

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ПРОФЕССИОНАЛЬНАЯ
ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ
«ЯКУТСКИЙ ГУМАНИТАРНЫЙ КОЛЛЕДЖ»

Одобрено на заседании
Педагогического совета
протокол № 5 от 28.04.2025 г.

УТВЕРЖДАЮ

Зам. директора по учебной работе
А.Д. Рабинович



Рабочая программа дисциплины

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

По специальности среднего профессионального образования
09.02.07 Информационные системы и программирование

Уровень образования: основное общее образование, среднее общее образование

Форма обучения: очная

Якутск, 2025

1. ПАСПОРТ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования

1.1 Область применения рабочей программы

Рабочая программа учебной дисциплины является частью рабочей основной профессиональной образовательной программы подготовки специалистов среднего звена в соответствии с Министерством просвещения Российской Федерации по специальности 09.02.07 Информационные системы и программирование.

1.2 Место учебной дисциплины «Основы алгоритмизации» в структуре основной профессиональной образовательной программы

Учебная дисциплина ОП 04 «Основы алгоритмизации и программирования» относится к общепрофессиональному циклу основной профессиональной образовательной программы подготовки специалистов среднего звена в соответствии с Министерством просвещения Российской Федерации по специальности 09.02.07 Информационные системы и программирование.

1.3 Цель и задачи учебной дисциплины - требования к результатам освоения учебной дисциплины

Целью освоения дисциплины ОП 04 «Основы алгоритмизации и программирования» является изучение основных численных методов решения прикладных задач с использованием вычислительной техники для использования в профессиональной, научно-исследовательской, научно-исследовательской, производственно-технологической деятельности в сфере информационных технологий.

В результате освоения учебной дисциплины «Основы алгоритмизации и программирования» обучающиеся по специальности среднего профессионального образования Информационные системы и программирование (09.02.07) должны

уметь:

- формализовать поставленную задачу;
- применять полученные знания к различным предметным областям;
- составлять и оформлять программы на языках программирования;
- тестировать и отлаживать программы;

знать:

- общие принципы построения и использования языков программирования, их классификацию;
 - современные интегрированные среды разработки программ;
 - процесс создания программ;
 - стандарты языков программирования, общую характеристику языков ассемблера;
- назначение, принципы построения и использования

1.4. Рекомендуемое количество часов на освоение программы дисциплины:

максимальной учебной нагрузки обучающегося 169 часов, в том числе:

обязательной аудиторной учебной нагрузки обучающегося: очная форма обучения- 130 часов.

2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1 Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	169
Обязательная аудиторная учебная нагрузка (всего)	130
в том числе:	
теоретическое обучение	66
практические занятия	64
консультации	
Самостоятельная работа обучающегося (всего)	9
в том числе:	
Промежуточная аттестация	30
Итоговая аттестация в форме: 1 семестр – другие, 2 семестр – экзамен	

2.2

Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала	Объем часов	Уровень освоения	Осваиваемые элементы компетенций	Код личностных результатов реализации программы воспитания
1	2	3			4
Раздел 1.	1. Введение в программирование	10	3	О	
Тема 1.1. Языки программирования	Содержание учебного материала	8		К 1	О
	1. Понятие алгоритма, его свойства. Сущность алгоритмизации вычислительного процесса. Способы описания алгоритмов.			К 2	О
	2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.			К 4	О
	3. Жизненный цикл программы. Программа. Программный продукт и его характеристики.			К 1.1	П
	4. Основные этапы решения задач на компьютере.				
	В том числе практических занятий и лабораторных работ	2			
	Знакомство со средой программирования. Составление программ линейной структуры.				
Тема 1.2. Типы данных	Содержание учебного материала	8	3	О	ЛР
	1. Типы данных. Простые типы данных.	4		К 1	О
	2. Производные типы данных. Структурированные типы данных.			К 2	О
	В том числе практических занятий и лабораторных работ	4		К 4	О
	Составление программ разветвляющейся структуры. Составление программ циклической структуры				
Раздел 2.	2. Основы программирования	20	3	О	ЛР

Тема 2.1. Операторы языка программирования	1. Операции и выражения.	10		К 1 О К 2 О К 4 О К 5 О К 9 О К 10 П К 11-1.2	14,15,16
	2. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.				
	3. Условный оператор. Оператор выбора.				
	4. Цикл с постусловием. Цикл с предусловием. Цикл с параметром.				
	Вложенные циклы.				
	5.Массивы. Двумерные массивы.				
	6. Строки. Стандартные процедуры и функции для работы со строками.				
	7.Структурированный тип данных – множество.				
	8.Операции над множествами.				
	9.Комбинированный тип данных – запись.				
	10.Файлы последовательного доступа. Файлы прямого доступа	10			
	В том числе практических занятий и лабораторных работ				
	Обработка одномерных массивов.				
	Обработка двумерных массивов. Работа со строками.				
	Работа с данными типа множество. Файлы последовательного доступа. Типизированные файлы.				
Раздел 3.	3. Языки и системы программирования	10			
Тема 3.1. Структуризация в программировании	1. Основы структурного программирования. Методы структурного программирования.	4	3	О К 1 О К 2 ОК 4	ЛР 14,15,16
	В том числе практических занятий и лабораторных работ				
	Создание библиотеки подпрограмм.	6			
Тема 3.2. Модульное программирование	Содержание учебного материала	12			
	1. Модульное программирование. Понятие модуля. Структура модуля.	6			
	Компиляция и компоновка программы.				
	2. Стандартные модули.				
	В том числе практических занятий и лабораторных работ	6			
	Программирование модуля.				
Раздел 4	4. Основные конструкции языков программирования	8	3	О К 1 О К 2 О	ЛР 13,14,15,16
Тема 4.1. Указатели.	Содержание учебного материала	4			
	1. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных				

