МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

Рабочая программа учебной дисциплины

ОП.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

для специальности

09.02.07 «Информационные системы и программирование»

Екатеринбург

2025

Чистяков Валерий Николаевич
я подтверждаю этот документ своей удостоверяющей подписью
место подписания
2026.04.08 16:59:17+05'00'

Рабочая программа ОП.04 **«Основы алгоритмизации и программирования»** разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности **09.02.07 «Информационные системы и программирование».**

Организация-разработчик: **ГАПОУ СО «Екатеринбургский монтажный колледж»**

Разработчики:
Грибова А.В. .– преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», ВКК
Кычакова Е. А.– преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж»

Рассмотрена методическим объединением Информационных технологий
Руководитель МО _____Софьина Н.А.
7.05.25

УТВЕРЖДАЮ
Заместитель директора по учебной работе
_____ Л. С. Хоринова
«13» мая 2025

СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» 4
1.1. Область применения рабочей программы 4
1.2. Место дисциплины в структуре основной образовательной программы 4
1.3. Цели и задачи учебной дисциплины 4
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ 5
3. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ 6
3.1. Количество часов на освоение рабочей программы учебной дисциплины 6
3.2. Тематический план и содержание учебной дисциплины 6
4. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ 13
4.1. Требования к минимальному материально-техническому обеспечению 13
4.2. Информационное обеспечение реализации программы 13
4.3. Общие требования к организации образовательного процесса 15
4.4. Кадровое обеспечение образовательного процесса 15
5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ 15
ФОНД ОЦЕНОЧНЫХ СРЕДСТВ 18

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

1.1. Область применения рабочей программы

Рабочая программа учебной дисциплины «Основы алгоритмизации и программирования» является частью основной образовательной программы в соответствии с ФГОС по специальности СПО **09.02.07 Информационные системы и программирование; квалификация выпускника – Разработчик веб и мультимедийных приложений** в части подготовки студентов к освоению профессиональных модулей **ПМ.05 «Проектирование и разработка информационных систем»** и **ПМ.09 «Проектирование, разработка и оптимизация веб-приложений»** и овладению профессиональных компетенций.

1.2. Место дисциплины в структуре основной образовательной программы

Учебная дисциплина входит в общепрофессиональный цикл.

1.3. Цели и задачи учебной дисциплины

В результате освоения дисциплины студент должен

знать:

- принципы построения алгоритмов;
- типы данных и базовые конструкции изучаемых языков программирования;
- основные приемы программирования, алгоритмы;
- интегрированные среды изучаемых языков программирования;
- основные принципы отладки и тестирования программных продуктов,
- основные принципы процесса разработки программного обеспечения;
- основные подходы к интегрированию программных модулей;

уметь:

- Разрабатывать алгоритмы для конкретных задач;
- Оформлять код программы в соответствии со стандартом кодирования;
- Выполнять проверку, отладку кода программы.
- составлять простые блок-схемы алгоритмов;
- работать в интегрированной среде изучаемых языков программирования;
- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;

- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля;
- осуществлять разработку кода программного модуля на современных языках программирования;
- уметь выполнять оптимизацию программного кода.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Результатом освоения учебной дисциплины является овладение обучающимися общими (ОК) и профессиональными (ПК) компетенциями:

Код	Наименование результата обучения
ОК 01.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК 04.	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05.	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09.	Пользоваться профессиональной документацией на государственном и иностранном языках. (п. 3.2 в ред. Приказа Минпросвещения России от 01.09.2022 N 796)
ПК. 5.4	Производить разработку модулей информационной системы в соответствии с техническим заданием

3. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Количество часов на освоение рабочей программы учебной дисциплины

Вид учебной работы	**Объем в часах**
Объем образовательной программы	**186**
в том числе:	
теоретическое обучение	60
практические занятия	90
Самостоятельная работа	30
Промежуточная аттестация проводится в форме экзамена	**6**

3.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	**Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовая работа (проект)** *(если предусмотрены)*	**Объем часов**	**Коды компетенций, формированию которых способствует элемент программы**
1	**2**	**3**	**4**
Раздел 1. Создание алгоритмов			
Тема 1.1. Знакомство с алгоритмическими	**Содержание**	**2**	ОК 1, ОК 2, ОК 3, ОК 4,
	Понятие **а**лгоритма. Понятие алгоритмизации, свойства алгоритмов, способы представления	2	

структурами. Построение блок-схем	алгоритмов. Этапы решения задачи с помощью ПК. Основные типы алгоритмов: линейные алгоритмы, разветвляющий алгоритмы, циклические алгоритмы.		ПК 5.4
	Тематика практических занятий и лабораторных работ	**6**	
	1. Практическое занятие «Линейные алгоритмы»	2	
	2. Практическое занятие «Алгоритмы ветвления»	2	
	3. Практическое занятие «Циклические алгоритмы»	2	
	Самостоятельная учебная работа при изучении раздела 1 Создание блок-схем линейных алгоритмов, разветвляющихся алгоритмов, циклических алгоритмов	**10**	
Раздел 2. Основы программирования на выбранном языке			
Тема 2.1. Простые структуры данных операции над ними	**Содержание**	**14**	ОК 1, ОК 2, ОК 3, ОК 4, ПК 5.4
	Понятие переменной. Оператор присваивания. Совместимость и преобразование типов. Типы данных: целые числа, дробные числа, строки, логический тип данных. Математические операции. Встроенные математические функции. Генератор случайных чисел. Операции с целыми числами. Операции с вещественными числами	2	
	Операторы ввода (вывода) информации на экран. Форматный вывод данных. Построение линейных алгоритмов с использованием математических функций. Параллельный ввод нескольких переменных. Алгоритм перестановки двух чисел. Алгоритм нахождения цифр числа	2	
	Обработка исключений при возникновении ошибок ввода-вывода. Блоки и отступы при написании кода программы.	2	
	Виды подпрограмм: функции, процедуры. Создание процедур и функций. правила документирования функций в Python с использованием docstring. Вызов подпрограммы из тела основной программы. Формальные и фактические параметры. Локальные и глобальные	2	

