнический университет

Уральский лесотехнический колледж
ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

ОП.06. «Основы алгоритмизации и программирования»

1. Алгоритмы. Правила записи схем алгоритмов.
Определить значение переменной s после выполнения следующих операторов: `int s=0; int n=4; for (int i=2; i< n; i++) s+=10/i;`
2. ООП. Наследование.
3. Практическое задание: Разработать и произвести отладку программы:
Найти сумму бесконечного ряда. Суммировать до тех пор, пока сумма не станет больше заданного $p > 0$. Вывести эти числа.

Преподаватель: _/ /

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО Уральский государственный лесотехнический университет

Уральский лесотехнический колледж
ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 2

ОП.06. «Основы алгоритмизации и программирования»

1. Принципы объектно-ориентированного программирования: инкапсуляция.
2. Функции деления с остатком. Определить значения переменных a и b после выполнения последовательности действий:
`a= 4*5 / 3 %2; b= 4*5 /(3 %2);`
3. Практическое задание:
Разработать и произвести отладку программы для определения $N! - M!$. $N! = 1*2*3*4*... *n$

Преподаватель: _/ /

Критерии оценки за ответ на теоретические вопросы

Оценка	Критерии оценки ответа студента
«Отлично»	Обстоятельно и с достаточной полнотой излагает материал вопро- сов. Даёт ответ на вопрос в определенной логической последователь- сти. Даёт правильные формулировки, точные определения понятий и терминов. Демонстрирует полное понимание материала, даёт полный и аргу- ментированный ответ на вопрос, приводит необходимые примеры (не только рассмотренные на занятиях, но и подобранные самосто- ятельно). Свободно владеет речью (показывает связанность и последователь- ность в изложении).
«Хорошо»	Даёт ответ, удовлетворяющий тем же требованиям, что и для оцен- ки «отлично», но допускает единичные ошибки, неточности, кото- рые сам же исправляет после замечаний преподавателя.
«Удовлетворительно»	Обнаруживает знание и понимание основных положений, но: – допускает неточности в формулировке определений, терминов; – излагает материал недостаточно связно и последовательно; – на вопросы экзаменаторов отвечает некорректно.
«Неудовлетворительно»	Обнаруживает непонимание основного содержания учебного мате- риала. Допускает в формулировке определений ошибки, искажающие их смысл. Допускает существенные ошибки, которые не может исправить при наводящих вопросах преподавателя или ответ отсутствует. Беспорядочно и неуверенно излагает материал. Сопровождает изложение частыми заминками и перерывами.

Критерии оценки за выполнение практического задания

Оценка	Критерии
«Отлично»	Показал полное знание технологии выполнения задания. Продemonстрировал умение применять теоретические зна- ния/правила выполнения/технологию при выполнении задания. Уверенно выполнил действия согласно условию задания.
«Хорошо»	Задание в целом выполнил, но допустил неточности. Показал знание технологии/алгоритма выполнения задания, но не- достаточно уверенно применил их на практике. Выполнил норматив на положительную оценку.
«Удовлетворительно»	Показал знание общих положений, задание выполнил с ошибками. Задание выполнил на положительную оценку, но превысил время, отведенное на выполнение задания.

«Неудовлетворительно»	Не выполнил задание. Не продемонстрировал умения самостоятельного выполнения задания. Не знает технологию/алгоритм выполнения задания. Не выполнил норматив на положительную оценку.
-----------------------	---