	2. Структуры данных на основе указателей. 3. Задача о стеке. В том числе практических занятий и лабораторных работ Использование указателей для организации связанных списков.	4		К 4 П К 1.1-ПК 1.5 П К 2.4, 2.5	
Раздел 5	5. Разработка приложения	16			
Тема 5.1. Визуальное событийно- управляемое программиров ание	1. Основные компоненты (элементы управления) интегрированной среды раз- работки, 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. В том числе практических занятий и лабораторных работ Создание процедур на основе событий. Создание проекта с использованием кнопочных компонентов. Создание проекта с использованием компонентов стандартных диалогов и системы меню.	10	3	П К 1.5 ПК 2.4, 2.5	ЛР 13,14,15,16
		6			
		30			
		14			
		16			
		Разработка функциональной схемы работы приложения.			
Тема 5.4. Разработка оконного приложения	Содержание учебного материала 1. Разработка функционального интерфейса приложения. Создание интерфейса приложения. 2. Разработка функциональной схемы работы приложения. 3. Разработка игрового приложения. В том числе практических занятий и лабораторных работ Разработка функциональной схемы работы приложения. Разработка оконного приложения с несколькими формами. Тестирование, отладка приложения. Разработка игрового приложения				
		9			
		130			

3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1 Требования к минимальному материально-техническому обеспечению

Перечень основного оборудования, учебно-наглядных пособий:

- комплект мебели для обучающихся (стол ученический – 6, стул ученический – 12);
- маркерная доска – 1,
- автоматизированные рабочие места на 16 обучающихся (Мини ПК DEXP MINI ENTRY (Intel N100, 8 ГБ DDR4, SSD 256 ГБ, Windows 11 Pro, 2 x HDMI, Wi-Fi, Bluetooth, SoC, блок питания – 36) Монитор, клавиатура, компьютерная мышь, доступ в интернет,
- автоматизированное рабочее место преподавателя – 1 (Мини ПК DEXP MINI ENTRY (Intel N100, 8 ГБ DDR4, SSD 256 ГБ, Windows 11 Pro, 2 x HDMI, Wi-Fi, Bluetooth, SoC, блок питания – 36) Монитор, клавиатура, компьютерная мышь, доступ в интернет, МФУ Canon imageCLASS MF3010 A4, 18 стр./мин, 64 Мб, 1200х600 dpi, USB, лоток 150 л. 5252B008 / 5252B011 / 5252B007)
- сервер (удаленно),
- мультимедиа-проектор – 1,
- экран настенный -1,
- комплект учебников (учебных пособий),
- жалюзи – 3,
- доступ в интернет – 16
- комплект наглядных пособий по предметам учебного плана;
- облучатель - рециркулятор бактерицидный для обеззараживания воздуха «AirRec»;
- программное обеспечение:

Пакет программ MicrosoftOffice (**Microsoft Word, Microsoft Excel, Microsoft PowerPoint, Microsoft Outlook, Microsoft Access, Microsoft OneNote, Microsoft Teams**);

ABCPascal(свободно распространяемое программное обеспечение)

Visual Studio Community (Бесплатная версия)

SQL Server Management Studio (Бесплатная версия)

Embarcadero Delphi. Community (Бесплатная версия)

MicrosoftSQLServerJavaConnector (свободно распространяемое программное обеспечение)

PN KL 4851RATFQ Kaspersky WorkSpace Security Russian Edition. 250-499 User 1

3.2. Информационное обеспечение обучения

Основная литература:

1. Программирование : учебное пособие / О. Ю. Горбадей, О. П. Рябычина, А. Н. Балаш, Н. И. Янович. – Минск : РИПО, 2023. – 300 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=717782>. – Библиогр. в кн. – ISBN 978-985-895-148-1. – Текст : электронный.
2. Окулов, С. М. Программирование в алгоритмах : учебное пособие : [12+] / С. М. Окулов. – 7-е изд. – Москва : Лаборатория знаний, 2021. – 386 с. : ил., табл., схем., граф. – (Развитие интеллекта школьников). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=712953>. – ISBN 978-5-93208-521-9. – Текст : электронный.
3. Нагаева, И. А. Основы алгоритмизации и программирования: практикум : учебное пособие : [12+] / И. А. Нагаева, И. А. Кузнецов. – Москва ; Берлин : Директ-Медиа, 2021. – 168 с. : схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=598404>. – Библиогр.: с. 162-163. – ISBN 978-5-4499-1612-9. – DOI 10.23681/598404. – Текст : электронный.

Дополнительная литература:

1. Лебеденко, Л. Ф. Основы программирования на C++ : учебное пособие : [16+] / Л. Ф. Лебеденко, О. И. Моренкова. – 2-е изд., перераб. и доп. – Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2021. – 200 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=694769>. – Библиогр. в кн. – Текст : электронный.
2. Долгов, А. И. Алгоритмизация прикладных задач : учебное пособие : [16+] / А. И. Долгов. – Москва : ФЛИНТА, 2021. – 136 с. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=83142>. – ISBN 978-5-9765-0086-2. – Текст : электронный.
3. Лубашева, Т. В. Основы алгоритмизации и программирования : учебное пособие : [12+] / Т. В. Лубашева, Б. А. Железко. – Минск : РИПО, 2016. – 378 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=463632>. – Библиогр. в кн. – ISBN 978-985-503-625-9. – Текст : электронный.

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.