	переменные. Возврат значений из функции. Рекурсивные процедуры и функции		
	Логические переменные. Логические выражения и операторы. Операции сравнения.	2	
	Условные конструкции языка. Условные конструкции языка: неполная форма, полная форма, вложенные операторы. Алгоритм нахождения минимума и максимума.	2	
	Циклические конструкции языка. Основные виды циклических конструкций. Программирование с использованием операторов цикла. Алгоритмы нахождения суммы и произведения. Циклы с предусловием, с постусловием. Операторы прерывания цикла.	2	
	Тематика практических занятий и лабораторных работ	**18**	
	1. Практическое занятие «Построение линейных алгоритмов. Обработка ошибок ввода-вывода»	2	
	2. Практическое занятие «Программы линейной структуры с использованием математических функций»	2	
	3. Практическое занятие «Составление программ с разветвляющейся структурой»	2	
	4. Практическое занятие «Составление программ с разветвляющейся структурой»	2	
	5. Практическое занятие «Составление программ с циклической структурой»	2	
	6. Практическое занятие «Составление программ с циклической структурой»	2	
	7. Практическое занятие «Создание процедур и функций»	2	
	8. Практическое занятие «Создание процедур и функций»	2	
	9. Практическое занятие «Рекурсивные процедуры и функции»	2	
Тема 2.2. Структурированные типы данных языка	**Содержание**	**44**	ОК 1, ОК 2, ОК 3, ОК 4, ПК 5.4
	Строки и операции над ними. Операции и функции для обработки строковых данных. Методы для работы со строками.	2	
	Алгоритмы работы со строками: поиск необходимых элементов в строке, замена символов	2	

	строк.		
	Регулярные выражения. Правила построения регулярных выражений	2	
	Массивы. Операции над массивами. Заполнение списка случайными числами. Ввод элементов массива с клавиатуры. Вывод элементов массива на экран. Способы перебора элементов массива. Поиск элемента массива.	2	
	Массивы. Способы обработки элементов массивов.	2	
	Сортировка. Методы и функции сортировки. Сортировка слиянием. Быстрая сортировка.	2	ОК 1, ОК 2, ОК 3, ОК 4, ПК 5.4
	Алгоритмы сортировки одномерных списков: метод пузырька, метод выбора, сортировка методом подсчета. Анализ эффективности алгоритмов сортировки	2	
	Словари. Создание словаря.	2	
	Словари. Операции над словарями. Методы для обработки словарей.	2	
	Двумерные алгоритмические структуры (Матрицы). Создание матриц.	2	
	Обработка и вывод данных матрицы.	2	
	Составные структуры данных	2	
	Сортировка составных структур данных по заданному критерию	2	
	Множества. Задание множеств. Работа с элементами множества. Операции с множествами	2	
	Файлы. Запись и чтение информации из файла.	2	
	Работа с файлами определенного типа (csv, xls, doc, …)	2	
	Файлы и каталоги. Работа с файлами и каталогами.	2	
	Стек	2	
	Очередь	2	
	Формат данных json.	2	
	Сериализация /десериализация данных	2	

	Сериализация /десериализация данных	2	
Тема 2.2. Структурированные типы данных языка (продолжение)	**Тематика практических занятий и лабораторных работ**	**66**	
	1. Практическое занятие «Строки и операции над ними»	2	
	2. Практическое занятие «Строки и операции над ними»	2	
	3. Практическое занятие «Массивы и операции над ними»	2	
	4. Практическое занятие «Массивы и операции над ними»	2	
	5. Практическое занятие «Массивы и операции над ними»	2	
	6. Практическое занятие «Сортировка массивов»	2	
	7. Практическое занятие «Сортировка массивов»	2	
	8. Практическое занятие «Словари и операции над ними»	2	
	9. Практическое занятие «Словари и операции над ними»	2	ОК 1, ОК 2, ОК 3, ОК 4, ПК 5.4
	10. Практическое занятие «Множества и операции над ними»	2	
	11. Практическое занятие «Множества и операции над ними»	2	
	12. Практическое занятие «Двумерные массивы (матрицы) и работа с ними»	2	
	13. Практическое занятие «Составные структуры данных»	2	
	14. Практическое занятие «Составные структуры данных»	2	
	15. Практическое занятие «Сортировка составных структур данных»	2	
	16. Практическое занятие «Работа с файлами» (чтение и запись в файл)	2	
	17. Практическое занятие «Работа с файлами» (чтение и запись в файл)	2	
	18. Практическое занятие «Работа с файлами» (чтение и запись в файл)	2	
	19. Практическое занятие «Работа с файлами и каталогами» (операции работы с файлами и каталогами: создание, перемещение, удаление)	2	

	20. Практическое занятие «Работа с файлами и каталогами» (операции работы с файлами и каталогами: создание, перемещение, удаление)	2	
	21. Практическое занятие «Работа с файлами и каталогами» (операции работы с файлами и каталогами: создание, перемещение, удаление)	2	
	22. Практическое занятие «Стек»	2	
	23. Практическое занятие «Стек»	2	
	24. Практическое занятие «Очередь»	2	
	25. Практическое занятие «Очередь»	2	
	26. Практическое занятие «Очередь»	2	
	27. Модульное программирование	2	
	28. Модульное программирование	2	
	29. Работа с Json	2	
	30. Работа с Json	2	
	31. Сериализация/десериализация данных	2	
	32. Сериализация/десериализация данных	2	
	33. Сериализация/десериализация данных	2	
	Самостоятельная учебная работа при изучении раздела 2 1. Реализация алгоритмов с элементами игры – Игра «Угадай число» – Игра «Калькулятор» – Игра «Угадай слово» – Игра «Гадалка» – Игра «Угадай число по предложенным буквам»	20	ОК 1, ОК 2, ОК 3, ОК 4, ПК 5.4,

	2. Разработка программы по выполнению основных алгоритмов: – Алгоритм перестановки двух чисел – Алгоритм нахождения минимального элемента в последовательности – Алгоритм нахождения максимального элемента в последовательности – Алгоритм нахождения суммы – Алгоритм нахождения произведения – Алгоритм нахождения всех цифр числа – Алгоритм нахождения всех делителей числа – Алгоритм замены одного символа в строке на другой – Алгоритм нахождения элементов главной диагонали массива – Алгоритм определения того, что введенный символ является числом – Алгоритм определения того, что введенный символ является буквой латинского алфавита – Алгоритм динамического заполнения массива – Алгоритм перебора элементов массива по индексам и изменения элемента по условию – Алгоритм подсчёта количества элементов в массиве с помощью словаря – Алгоритм подсчёта количества элементов в массиве с помощью множества – Алгоритм сортировки массива (методом пузырька, методом выбора, методом подсчета, быстрая сортировка, сортировка слиянием и т.д.)		
Экзамен		6	

4. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

4.1. Требования к минимальному материально-техническому обеспечению

Реализация программы предполагает наличие стандартного учебного кабинета и лаборатории программирования.

Оборудование учебного кабинета:

- Столы ученические двухместные, нерегулируемые;
- Стулья ученические на ножках;
- Доска меловая (магнитно-маркерная);
- Автоматизированное рабочее место преподавателя;
- Комплект учебного наглядного материала по темам;
- Проектор портативный;
- Экран проекционный рулонный.

Оборудование лаборатории и рабочих мест лаборатории:

- Учебная доска;
- Рабочие места по количеству обучающихся;
- Рабочее место преподавателя;
- Персональный компьютер с лицензионным программным обеспечением;
- Проектор и мультимедийная доска;
- Средства аудиовизуализации;
- Наглядные пособия;
- Программное обеспечение общего и профессионального назначения.

4.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд ГАПОУ СО «ЕМК» обладает следующим перечнем используемых учебных изданий, Интернет- ресурсов, дополнительной литературы

Основные источники

1. Голицына, О. Л. Основы алгоритмизации и программирования: учебное пособие / О.Л. Голицына, И.И. Попов. - 4-е изд., испр. и доп. - Москва: ФОРУМ: ИНФРА-М, 2021. - 431 с. -

(Среднее профессиональное образование). - ISBN 978- 5-00091-570-7. - Текст: электронный. - URL: https://znanium.com/ catalog/product/1150328 – Режим доступа: по подписке.

2. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. – 4-е изд. – Москва: Издательство Юрайт, 2023. – 119 с. – (Профессиональное образование). – ISBN 978-5-534-17498-4. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: https://urait.ru/bcode/533200

3. Канцедал, С.А. Алгоритмизация и программирование: учебное пособие / С.А. Канцедал. - Москва: ФОРУМ: ИНФРА-М, 2021. - 352 с. - (Среднее профессиональное образование). - ISBN 978-5-8199-0727-6. - Текст: электронный. URL: https://znanium.com/ catalog/product/1189320. – Режим доступа: по подписке.

4. Колдаев, В.Д. Основы алгоритмизации и программирования: учебное пособие / В.Д. Колдаев; под ред. проф. Л.Г. Гагариной. - Москва: ФОРУМ: ИНФРА-М, 2022. - 414 с. - (Среднее профессиональное образование). - ISBN 978- 5-8199-0733-7. - Текст: электронный.

Основные электронные издания

1. METANIT.COM. Сайт о программировании. Режим доступа: https://metanit.com/

2. Руководство по Python – Документация по Python 3. Режим доступа: https://docs.python.org/3

3. Электронный курс на системе дистанционного обучения ГАПОУ СО «ЕМК»: http://188.234.244.32/course/view.php?id=569

Дополнительные источники

1. Зыкова, Г. В. Основы программирования на языке Python : учебно-методическое пособие / Г. В. Зыкова, А. С. Попов, Т. Н. Сапуглецева ; под. ред. Г. В. Зыковой. - 2-е изд., стер. - Москва : ФЛИНТА, 2020. - 135 с. - ISBN 978-5-9765-4430-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/1860057 (дата обращения: 28.09.2025). – Режим доступа: по подписке.

2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. – Москва : ИНФРА-М, 2025. – 343 с. – (Высшее образование). - ISBN 978-5-16-020255-6. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2166199 (дата обращения: 28.09.2025). – Режим доступа: по подписке.

3. Уроки по языку программирования Python. Режим доступа: https://devpractice.ru/python-lessons

4. "Поколение Python": курс для начинающих. Режим доступа: https://stepik.org/course/58852/syllabus

5. "Поколение Python": курс для продвинутых. Режим доступа: https://stepik.org/course/68343/syllabus

6. Основной учебник Python для новичков | Трепачёв Дмитрий. Режим доступа: https://devpractice.ru/python-lessons/

7. Интерактивный учебник языка Питон. Режим доступа: https://ru.pythontutor.ru

4.3. Общие требования к организации образовательного процесса

Учебная дисциплина предназначена для реализации требований к минимуму содержания и уровню подготовки выпускников по специальности среднего профессионального образования **09.02.07 Информационные системы и программирование; квалификация выпускника – Разработчик веб и мультимедийных приложений.** Учебные занятия должны проводиться в специализированных лабораториях, которые оснащаются современным оборудованием и инструментарием.

Теоретические положения должны подкрепляться практическими занятиями. Учебная практика проводится образовательным учреждением при освоении обучающимися общих компетенций и могут реализовываться как концентрированно в несколько периодов, так и рассредоточено, чередуясь с теоретическими занятиями в рамках учебной дисциплины.

Консультационные мероприятия проходят в обязательном порядке, согласно учебного плана и дополнительно по желанию учащихся.