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения	Критерии оценки
умения: Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. Эволюцию языков программирования, их классификацию, понятиесистемы 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 объект-но-ориентированного программирования на примере алгоритмического языка:	Формы и методы контроля и оценки результатов обучения Формы контроля обучения: Контроль домашнего задания Контрольная работа Защита практической работы Отчетная работа по выполнению практической работы Дифференцированный зачет Формы оценки результативности обучения: Традиционная система баллов, на основе которых выставляется итоговая отметка Методы контроля направлены на проверку умения обучающихся: Выполнять условие задания на творческом уровне с представлением собственной позиции Делать осознанный выбор способов действий из ранее известных Осуществлять коррекцию сделанных ошибок на новом уровне предлагаемых заданий - Работать в группе и представлять как свою, так и позицию группы Методы оценки результатов обучения: Мониторинг роста творческой самостоятельности и навыков получения нового знания каждым обучающимся	«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. «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

знания: Разрабатывать алгоритмы для конкретных задач. Использовать программы для графического отображения алгоритмов.		учебные задания содержат грубые ошибки.
ЛР 13 Демонстрирующий умение эффективно взаимодействовать в команде, вести диалог, в том числе с использованием средств коммуникации		– участие в конкурсах профессионального мастерства, олимпиадах по профессии, викторинах, в предметных неделях; – конструктивное взаимодействие в учебном коллективе/бригаде; – участие в исследовательской и проектной работе;
ЛР 14 Демонстрирующий навыки анализа и интерпретации информации из различных источников с учетом нормативно-правовых норм		□ проявление культуры потребления информации, умений и навыков пользования компьютерной техникой, навыков отбора и критического анализа информации, умения ориентироваться в информационном пространстве;
ЛР 15 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.		– участие в конкурсах профессионального мастерства, олимпиадах по профессии, викторинах, в предметных неделях; – конструктивное взаимодействие в учебном коллективе/бригаде; – участие в исследовательской и проектной

1.ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.04 «Основы алгоритмизации и программирования»

1.1. Область применения фонда оценочных средств

Фонд оценочных средств предназначен для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Основы алгоритмизации и программирования».

ФОС включают контрольные материалы для проведения текущего контроля в форме выполнения тестовых заданий, контрольных и проектных заданий и промежуточной аттестации в форме дифференцированного зачета.

1.2. Требования к результатам освоения дисциплины

В результате освоения учебной дисциплины обучающийся должен обладать предусмотренными ФГОС по специальности 09.02.07 Информационные системы и программирование следующими знаниями и умениями:

З1:- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;

З2:- эволюцию языков программирования, их классификацию, понятие системы программирования;

З3:- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;

З4:- подпрограммы, составление библиотек подпрограмм ;

З5:- объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка;

З6:- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

У1:- разрабатывать алгоритмы для конкретных задач;

У2:- использовать программы для графического отображения алгоритмов. - определять сложность работы алгоритмов;

У3:- работать в среде программирования;

У4:- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;

У5:- оформлять код программы в соответствии со стандартом кодирования;

У6:- выполнять проверку, отладку кода программы.

В соответствии с ФГОС СПО по специальности Информационные системы и программирование в рамках освоения учебной дисциплины «Основы алгоритмизации и программирования» у студентов формируются следующие общие компетенции (ОК):

Код	Наименование результата обучения
ОК 1	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК 2	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 4	Эффективно взаимодействовать и работать в коллективе и команде;
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.
ОК 9	Пользоваться профессиональной документацией на государственном и иностранном языках.

профессиональные компетенции (ПК):

Код	Наименование результата обучения
ПК 1.1.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
ПК 1.2	Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности.
ПК 1.3	Производить модификацию отдельных модулей информационной системы в соответствии с рабочим заданием, документировать произведенные изменения.
ПК 1.4.	Выполнять тестирование программных модулей.
ПК 1.5.	Осуществлять рефакторинг и оптимизацию программного кода.
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.

2. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.04 «Основы алгоритмизации и программирования»

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

Результаты обучения	Критерии оценки	Формы и методы контроля и оценки
Знания		
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. -	Критерии формирования оценки за устный ответ: Оценка «5 (отлично)» ставится, если обучающийся: полно и аргументировано отвечает по содержанию вопроса; обнаруживает понимание материала, Оценка «4 (хорошо)» ставится, если обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 ошибки, которые сам же исправляет. Оценка «3 (удовлетворительно)» ставится, если обучающийся обнаруживает знание и понимание основных положений темы, но: излагает материал неполно и допускает неточности в определении понятий; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки. Оценка «2 (неудовлетворительно)» ставится, если обучающийся обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал. Критерии оценки результатов тестирования «5» - 80-100% верных ответов «4» - 79-60% верных ответов «3» - 59-50% верных ответов «2» - 49% и менее	Текущий контроль в форме: - устно - тестирования; - оценка результатов самостоятельной работы Промежуточная аттестация в форме: экзамена
Умения		

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.	Планировано и реализовано собственное профессиональное и личностное развитие. Критерии оценивания результатов практических работ: Оценка 5 «отлично»- если студент: показал полное знание технологии выполнения задания; продемонстрировал умение применять теоретические знания/правила выполнения/технология при выполнении задания; уверенно выполнил действия согласно условию задания; работа выполнена в установленное время. Оценка 4 «хорошо» - если студент: задание в целом выполнил, но допустил неточности; показал знание технологии/алгоритма выполнения задания, но недостаточно уверенно применил их на практике; выполнил норматив на положительную оценку; работа выполнена в установленное время. Оценка 3 «удовлетворительно» - если студент: показал знание общих положений, задание выполнил с ошибками; задание выполнил на положительную оценку, но превысил время, отведенное на выполнение задания Оценка 2 «неудовлетворительно» -- если студент: не выполнил задание; не продемонстрировал умения самостоятельного выполнения задания; не знает технологию/алгоритм выполнения задания; не выполнил норматив на положительную оценку.	Текущий контроль в форме: - наблюдение за выполнением заданий и оценки на практических занятиях; - оценка результатов выполнения самостоятельной работы. Промежуточная аттестация в форме: экзамена
---	---	--

Контроль и оценка освоения учебной дисциплины по разделам (темам)

Раздел / тема учебной дисциплины	Форма текущего контроля	Коды знаний и умений	Коды формируемых ПК и ОК
Тема 1 Введение в программирование	Тестирование	31	ОК 01, ОК02, ОК 05
Тема 2 1. Основы программирования	Тестирование, Проверочная работа	31, У1, У4, У5, У6	ОК 01, ОК02, ПК 2.4
Тема 3 1. Языки и системы программирования	Тестирование, Самостоятельная работа	31, У1, У4, У5, У6	ОК 01, ОК02, ОК 04, ОК 05, ПК 1.1. ПК 1.2 ПК 1.3 ПК 1.4. ПК 1.5. ПК 2.4, ПК 2.5
Тема 4.1. Основные конструкции языков программирования	Тестирование Проверочная работа	34, У1, У4, У5, У6	ОК 01, ОК02, ОК 04, ПК 2.4
Тема 5 1. Разработка приложения	Устный опрос	32, 35, У3	ОК 01, ОК02, ОК 04, ОК 09 ПК 2.4, ПК 2.5

Критерии и шкала оценивания в результате изучения дисциплины при проведении текущего контроля и промежуточной аттестации:

Критерии оценки за ответ на теоретические вопросы

Шкала оценивания	Критерии оценки
«отлично»	Обстоятельно и с достаточной полнотой излагает материал вопросов. Даёт ответ на вопрос в определенной логической последовательности. Даёт правильные формулировки, точные определения понятий и терминов. Демонстрирует полное понимание материала, даёт полный и аргументированный ответ на вопрос, приводит необходимые примеры (не только рассмотренные на занятиях, но и подобранные самостоятельно). Свободно владеет речью (показывает связанность и последовательность в изложении).
«хорошо»	Даёт ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает единичные ошибки, неточности, которые 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 материала. 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

Шкала оценивания	Критерии оценки
«отлично»	Показал полное знание технологии выполнения задания. Продemonстрировал умение применять теоретические знания/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 недостаточно 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.

1. ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

При проведении промежуточной аттестации в колледже используются традиционные формы аттестации:

Элемент модуля	Форма промежуточной аттестации	Шкала оценивания
ОП.04 Основы алгоритмизации и программирования	Экзамен	"отлично", "хорошо", "удовлетворительно", "неудовлетворительно"

ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ И (ИЛИ) ДЛЯ ИТОГОВОГО КОНТРОЛЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИИ

Промежуточный контроль (аттестация) обучающихся по дисциплине «Основы алгоритмизации и программирования» проводится в *3 семестре* проводится в форме зачета, в *4 семестре* проводится в форме экзамена.

Тест включает 25 вопросов (приложение 1) по изученному материалу.

Критерии оценок.

- ✓ **Оценка «5»** - 85-100%
- ✓ **Оценка «4»** - 70-84 %
- ✓ **Оценка «3»** - 55 -69 %
- ✓ **Оценка «2»** - ниже 54%

Приложение 1

Перечень вопросов к зачету

1. Перечислите этапы решения задач в правильной последовательности

1) **постановка задачи, математическая формализация, построение алгоритма, перевод алгоритма на язык программирования, отладка и тестирование программы;**

2) построение алгоритма, математическая формализация, постановка задачи, перевод алгоритма на язык программирования, отладка и тестирование программы;

3) построение алгоритма, перевод алгоритма на язык программирования, постановка задачи, математическая формализация, отладка и тестирование программы.

2. Понятное и точное предписание исполнителю выполнить конечную последовательность команд, приводящую от исходных данных к искомому результату, называется:

- 1) моделью;
- 2) **алгоритмом;**
- 3) системой;
- 4) технологией.

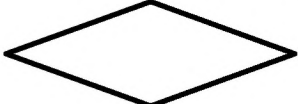
3. Представление алгоритма с помощью схем алгоритмов называется:

- 1) словесным;
- 2) **графическим;**
- 3) псевдокодами.

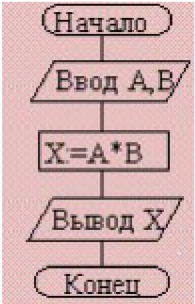
4. Когда некоторые этапы алгоритма повторяются многократно, алгоритмическая конструкция носит название:

- 1) линейной;
- 2) ветвления;
- 3) **циклической.**

5. В блок-схеме начало циклического алгоритма с счетчиком, обозначает фигура:

- 1)  - правильный ответ
- 2) 
- 3) 
- 4) 

6. Алгоритм какого типа изображен на блок-схеме

- 1) 
- 2) циклический;
- 3) разветвляющийся;
- 4) **линейный.**

7. Величина, к которой обращаются по имени, принимающая различные значения в ходе выполнения программы, называется:

- 1) константой;
- 2) **переменной;**
- 3) строкой;
- 4) символом.

8. Переменная, изменяющая свое значение при каждом вхождении в цикл, называется:

- 1) телом цикла;
- 2) **параметром цикла;**
- 3) индексом;
- 4) размером.

9. Выражению $\sqrt{x^2 - x^3} - \frac{7x}{x^3 - 15x}$ соответствует следующая запись на языке программирования:

- 1) **math.sqrt(x**2-x**3)-((7*x)/(x**3-15*x));**
- 2) **math.sqrt(x**2-x**3)-(7*x/x**3-15*x);**
- 3) **sqrt(x**2-x**3)-((7*x)/(x**3-15*x));**
- 4) **math.sqrt(x*2-x*3)-((7*x)/(x*3-15*x)).**

10. Запись алгоритма с использованием фраз естественного и алгоритмического языка называется:

- 1) словесным;
- 2) графическим;
- 3) **псевдокодом.**

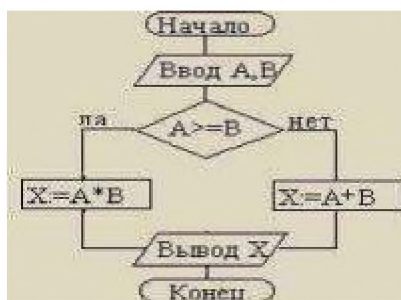
11. Алгоритмическая конструкция, предполагающая выполнение либо одного, либо другого действия в зависимости от истинности или ложности условия, называется:

- 1) линейной;
- 2) **ветвлением;**
- 3) циклической.