Освоению данной дисциплины предшествует изучение следующих учебных дисциплин:

- Введение в специальность
- Информатика

4.4. Кадровое обеспечение образовательного процесса

Реализация образовательной программы обеспечивается дипломированными педагогическими кадрами, имеющими высшее профессиональное образование, соответствующее профилю по специальности **09.02.07 Информационные системы и программирование; квалификация выпускника – Разработчик веб и мультимедийных приложений.**

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Результаты обучения	Критерии оценки	Формы и методы оценки
Перечень осваиваемых компетенций в рамках дисциплины: ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ПК 5.4 *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 *Перечень знаний, осваиваемых в рамках дисциплины:* – Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.	Оценка «отлично» - обучающийся показывает полные и глубокие знания программного материала, логично и аргументировано отвечает на поставленный вопрос, а также дополнительные вопросы, показывает высокий уровень теоретических знаний. Практическую часть выполняет на 100%. Оценка «хорошо» - обучающийся показывает глубокие знания программного материала, грамотно его излагает, достаточно полно отвечает на поставленный вопрос и дополнительные вопросы, умело формулирует выводы. В тоже время при ответе допускает несущественные погрешности. Практическую часть выполняет на 90%-80%. Оценка «удовлетворительно» - обучающийся показывает достаточные, но не глубокие знания программного материала; при ответе не допускает грубых ошибок или	**Текущий контроль в форме:** – практических заданий; – задач для самостоятельного выполнения. **Промежуточная аттестация:** экзамен. Оценка: – результативности работы обучающегося при выполнении практических заданий, задач для самостоятельного выполнения; – результата подготовки к экзамену.

– Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы – Подпрограммы, составление библиотек подпрограмм	противоречий, однако в формулировании ответа отсутствует должная связь между анализом, аргументацией и выводами. Для получения правильного ответа требуется уточняющие вопросы. Практическую часть выполняет на 70%- 60%. Оценка «неудовлетворительно» - обучающийся показывает недостаточные знания программного материала, не способен аргументировано и последовательно его излагать, допускается грубые ошибки в ответах, неправильно отвечает на поставленный вопрос или затрудняется с ответом. Практическую часть выполняет на менее 50%	

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ОП. 04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

для специальности

09.02.07 «Информационные системы и программирование»

Екатеринбург

2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данный фонд оценочных средств разработан для сопровождения учебной дисциплины ОП.04 «Основы алгоритмизации и программирования», входящей в общепрофессиональный цикл основной образовательной программы по специальности 09.02.07 «Информационные системы и программирование». Фонд ориентирован на формирование у обучающихся фундаментальных знаний и практических навыков в области алгоритмического мышления, структурного и объектно-ориентированного программирования, а также на развитие профессиональных компетенций, необходимых для последующего освоения профессиональных модулей и успешной профессиональной деятельности в сфере разработки программного обеспечения.

Основной целью фонда оценочных средств является обеспечение системного контроля уровня освоения дисциплины через комплекс теоретических вопросов, практических заданий и критериев оценки, соответствующих требованиям ФГОС СПО. Он призван помочь обучающимся не только усвоить ключевые понятия алгоритмизации и программирования, но и научиться применять их на практике при решении типовых и нетиповых задач.

Структура фонда оценочных средств

Материал фонда оценочных средств структурирован в соответствии с разделами рабочей программы, описание фонда оценочных средств составлено на примере языка Python и включает:

1. **Теоретические задания** – направлены на проверку знаний по основам алгоритмизации, типам данных, управляющим конструкциям, подпрограммам, принципам ООП и этапам разработки приложений.
2. **Практические задания** – охватывают все ключевые темы дисциплины: от построения блок-схем и написания программ линейной, разветвляющейся и циклической структуры до работы со структурированными типами данных, файлами, функциями, рекурсией и разработкой простых настольных приложений с графическим интерфейсом.
3. **Критерии оценки** – обеспечивают объективную и прозрачную систему оценивания, основанную на уровне усвоения знаний, сформированности умений и проявления общих и профессиональных компетенций.

Учебные цели и задачи

Основные учебные цели дисциплины:

- Сформировать у обучающихся системное представление об алгоритмах, их свойствах и способах реализации.
- Обеспечить освоение базовых конструкций языка программирования
- Развить навыки структурного и модульного программирования.
- Подготовить студентов к применению принципов объектно-ориентированного подхода при разработке программных решений.
- Сформировать культуру написания, оформления и отладки программного кода в соответствии с профессиональными стандартами.

Задачи дисциплины:

- Научить разрабатывать и анализировать алгоритмы для решения учебных и прикладных задач.
- Обеспечить умение работать в современных средах программирования.
- Развить способность к самостоятельной отладке и тестированию программ.
- Подготовить обучающихся к последующему освоению профессиональных модулей, связанных с проектированием и разработкой информационных систем.

Контроль знаний

Контроль освоения дисциплины осуществляется на всех этапах обучения:

- Текущий контроль – через выполнение практических работ, решение задач, защиту самостоятельных заданий.
- Рубежный контроль – по завершении ключевых разделов (алгоритмы, основы программирования, структуры данных, подпрограммы).
- Промежуточная аттестация – в форме экзамена, включающего как теоретические вопросы, так и практические задания по реализации алгоритмов и программ.

Оценочная деятельность

Оценка результатов обучения проводится по следующим критериям:

- Полнота и точность усвоения теоретического материала.
- Корректность и эффективность реализации алгоритмов в виде программного кода.
- Соблюдение стандартов оформления и структурирования кода.
- Умение выполнять отладку и тестирование программ.
- Проявление общих (ОК) и профессиональных (ПК) компетенций, включая работу в команде, использование ИКТ и профессиональной документации.

В итоге данный фонд оценочных средств обеспечивает качественное сопровождение образовательного процесса по дисциплине «Основы алгоритмизации и программирования», способствует формированию у обучающихся прочной теоретической базы и практических навыков, необходимых для дальнейшего профессионального роста в сфере информационных технологий.

Краткое изложение планов хода лекционных занятий

1. **Понятие алгоритма и его свойства.** Свойства алгоритмов: дискретность, детерминированность, конечность. Способы представления алгоритмов: словесный, графический, программный. Этапы решения задачи с использованием компьютера. Основные типы алгоритмов: линейные, разветвляющиеся, циклические. *Образовательный результат*: Понимает сущность алгоритмизации, знает свойства алгоритмов и способы их представления, различает основные типы алгоритмов, понимает этапы решения задач с применением ПК. Знает основные характеристики языка Python, умеет работать в среде разработки (IDLE, PyCharm, VS Code), понимает различие между интерпретатором и компилятором. Знает требования PEP 8, умеет писать читаемый, документированный и поддерживаемый код; понимает важность инспекции и рефакторинга.