12. Многократно повторяющаяся часть алгоритма называется:

- 1) параметром цикла;
- 2) **телом цикла;**
- 3) перебором.

13. Алгоритм какого типа изображен на блок-схеме



- 1)
- 2)
- 3)
- 4)

циклический;
разветвляющийся;
вспомогательный;
линейный.

14. Чем характеризуется переменная?

- 1) **именем, типом, значением;**
- 2) именем, значением;
- 3) значением, типом.

15. Линейная структура построения программы подразумевает:

- 1) Неоднократное повторение отдельных частей программы;
- 2) **Последовательное выполнение всех элементов программы;**
- 3) Выполнение лишь нескольких, удовлетворяющих заданному условию частей программы;
- 4) Верного ответа нет.

16. Оператор присваивания имеет вид:

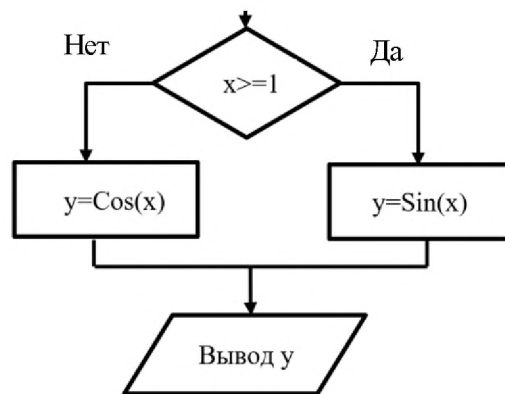
- 1) = ;
- 2) := ;
- 3) =: ;
- 4) Верного ответа нет.

17. Ветвящийся алгоритм подразумевает:

- 1) Неоднократное повторение отдельных частей программы;
- 2) Последовательное выполнение всех элементов программы;
- 3) **Выполнение лишь нескольких, удовлетворяющих заданному условию частей программы;**
- 4) Верного ответа нет.

18. Установить правильную последовательность команд согласно части блок-схемы

- 1) `y=math.sin(x)`
- 2) `print("y=", y)`
- 3) `if x>=1:`
- 4) `y=math.cos(x)`
- 5) `else:`



Ответ: 3, 1, 5, 4, 2

19. Установить правильную последовательность команд

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

- 1) `-4*a*c))`
- 2) `+math.sqrt(`
- 3) `/(2*a)`
- 4) `(-b`
- 5) `math.pow(b,2)`

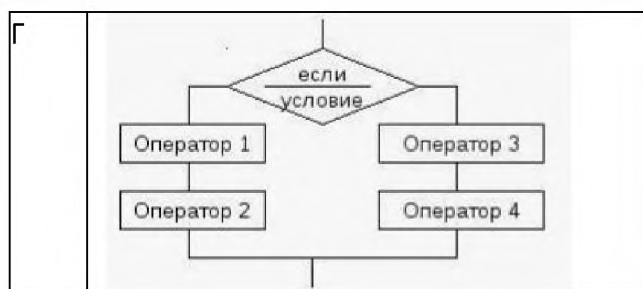
Ответ: 4, 2, 5, 1, 3

20. Установите правильное соответствие между определением и блок-схемой.

1	Простой условный оператор
2	Составной условный оператор
3	Сокращенный условный оператор

А	
Б	
В	

4	Многозначное ветвление
---	------------------------



1 – В; 2 – Г; 3 – А; 4 – Б

21. Определите соответствие Escape-последовательности и определением.

1	\n
2	\t
3	\\
4	\'
5	\"

А	Выводит одиночную кавычку
Б	Переводит курсор на следующую строку
В	Выводит двойную кавычку
Г	Равносильно использованию клавиши Tab
Д	Выводит обратный слеш

1 – Б; 2 – Г; 3 – Д; 4 – А; 5 – В

22. Определите соответствие функций приведения типов и определений.

1	bool()
2	int()
3	float()
4	str()
5	list()
6	tuple()

А	приводит объект к строковому типу
Б	преобразует элементы последовательности в список
В	приводит объект к логическому типу
Г	приводит объект к вещественному типу
Д	преобразует элементы последовательности в кортеж
Е	приводит объект к целому типу

1 – В; 2 – Е; 3 – Г; 4 – А; 5 – Б; 6 – Д

23. Определите соответствие математических функций и действий, выполняемых функциями.

1	math.abs(x)
2	math.sqrt(x)
3	math.factorial(x)
4	math.sin(x)
5	math.cos(x)
6	math.radians(x)

А	Возвращает результат извлечения квадратного корня числа x
Б	Возвращает значение функции Cos от числа x
В	Возвращает значение функции Sin от числа x
Г	Преобразует градусы в радианы
Д	Возвращает абсолютную величину числа x

1 – Г; 2 – А; 3 – Е; 4 – В; 5 – Б; 6 – Г

24. Укажите основные виды алгоритмов:

1. **Линейный алгоритм;**
2. **Ветвящийся алгоритм;**
3. **Зацикленный алгоритм;**
4. **Циклический алгоритм;**
5. **Развивающийся алгоритм.**

25. Алгоритм — это последовательность команд, в результате выполнения которой, решается поставленная задача.

Промежуточный контроль (аттестация) обучающихся по дисциплине Основы алгоритмизации и программирования проводится в 4 семестре в форме экзамена.

К экзамену допускаются обучающиеся, имеющие выполненные, оформленные, проверенные и защищенные на положительную оценку практические работы.

Перечень вопросов к экзамену

1. Какой алгоритм называется линейным? Нарисуйте общий вид линейного алгоритма.

Линейный алгоритм — это алгоритм, образуемый командами, которые выполняются однократно и именно в той последовательности, в которой записаны.

Общий вид линейного алгоритма можно представить в виде блок-схемы, которая представляет собой геометрические фигуры (блоки), соединённые стрелками. Стрелки показывают связь между этапами и последовательность их выполнения. Каждый блок сопровождается надписью.

Пример линейного алгоритма — алгоритм «Открой дверь»:

1. **Начало.**
2. **Достань ключ из кармана.**
3. **Вставь ключ в замочную скважину.**
4. **Поверни ключ два раза.**
5. **Вытащи ключ.**
6. **Конец.**

2. С какой целью используется оператор `exit (0)` в программах, написанных на языке Python?

Оператор `exit(0)` в программах на языке Python используется для успешного завершения программы.

Принято считать, что статус 0 означает успешное выполнение, а любой ненулевой статус указывает на ошибку или ненормальное завершение.

3. Поясните назначение метода `format` и приведите примеры его применения.

Назначение метода `format()` в Python — форматирование строк. Он позволяет вставлять значения в строки, заменяя заданные метки формата в строке на конкретные значения. Это полезно, когда нужно выводить строки, которые зависят от некоторых данных, например, при выводе результатов вычислений.

Примеры применения метода `format()`: вывод числа с определённым числом знаков после запятой.

```
pi = 3.141592653589793
formatted_pi = "The value of pi is {:.2f}".format(pi)
print(formatted_pi) # "The value of pi is 3.14"
```

4. Что называется разветвляющимся алгоритмом?

Разветвляющийся алгоритм — это алгоритм, в котором действие выполняется по одной из возможных ветвей решения задачи, в зависимости от выполнения условий.

Каждое из возможных направлений дальнейших действий называется ветвью.

Разветвляющиеся алгоритмы могут быть полными и неполными.

Полное ветвление — это тип алгоритма, в котором выполняется проверка условия и в зависимости от результата проверки выполняется то или иное действие.

Например: «Если (условие выполняется) телефон разрядился, то (истина) заряджу его, иначе (ложь) буду им пользоваться дальше».

Неполное ветвление — это тип алгоритма, в котором проверка условия не выполняется. Если условие не выполняется, то выполняется действие, расположенное после «если». Например: «Если пойдёт дождь, то возьму зонт».

5. Как записывается простой условный оператор в блок-схемах, в программах? Как работает простой условный оператор?

Простой условный оператор в Python записывается с использованием оператора if. Затем следует само условие, после идёт двоеточие и с новой строки с отступом начинается код.

Пример:

```
x = 10
if x > 5:
    print('x больше 5')
```

В этом примере, если переменная x больше пяти, Python выведет в консоль сообщение «x больше 5». В остальных случаях код внутри блока if не будет выполняться, и пользователь не увидит сообщение.

Конструкция if-else позволяет выполнять два блока кода: для истинного и ложного условий. Это позволяет управлять поведением программы в зависимости от входных данных и состояний.

Синтаксис:

```
if условие:
    Блок_кода_1
else:
    Блок_кода_2
```

Простой условный оператор в Python работает так: программа оценивает тестовое выражение и выполняет оператор, только если тестовое выражение истинно. Если тестовое выражение имеет значение False, оператор не выполняется.

6. Как записывается сокращенный условный оператор в блок-схемах, в программах? Как работает сокращенный условный оператор?

Сокращённый условный оператор в Python записывается с использованием условных выражений (тернарных операторов). Они позволяют сократить код условного оператора до одной строки.

Синтаксис:

<выражение1> if <условие> else <выражение2> Пример:

```
age = 18
status = "голосующий" if age >= 18 else "неголосующий"
print(f"Вы являетесь {status}.")
```

Работа сокращённого условного оператора: если условие истинно, выполняется блок кода, находящийся внутри if. Если условие ложно, выполняется блок кода, находящийся внутри else.

Также в Python позволительна сокращённая запись сложного логического выражения, например:

```
if 0 < a < b: print(b - a)
```

7. Как записывается составной условный оператор в блок-схемах, в программах? Как работает составной условный оператор?

Составной условный оператор в Python записывается с использованием оператора elif. Он позволяет проверять несколько выражений на истинность и выполнять блок кода, как только одно из условий оценивается как истинное.

Конструкция оператора elif:

```
if выражение1: оператор(ы)
elif выражение2: оператор(ы)
elif выражение3: оператор(ы)
else: оператор(ы)
```

Оператор else в этом случае содержит блок кода, который выполняется, если условное выражение в операторе if принимает значение 0 или ложь (FALSE). Оператор else является необязательным, и за ним может быть только один оператор else.

Работа составного условного оператора заключается в том, что он выбирает не более одного блока команд, который будет выполнен. Если ни одно условие не выполняется и не задана секция else, условный оператор не перейдёт ни к какому блоку команд и передаст управление команде, следующей сразу после последнего блока команд, описанного в условном операторе.

8. Как записываются многозначные ветвления в блок-схемах, в программах?

В языке Python многозначные ветвления записываются с помощью каскадного ветвления. Для этого между блоками if и else добавляется ещё один блок — elif с некоторым условием и инструкциями.

Пример записи:

```
if условие 1:
    Блок инструкций 1
```


elif условие 2:

Блок инструкций 2 else:

Блок инструкций 3

Дополнительных условий и связанных с ними блоков elif может быть сколько угодно, но важно отметить, что в такой сложной конструкции будет выполнен всегда только один блок кода. Другими словами, как только некоторое условие оказалось истинным, соответствующий блок кода выполняется, и дальнейшие условия не проверяются.

Также внутри условного оператора могут находиться любые операторы, в том числе и другие условные операторы, что позволяет сделать выбор не из двух, а из нескольких вариантов.

9. Как работает условный оператор if при проверке нескольких условий?