2. **Понятие переменной и оператор присваивания**. Совместимость и преобразование типов данных (целые, дробные, строки, логический). Математические операции и встроенные функции, генератор случайных чисел. *Образовательный результат*: Знает типы данных и особенности их преобразования, умеет использовать операторы присваивания и математические функции. Знает встроенные типы данных Python (int, float, bool, str, None), понимает организацию программы, правила именования переменных и констант.

3. **Операторы ввода и вывода информации**. Форматный вывод данных и параллельный ввод нескольких переменных. Построение линейных алгоритмов с использованием математических функций. Алгоритмы перестановки чисел и нахождения цифр числа. *Образовательный результат*: Умеет организовать ввод/вывод данных, применять форматный вывод, строить простые алгоритмы с математическими операциями. Использует функции input() и print(), применяет условные конструкции if, elif, else, строит логические выражения с операторами сравнения и and/or/not.

4. **Обработка ошибок ввода-вывода**. Использование блоков и отступов для структурирования кода. *Образовательный результат*: Понимает необходимость обработки исключений, умеет правильно оформлять код с помощью отступов и блоков.

5. **Подпрограммы: функции и процедуры**. Создание, вызов, параметры (формальные, фактические), локальные и глобальные переменные. Возврат значений из функций. Рекурсивные процедуры и функции. *Образовательный результат*: Умеет создавать и вызывать функции и процедуры, понимает использование параметров и область видимости переменных, реализует рекурсию. Создаёт и вызывает функции с параметрами и

возвращаемыми значениями, понимает область видимости переменных (global, local), использует аннотации типов.

6. **Логические переменные**. Логические переменные, выражения и операторы, операции сравнения.

Образовательный результат: Знает логические типы, умеет строить логические выражения и применять операторы сравнения.

7. **Условные конструкции языка**. Условные конструкции языка: неполная и полная формы, вложенные операторы. Алгоритмы нахождения минимума и максимума. *Образовательный результат*: Умеет использовать условные операторы различных форм, строит алгоритмы выбора минимальных и максимальных значений.

8. **Циклические конструкции**. Основные виды циклов, программирование с циклами. Алгоритмы нахождения суммы и произведения. Циклы с предусловием и постусловием. Операторы прерывания цикла. *Образовательный результат*: Реализует циклы различных видов, использует операторы прерывания, строит алгоритмы с повторяющимися вычислениями. Реализует циклы с помощью for и while, использует операторы break и continue, понимает вложенные циклы и их применение.

9. **Строки и операции над ними**. Основные функции и методы для обработки строк, включая поиск и замену символов. *Образовательный результат*: Умеет выполнять операции со строками, применяет встроенные методы для поиска и замены данных в строках. Выполняет обработку строк с использованием методов (split(), join(), replace(), strip() и др.), применяет форматирование строк (f-строки, .format()).

10. **Регулярные выражения**. Правила построения регулярных выражений. *Образовательный результат*: Понимает синтаксис и правила создания регулярных выражений, использует их для эффективного поиска и обработки текстовых данных. Использует модуль re для поиска, замены и валидации текстовых данных с помощью регулярных выражений.

11. **Массивы**. Массивы: создание, ввод, вывод, заполнение случайными числами. Способы перебора элементов и поиск в массиве. *Образовательный результат*: Умеет работать с массивами, вводить и выводить данные, заполнять списки и искать элементы. Создаёт и обрабатывает списки и кортежи, использует срезы, методы списков (append(), remove(), sort() и др.), понимает отличия изменяемых и неизменяемых типов.

12. **Массивы**. Обработка элементов массивов: применение различных способов обработки и преобразования данных.
Образовательный результат: Направленно обрабатывает элементы массива с использованием циклов и функций.

13. **Сортировка одномерных массивов**. Сортировка одномерных массивов: методы пузырька, выбора, подсчёта, сортировка слиянием, быстрая сортировка. Анализ эффективности алгоритмов.
Образовательный результат: Знает основные алгоритмы сортировки, умеет их реализовывать и оценивать эффективность.

14. **Словари**. Словари: создание, операции и методы обработки словарей.
Образовательный результат: Создаёт и модифицирует словари, использует методы для обработки пар ключ-значение; использует словари для хранения и обработки пар «ключ–значение», знает методы keys(), values(), items().

15. **Двумерные структуры данных (матрицы)**. Создание, обработка и вывод данных матриц.
Образовательный результат: Умеет создавать матрицы, обращаться к их элементам и выполнять операции обработки.

16. **Составные структуры данных**. Составные структуры данных и их сортировка по заданному критерию.
Образовательный результат: Понимает работу с составными структурами, реализует сортировку по выбранным ключам.

17. **Множества**. Множества: задание, операции и работа с элементами.
Образовательный результат: Умеет создавать множества, выполнять основные операции над ними (объединение, пересечение, разность). Применяет множества для операций объединения, пересечения и разности

18. **Работа с файлами**. Работа с файлами: чтение и запись информации, обработка файлов различных форматов (csv, xls, doc и др.).
Образовательный результат: Осуществляет работу с файлами разных форматов, использует методы чтения и записи. Осуществляет чтение и запись текстовых и бинарных файлов с использованием контекстного менеджера with, понимает кодировки и обработку ошибок при работе с файлами.

19. **Работа с каталогами и файловой системой**.
Образовательный результат: Управляет файлами и каталогами, выполняет навигацию и операции с файловой системой.

20. **Стек**. Стек: понятие и операции. *Образовательный результат*: Знает структуру данных стек, умеет применять операции push, pop.

21. **Очередь**. Очередь: понятие и операции. *Образовательный результат*: Понимает принцип работы очереди, реализует операции добавления и удаления элементов.

22. **Формат данных JSON**. *Образовательный результат*: Знает структуру JSON, умеет сериализовать и десериализовать данные.