Условный оператор if в Python позволяет выполнить блок кода, если заданное условие истинно. Если условие равно False, блок кода будет пропущен.

Для проверки нескольких условий одновременно в операторе if в Python можно использовать логические операторы:

- **and** — возвращает True, если оба условия истинны, и False — если ложны;
- **or** — возвращает True, если хотя бы одно из условий истинно, и False в противном случае;
- **not** — возвращает True, если условие ложно, и False — если истинно.

Также для последовательной проверки нескольких условий можно использовать оператор if-elif-else:

1. **if** — проверяет первое условие;
2. **elif** (сокращение от else if) — проверяет следующие условия, если предыдущее условие ложно;
3. **else** — выполняет блок кода, если все предыдущие условия ложны.

10. Дайте определение циклического алгоритма.

Циклический алгоритм — это алгоритм, в котором некоторая часть операций повторяется многократно.

Он описывает действия, которые должны повторяться указанное число раз или пока не выполнено заданное условие.

11. Расскажите о работе оператора цикла for по возрастающим значениям параметра, нарисовав общий вид алгоритма и синтаксис этого оператора.

Оператор цикла for в Python поэлементно перебирает последовательность и выполняет код, который записан в теле цикла. Главное условие успешной работы цикла — объект должен быть итерируемым.

Общий вид алгоритма: указывают переменную, которая будет хранить значения элементов последовательности, затем указывают саму последовательность, которую нужно перебрать. Далее в теле цикла выполняют действия с каждым элементом последовательности.

Синтаксис цикла for:

```
for [элемент] in [последовательность]: [тело цикла]
```

Пример использования: чтобы просуммировать значения чисел от 1 до n, можно воспользоваться следующей программой:

```
sum = 0
for i in range(1, n + 1):
    sum += i
```

В этом примере переменная i принимает значения 1, 2, ..., n, и значение переменной sum последовательно увеличивается на указанные значения.

12. Расскажите о работе сложного циклического процесса, нарисовав общий вид алгоритма и синтаксис этого оператора.

Оператор цикла for в Python поэлементно перебирает последовательность и выполняет код, который записан в теле цикла. Главное условие успешной работы цикла — объект должен быть итерируемым.

Общий вид алгоритма: указывают переменную, которая будет хранить значения элементов последовательности, затем указывают саму последовательность, которую нужно перебрать. Далее в теле цикла выполняют действия с каждым элементом последовательности.

Синтаксис цикла for:

```
for [элемент] in [последовательность]: [тело цикла]
```

Пример использования: чтобы просуммировать значения чисел от 1 до n, можно воспользоваться следующей программой:

```
sum = 0
for i in range(1, n + 1):
    sum += i
```

В этом примере переменная i принимает значения 1, 2, ..., n, и значение переменной sum последовательно увеличивается на указанные значения.

13. Какой цикл называется внешним, а какой – внутренним?

Любой цикл, содержащий внутри себя один или несколько других циклов, называется вложенным. Цикл, охватывающий другие циклы, называется внешним, а остальные циклы – внутренними. Типы циклических алгоритмов внешнего и внутреннего циклов при этом могут быть любыми. Внутренний цикл, от подготовки до завершения, должен находиться внутри внешнего, точнее, быть его телом.

14. Нарисуйте общий вид алгоритма оператора цикла while. Напишите синтаксис оператора цикла while.

Общий вид алгоритма оператора цикла while в Python: сначала проверяется условие. Если оно ложно, то выполнение цикла прекращается и управление передаётся на следующую инструкцию после тела цикла while. Если условие истинно, то выполняется инструкция, после чего условие проверяется снова и снова выполняется инструкция. Так продолжается до тех пор, пока условие будет истинно. Как только условие станет ложно, работа цикла завершится и управление передастся следующей инструкции после цикла.

Синтаксис цикла while в Python:

```
while выражение: оператор(ы)
```

Здесь оператор(ы) могут быть одиночным оператором или блоком операторов. Условие может быть любым выражением, а true — любым ненулевым значением. В синтаксисе Python в конце строки с условием ставится двоеточие, а всё тело выделяется отступом (табуляцией или четырьмя пробелами).

15. Расскажите о работе оператора цикла while. Приведите примеры.

Цикл while в Python позволяет выполнять блок кода до тех пор, пока условие остаётся истинным. Это делает его особенно полезным для задач, где количество итераций заранее неизвестно.

Синтаксис цикла while:

```
while условие: блок_кода
```

Условие — это логическое выражение, которое проверяется перед каждой итерацией цикла. Если условие истинно (True), блок кода выполняется. Если условие ложно (False), выполнение цикла прекращается.

Пример базового цикла while, который выводит числа от 1 до 5: i = 1

```
while i <= 5:
    print(i)
    i += 1
```

Ещё один пример использования цикла while для определения количества цифр натурального числа n:

```
n = int(input())
length = 0
while n > 0:
    n //= 10
    length += 1
```

Ещё один пример обработки пользовательского ввода до тех пор, пока не будет введено корректное значение:

```
user_input = ""
while user_input.lower() != "exit":
    user_input = input("Введите команду (или 'exit' для выхода): ")
    print(f"Вы ввели: {user_input}")
```

Критерии зачета

Каждое тестовое задание имеет определенный порядковый номер, из которых - один

верный и три неверных ответа.

Критерии оценивания:

«отлично» - 90%-100% правильных ответов,

«хорошо»- 75%-89% правильных ответов,

«удовлетворительно»- 50%-74% правильных ответов,

«неудовлетворительно»- менее 50% правильных ответов. Время, которое отводится на выполнение теста - 90мин

Критерии контрольной работы Отметка «5 (отлично)» ставится в случае:

знания, понимания, глубины усвоения обучающимся всего объема программного материала;
творчески применять полученные знания в незнакомой ситуации; отсутствия ошибок и недочётов при воспроизведении изученного материала, при устных ответах, устранения отдельных неточностей с помощью дополнительных вопросов педагога;
соблюдения культуры письменной и устной речи, правил оформления письменных работ.

Отметка «4 (хорошо)» ставится в случае:

знания всего изученного материала;
умения выделять главные положения в изученном материале, на основании фактов и примеров обобщать, делать выводы, устанавливать межпредметные и внутрипредметные связи, применять полученные знания на практике;
наличие незначительных (негрубых) ошибок при воспроизведении изученного материала;
соблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

Отметка «3 (удовлетворительно)» ставится в случае:

- знания и усвоения материала на уровне минимальных требований программы, затруднения при самостоятельном воспроизведении, необходимости незначительной помощи учителя;
умения работать на уровне воспроизведения, затруднения при ответах на видоизменённые вопросы;
наличия 1-2 грубых ошибок, нескольких негрубых при воспроизведении изученного материала;
незначительного несоблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

Отметка «2 (неудовлетворительно)» ставится в случае:

знания и усвоения учебного материала на уровне ниже минимальных требований программы;
отсутствия умения работать на уровне воспроизведения, затруднения при ответах на стандартные вопросы;
наличия нескольких грубых ошибок, большого числа негрубых при воспроизведении изученного материала;
- значительного несоблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

Отметка «1 (неудовлетворительно)» ставится в случае:

- отказ обучающегося от ответа, выполнения работы, теста, отсутствие выполненного (в том числе, домашнего) задания.

При выставлении отметок необходимо учитывать классификацию ошибок и их количество: грубые ошибки; однотипные ошибки; негрубые ошибки; недочеты.

К грубым ошибкам следует относить:

незнание определения основных понятий, правил, неумение выделять главное в ответе; неумение делать выводы и обобщения;

неумение пользоваться первоисточниками, учебником и справочником. К **однотипным ошибкам** относятся ошибки на одно и то же правило. К **негрубым ошибкам** следует относить:

неточность формулировок, определений, понятий, правил, вызванная неполнотой охвата основных признаков определяемого понятия или замена 1-2 из этих признаков второстепенными; нерациональные методы работы с учебной и справочной литературой

2.1. Промежуточная аттестация

Вопросы к экзамену:

1. Этапы решения задач на компьютере и их характеристика.
2. Понятие и свойства алгоритма.
3. Общие принципы построения алгоритмов.
4. Способы представления алгоритмов.
5. Основные базовые алгоритмические конструкции.
6. Конструкция следование.
7. Логические операции.
8. Операции и математические выражения.
9. Общие принципы построения разветвленных алгоритмов.
10. Циклы с известным количеством повторений.
11. Циклы с неизвестным количеством повторений.
12. Итерационные циклы.
13. Понятие структуры данных.
14. Массив.
15. Обработка одномерных массивов.
16. Вложенные циклы.
17. Организация и обработка многомерных массивов.
18. Основы структурного программирования.
19. Организация и использование процедур и функций.
20. Асимптотическая сложность алгоритмов.
21. Нотация большого O : определение и применение для вычисления временной сложности.
22. Численные алгоритмы.
23. Рандомизация данных.
24. Работа с простыми числами.
25. Понятие структуры данных.
26. Связный список.
27. Организация и работа со связанным списком.
28. Однонаправленные связные списки.
29. Двухнаправленные связные списки.
30. Структуры данных Стек.
31. Структуры данных Очередь.
32. Хеш-таблицы.
33. Рекурсивные функции.
34. Использование рекурсии в базовых алгоритмах.
35. Алгоритмы сортировки.
36. Сортировка вставкой в массивах.
37. Сортировка выбора в массивах.
38. Пузырьковая сортировка.
39. Алгоритмы поиска. Линейный поиск.
40. Алгоритмы поиска. Бинарный поиск.

Практическая часть (примеры заданий)

№1. Разработать и произвести отладку программы вывода всех четырехзначных чисел, у которых сумма первых двух равна сумме двух последних.

№2. Разработать и произвести отладку программы : Дан массив P целых чисел из n

элементов, заполненный случайным образом числами из промежутка $[-10,10]$. Из элементов массива Р сформировать массив М той же размерности по возрастанию. Исходный и скорректированный массивы вывести на экран.

№3. Разработать и произвести отладку программы, находящей все нечетные числа в заданном диапазоне и их количество

№4. Разработать и произвести отладку программы : Дан массив Р целых чисел из n элементов, заполненный случайным образом числами из промежутка $[-30,30]$. Из элементов массива Р сформировать массив М из четных чисел. Исходный и скорректированный массивы вывести на экран

№5. Разработать и произвести отладку программы, находящей все четные числа в заданном диапазоне и их количество.

№6. Разработать и произвести отладку программы: Дан массив Р целых чисел из n элементов, заполненный случайным образом числами из промежутка $[-10,10]$. Из элементов массива Р сформировать массив М 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.

№7. Разработать и произвести отладку программы: В тексте имеется одна точка с запятой (;). Подсчитать количество символов до точки с запятой и после нее.

№8. Разработать и произвести отладку программы: Вывести все двухзначные числа, сумма которых равна N.

№9. 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.

№10. Разработать и произвести отладку программы : Дан текст со скобками, удалить текст в скобках вместе со скобками и вывести его отдельно.

№11. Разработать и произвести отладку программы- "перевертыш" (так называются слова, читающиеся одинаково слева направо и справа налево, например: ПОТОП, КАЗАК).

№12. Разработать и произвести отладку программы: Дана последовательность из N целых чисел. Определить произведение максимального и минимального элементов этой последовательности.

№13. Разработать и произвести отладку программы: Произведение N первых нечетных чисел равно p. Сколько сомножителей взято?

№14. Разработать и произвести отладку программы, печатающую все делители целого числа в порядке возрастания

№15. Разработать и произвести отладку программы, печатающую все делители целого числа в порядке убывания.