23. **Сериализация и десериализация данных**. *Образовательный результат*: Понимает процессы преобразования объектов в поток данных и обратно, реализует соответствующие операции.

Краткое изложение планов хода практических занятий и примерные задания:

Практическое занятие «Линейные алгоритмы»

- Ход занятия: Анализ задач, написание программ с последовательным выполнением операторов.
- Задание: Написать программу вычисления площади прямоугольника; вычислить периметр треугольника.

Практическое занятие «Алгоритмы ветвления»

- Ход занятия: Изучение условных операторов if, elif, else; построение логики выбора.
- Задание: Определить, является ли число положительным, отрицательным или нулём; реализовать калькулятор; определения чётности числа

Практическое занятие «Циклические алгоритмы»

- Ход занятия: Работа с циклами for и while, вложенными циклами, операторами break и continue.
- Задание: Вывести числа от 1 до N; найти сумму цифр числа; вывести таблицу умножения.

Практическое занятие «Построение линейных алгоритмов. Обработка ошибок ввода-вывода. Знакомство со средой программирования»

- Ход занятия: Написание программ с обработкой исключений, использование try/except, правильное оформление кода. Использование input(), print(), преобразование типов.
- Задание: Создать программу, безопасно принимающую числовой ввод; реализовать вычисления с проверкой ошибок. Вывести на экран информацию о себе (ФИО, группа, дата).

Практическое занятие «Программы линейной структуры с использованием математических функций»

- Ход занятия: Подключение модуля math, использование sqrt(), pow().
- Задание: Вычислить длину гипотенузы и площадь прямоугольного треугольника.

Практическое занятие «Составление программ с разветвляющейся структурой»

- Ход занятия: Использование if, elif, else; вложенные условия.
- Задание:
  - Определить, попадает ли точка в заданную область на плоскости.
  - Создать программу для классификации возраста.
  - Реализовать калькулятор с выбором операции (+, –, *, /).

Практическое занятие «Составление программ с циклической структурой»

- Ход занятия: Работа с for, while, вложенными циклами.
- Задание:
  - Вывести все чётные числа в диапазоне.
  - Найти сумму цифр числа.
  - Вывести таблицу умножения.

Практическое занятие «Создание процедур и функций»

- Ход занятия: Создание процедур и функций, передача параметров, область видимости.
- Задание: Написать процедуру для вывода рамки из символов; функцию для вычисления факториала.

Практическое занятие «Рекурсивные процедуры и функции»

- Ход занятия: Изучение рекурсии, реализация рекурсивных функций.
- Задание: Реализовать рекурсивное вычисление чисел Фибоначчи; вычислить факториал числа.

Практическое занятие «Строки и операции над ними»

- Ход занятия: Применение методов строк, срезов, форматирования.
- Задание: Подсчитать количество слов в предложении; заменить гласные символы на «*».

Практическое занятие «Массивы и операции над ними»

- Ход занятия: Создание списков, перебор элементов, поиск и фильтрация.
- Задание: Создать список из 10 случайных чисел; найти минимальный и максимальный элементы; удалить отрицательные числа.

Практическое занятие «Сортировка массивов»

- Ход занятия: Реализация алгоритмов сортировки пузырьком, выбором, использование встроенных функций.
- Задание: Отсортировать список по возрастанию и убыванию.

Практическое занятие «Словари и операции над ними»

– Ход занятия: Создание и обработка словарей, методы доступа к элементам.
– Задание: Создать словарь «студент – оценка», вывести студентов с оценками ≥ 4; посчитать частоту букв в тексте.

Практическое занятие «Множества и операции над ними»

– Ход занятия: Создание множеств, основные операции (объединение, пересечение).
– Задание: Найти общие элементы двух списков; удалить дубликаты из списка.

Практическое занятие «Двумерные массивы (матрицы) и работа с ними»

– Ход занятия: Создание матриц, обращение к элементам, обработка данных.
– Задание: Вывести матрицу 4×4 случайных чисел; найти сумму главной диагонали.

Практическое занятие «Составные структуры данных»

– Ход занятия: Работа с составными структурами, доступ к вложенным элементам.
– Задание: Создать список словарей и отсортировать по заданному ключу.

Практическое занятие «Сортировка составных структур данных»

– Ход занятия: Сортировка по определённым критериям, использование лямбда-функций.
– Задание: Отсортировать список студентов по фамилии и среднему баллу.

Практическое занятие «Работа с файлами» (чтение и запись в файл)

– Ход занятия: Открытие, чтение и запись файлов с помощью контекстного менеджера.
– Задание: Записать список студентов в файл; прочитать данные из файла и вывести на экран.

Практическое занятие «Работа с файлами и каталогами» (создание, перемещение, удаление)

– Ход занятия: Операции с файлами и каталогами: создание, переименование, удаление.
– Задание: Создать каталог, создать внутри файл, переместить файл, удалить.

Практическое занятие «Стек»

– Ход занятия: Реализация структуры данных стек, операции push, pop.
– Задание: Создать стек, реализовать функцию переворота строки с помощью стека.

Практическое занятие «Очередь»

– Ход занятия: Реализация очереди, добавление и удаление элементов.
– Задание: Создать очередь, моделировать очередь в банке с обслуживанием клиентов.

Практическое занятие «Модульное программирование»

– Ход занятия: Создание и импорт модулей, использование стандартных библиотек.
– Задание: Создать модуль с набором функций и импортировать его в основную программу.

Практическое занятие «Работа с Json»

– Ход занятия: Сериализация и десериализация данных в формате JSON.

– Задание: Сохранить структуру данных в JSON файл и прочитать обратно.

Практическое занятие «Сериализация / десериализация данных»

– Ход занятия: Применение модулей для сериализации объектов (pickle, json).

– Задание: Сериализовать словарь в файл и затем загрузить данные обратно.

Варианты теоретических вопросов к экзамену

1. Дайте определение понятия «алгоритм». Перечислите и поясните его основные свойства.

2. Назовите способы представления алгоритмов. Охарактеризуйте преимущества и недостатки графического способа (блок-схем).

3. Опишите структуру линейного алгоритма. Приведите пример задачи, решаемой с использованием линейной алгоритмической конструкции.

4. Какие виды разветвляющихся алгоритмов существуют? В чём состоит различие между полной и неполной формой ветвления?

5. Поясните сущность циклической алгоритмической структуры. Назовите основные типы циклов.

6. Перечислите этапы решения задачи с использованием компьютера в логической последовательности.

7. Охарактеризуйте графический способ описания алгоритмов. Назовите основные типы блоков, применяемых в блок-схемах.

8. Обоснуйте необходимость соблюдения свойства детерминированности при построении алгоритма.

9. Установите связь между понятиями «алгоритм» и «программа».

10. Приведите пример задачи, решение которой требует использования вложенных циклических конструкций.

11. В чём состоит принципиальное различие между компилятором и интерпретатором? Какой из этих механизмов используется в языке Python?

12. Дайте определение понятия «система программирования». Назовите её основные компоненты.

13. Перечислите основные парадигмы программирования. Какая из них реализована в языке Python?

14. Раскройте различие между понятиями «программа» и «программный продукт».

15. Опишите основные этапы жизненного цикла программного обеспечения.

16. Обоснуйте необходимость учёта области применения при выборе языка программирования.

17. Что представляет собой стандарт языка программирования? С какой целью он разрабатывается?

18. Назовите основные функции современных интегрированных сред разработки (IDE), используемых при изучении дисциплины.

19. Какие особенности языка Python обуславливают его применение в качестве учебного языка программирования?

20. Перечислите основные простые типы данных, поддерживаемые языком Python. Приведите примеры их использования.

21. В чём заключается различие между изменяемыми и неизменяемыми типами данных в Python?

22. Поясните назначение функций input() и print(). Как осуществляется преобразование строкового представления числа в числовой тип?

23. Назовите дополнительные параметры функции print(). Опишите их функциональное назначение.

24. Охарактеризуйте логические операторы and, or, not в языке Python. Приведите примеры их применения.

25. Опишите синтаксис условного оператора if в Python. Как оформляются вложенные условные конструкции?

26. В чём состоит различие между циклами for и while? В каких ситуациях предпочтительно использование каждого из них?

27. Поясните назначение операторов break и continue. Приведите примеры их использования в циклических конструкциях.

28. Назовите основные методы работы со строками в Python. Продемонстрируйте применение методов split(), join(), replace().

29. Что такое срезы в Python? Как с их помощью извлечь подстроку или подсписок?

30. Опишите способы создания и модификации списков в Python: добавление, удаление и изменение элементов.

31. В чём состоит различие между списком и кортежем? В каких случаях целесообразно использование кортежа?

32. Дайте определение понятия «множество» в Python. Назовите основные операции, поддерживаемые над множествами.

33. Охарактеризуйте структуру словаря в Python. Как получить доступ к ключам и значениям словаря?

34. Что такое регулярные выражения? Для решения каких задач они применяются в программировании?

35. Опишите процесс открытия текстового файла для чтения и записи. Обоснуйте необходимость использования конструкции with open(...).

36. Назовите основные режимы открытия файлов (r, w, a, b). Поясните их функциональное различие.

37. Что такое сериализация данных? Как в Python осуществляется сохранение объекта в файл с использованием модуля pickle?

38. Опишите механизм форматирования строк с использованием f-строк. Приведите пример.

39. Что представляет собой специальное значение None в Python? В каких контекстах оно применяется?

40. Как проверить наличие ключа в словаре? Каким образом можно избежать возникновения исключения при обращении к несуществующему ключу?

41. Дайте определение понятия «подпрограмма». В чём состоит различие между процедурой и функцией в Python?

42. Опишите синтаксис объявления функции в Python. Поясните различие между параметрами и аргументами функции.

43. Что такое возвращаемое значение функции? Может ли функция не возвращать значение?

44. Поясните понятие области видимости переменных. В чём состоит различие между локальными и глобальными переменными?

45. С какой целью используется ключевое слово global? Приведите пример его применения.

46. Дайте определение рекурсии. Какие два условия обязательны для корректной реализации рекурсивной функции?

47. Приведите пример рекурсивной функции (например, вычисление факториала). Почему необходимо ограничивать глубину рекурсии?

48. В чём состоит суть структурного подхода к программированию? Назовите три базовые управляющие структуры, лежащие в его основе.

49. Почему необходимо избегать неструктурированного кода с множеством непредсказуемых переходов? Как структурный подход способствует повышению читаемости и сопровождаемости программы?

50. Что такое модуль в Python? Как создать собственный модуль и осуществить его импорт в другую программу?

51. Приведите примеры стандартных модулей Python, используемых при решении учебных задач (например, math, random, os).

52. Обоснуйте необходимость разделения программы на модули. Как это влияет на сопровождаемость и повторное использование кода?

53. Что такое библиотека подпрограмм? Каким образом она может быть использована в различных проектах?

54. Опишите правила документирования функций в Python с использованием docstring. С какой целью это осуществляется?

55. Обоснуйте целесообразность соблюдения принципа «одна функция — одна задача».

56. Что такое множество в Python и какими основными операциями над множествами можно пользоваться?

57. Что представляет собой словарь в Python и чем он отличается от списка?

58. Что такое функция в Python и зачем она нужна?

59. Какие существуют способы перебора элементов списка или словаря?

60. Что такое исключения и зачем их обрабатывать?

61. Какие существуют способы форматирования строк в Python?

62. Что такое аргументы функции и какие их виды бывают?

63. Что такое конкатенация строк?

64. Какие значения считаются ложными (False) в Python?

Варианты практических задач к экзамену

Все задания выполняются на языке программирования Python.

1. Разработать программу, запрашивающую у пользователя два числа и выводящую их сумму, разность, произведение и частное.

2. Реализовать программу, определяющую, является ли введённое целое число чётным или нечётным.

3. Создать программу, находящую максимальное из трёх введённых чисел с использованием условных операторов.

4. Написать программу, выводящую все чётные числа в диапазоне от 1 до 50 с использованием цикла for.

5. Разработать консольную игру «Угадай число»: программа загадывает число от 1 до 100, пользователь вводит свои варианты, получая подсказки «больше» или «меньше».

6. Реализовать функцию, вычисляющую факториал натурального числа с использованием циклической конструкции.

7. Написать рекурсивную функцию для вычисления n-го числа Фибоначчи.

8. Создать список из 10 случайных целых чисел и вывести его на экран.

9. Найти минимальный и максимальный элементы в списке и вывести их значения и индексы.

10. Отсортировать список целых чисел по возрастанию с использованием встроенного метода sort().

11. Создать двумерный список (матрицу 3×3), заполненный нулями, и вывести его в табличном виде.

12. Найти сумму элементов главной диагонали квадратной матрицы.

13. Объединить два списка в одно множество и вывести результат.

14. Создать словарь «студент – оценка» и вывести список студентов, имеющих оценку выше 4.

15. Подсчитать частоту встречаемости каждой буквы в введённой строке с использованием словаря.

16. Разбить введённое предложение на слова и вывести их в обратном порядке.

17. Заменить все гласные буквы в строке на символ «*» с использованием строковых методов.

18. Сохранить введённый пользователем текст в текстовый файл note.txt и прочитать его содержимое обратно.

19. Создать список чисел и определить количество положительных, отрицательных и нулевых элементов.

20. Написать функцию, принимающую список чисел и возвращающую сумму положительных элементов.

21. Создать модуль, содержащий функцию нахождения наибольшего общего делителя (НОД), и продемонстрировать его использование в основной программе.

22. Разработать программу, проверяющую, является ли введённая строка палиндромом (без учёта регистра, пробелов и знаков препинания).

23. Реализовать функцию, удаляющую дубликаты из списка с сохранением порядка первых вхождений элементов.

24. Создать программу, находящую все делители заданного натурального числа.

25. Написать скрипт, считывающий список имён из файла students.txt и выводящий их в алфавитном порядке.

26. Разработать программу «Каталог книг», реализующую добавление книг, их хранение в списке и вывод на экран.

27. Реализовать программу, проверяющую, состоит ли введённая строка только из цифр, и преобразующую её в целое число при успешной проверке.

28. Создать функцию, принимающую строку и возвращающую новую строку, в которой каждое слово начинается с заглавной буквы.

29. Написать программу, подсчитывающую количество положительных, отрицательных и нулевых значений в списке чисел.

30. Реализовать программу, формирующую словарь из двух списков (ключи и значения) с проверкой на равенство их длин.

31. Создать программу, заменяющую в тексте все вхождения заданного слова на другое с использованием строковых методов.

32. Разработать программу, реализующую простой калькулятор с функциями сложения, вычитания, умножения и деления, включая обработку деления на ноль.

Критерии оценивания:

1. **Полнота и точность ответа на теоретические вопросы**

- Теоретические вопросы раскрыты полностью, с приведением корректных определений, пояснений и примеров; продемонстрировано глубокое понимание сущности понятий.
- Ответы в целом соответствуют содержанию вопросов, однако допущены незначительные неточности или отсутствуют отдельные элементы раскрытия темы.
- Теоретические вопросы раскрыты фрагментарно, содержат фактические ошибки или свидетельствуют о недостаточном уровне усвоения материала.

2. **Корректность выполнения практического задания**

- Практическое задание выполнено в полном объёме; реализованное решение соответствует поставленной задаче и работает корректно во всех предусмотренных сценариях.
- Задание выполнено, однако допущены незначительные логические или синтаксические ошибки, не влияющие на основной функционал программы.
- Решение не соответствует условию задания; программа не реализует требуемую функциональность или содержит ошибки, препятствующие её выполнению.

3. **Соответствие техническим требованиям задания**

- В решении использованы все требуемые языковые конструкции (типы данных, управляющие структуры, функции и др.), соблюдены условия формулировки задания.
- Технические требования выполнены частично; отдельные элементы реализованы с отклонениями, не искажающими общую логику решения.
- Требования задания нарушены существенно: отсутствуют обязательные компоненты, использованы не предусмотренные методы или подходы.

4. **Качество оформления и читаемость программного кода**

- Код оформлен в соответствии с принятыми стандартами (PEP 8), структурирован, содержит осмысленные имена переменных и функций, при необходимости снабжён комментариями.
- Оформление кода в целом соответствует требованиям, однако имеются отдельные нарушения стиля или недостаточная ясность именования.
- Код не соответствует стандартам оформления, труден для восприятия, содержит неинформативные идентификаторы, отсутствует логическая структура.

5. **Самостоятельность и оригинальность решения**

- Работа выполнена самостоятельно; решение отражает индивидуальный подход и понимание поставленной задачи.
- При выполнении использованы типовые шаблоны, однако присутствует адаптация под конкретные условия задания.
- Работа содержит признаки заимствования без надлежащего оформления или демонстрирует отсутствие самостоятельного вклада обучающегося.

Балльная шкала оценивания:

- «Отлично» (оценка 5) – Обучающийся демонстрирует полные и глубокие знания по теоретическим вопросам, логично и аргументированно отвечает на поставленные вопросы. Практическое задание выполнено полностью, корректно и в соответствии со всеми требованиями.
- «Хорошо» (оценка 4) – Обучающийся владеет программным материалом, отвечает грамотно и достаточно полно, допуская лишь незначительные неточности. Практическое задание выполнено верно, но содержит отдельные недочёты, не влияющие на общую корректность решения.
- «Удовлетворительно» (оценка 3) – Обучающийся раскрывает теоретические вопросы фрагментарно, с пробелами в знаниях или неточностями, но без грубых ошибок. Практическое задание выполнено частично или с существенными недостатками, нарушающими логику решения.
- «Неудовлетворительно» (оценка 2) – Обучающийся не владеет необходимым объёмом знаний: ответы на теоретические вопросы неполные, противоречивые или содержат грубые ошибки. Практическое задание не выполнено или выполнено с ошибками, делающими решение неработоспособным.