

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«Национальный исследовательский ядерный университет «МИФИ»  
**Новоуральский технологический институт**  
филиал федерального государственного автономного образовательного учреждения высшего  
образования «Национальный исследовательский ядерный университет «МИФИ»  
**(НТИ НИЯУ МИФИ)**  
**Колледж НТИ**

---

Цикловая методическая комиссия информационных технологий

ОДОБРЕНО

Учёным Советом НТИ НИЯУ МИФИ

Протокол № 5 от 02 сентября 2025г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
**УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.03 «ОСНОВЫ**  
**АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

для студентов колледжа НТИ НИЯУ МИФИ,  
обучающихся по программе среднего профессионального образования

специальность 09.02.09

«Веб-разработка»

очная форма обучения

на базе основного общего образования

квалификация

разработчик веб-приложений

Новоуральск 2025

ОДОБРЕНО:  
на заседании  
цикловой методической комиссии  
информационных технологий  
Протокол № 8 от 01.09.2025 г.

Председатель ЦМК ИТ

 И.И. Горницкая

Разработан на основе рабочей  
программы учебной дисциплины  
ОП.03 «Основы алгоритмизации и  
программирования» по специальности  
09.02.09 «Веб-разработка»

Фонд оценочных средств учебной дисциплине ОП.03 «Основы алгоритмизации и программирования» – Новоуральск: Изд-во колледжа НТИ НИЯУ МИФИ, 2025. –18с.

## АННОТАЦИЯ

Фонд оценочных средств предназначен для текущего контроля и промежуточной аттестации обучающихся по специальности 09.02.09 Веб-разработка на соответствие их персональных достижений поэтапным требованиям программы подготовки специалистов среднего звена по учебной дисциплине ОП.03 «Основы алгоритмизации и программирования». Комплектация фонда оценочных средств: паспорт, программа оценивания, оценочные средства для текущего контроля и промежуточной аттестации по учебной дисциплине, критерии оценивания. В паспорте фонда оценочных средств указаны: место дисциплины в структуре программы подготовки специалистов среднего звена, требования ФГОС СПО к результатам освоения, перечень формируемых компетенций, компоненты фонда оценочных средств.

Разработчик: Лебедева А.Н., преподаватель ЦМК информационных технологий.

Редактор: Горницкая И.И.

## СОДЕРЖАНИЕ

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ ОП.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ».....	4
2. ПРОГРАММА ОЦЕНИВАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.03 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» .....	7
3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ .....	8
3.1 Тестовая форма контроля .....	8
3.2 Критерии оценки результатов выполнения тестовых заданий .....	10
3.3 Письменная форма контроля.....	10
3.4. Критерии оценки результатов выполнения письменных заданий.....	12
3.5 Устная форма контроля .....	12
3.6 Критерии оценки результатов выполнения устных заданий .....	14
4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ16	
4.1 Критерии оценивания знаний, обучающихся на семестровом зачете. <b>Ошибка! Закладка не опре</b>	
4.2 Критерии оценивания знаний, обучающихся на экзамене. ....	16

# **1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ ОП.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Фонд оценочных средств является составной частью учебно-методических документов, обеспечивающих реализацию основной профессиональной образовательной программы СПО по специальности 09.02.09 Веб-разработка.

Фонд оценочных средств предназначен для проверки результатов освоения учебной дисциплины ОП.03 «Основы алгоритмизации и программирования».

**Место учебной дисциплины в структуре основной профессиональной образовательной программы:** дисциплина ОП.03 «Основы алгоритмизации и программирования» принадлежит к общепрофессиональному циклу.

## **Цель и планируемые результаты освоения учебной дисциплины:**

В результате изучения учебной дисциплины студент должен:

### **Уметь:**

- 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.

### **Знать:**

- 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.

В результате изучения дисциплины студент должен освоить профессиональные и общие компетенции:

Перечень формируемых компетенций в соответствии с требованиями ФГОС СПО:

Общие компетенции (ОК)

<b>Код</b>	<b>Наименование общих компетенций</b>
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 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях

## Профессиональные компетенции (ПК)

<b>Код</b>	<b>Наименование профессиональных компетенций</b>
ПК 1.1	Проектировать информационные ресурсы
ПК 1.3	Интегрировать программный код в соответствующую инфраструктуру
ПК 1.5	Выполнять процедуры тестирования программного кода
ПК 2.2	Проводить работы по резервному копированию и развертыванию резервной копии информационных ресурсов
ПК 2.3	Настраивать права пользователей в соответствии с функциональными задачами (ролями) и на основании информации о поведенческих факторах
ПК 2.4	Применять программные средства обеспечения безопасности информации веб-приложений

**2. ПРОГРАММА ОЦЕНИВАНИЯ  
УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.03 «ОСНОВЫ  
АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

<b>№ п/п</b>	<b>Контролируе мые элементы учебной дисциплины</b>	<b>Контролируемые компетенции</b>	<b>Форма аттестации</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
1	Темы 1.1-2.4	ОК 01, ОК 02, ОК 05, ОК 07, ПК 1.1, ПК 1.3, ПК 1.5, ПК 2.2, ПК 2.3, ПК 2.4.	тестовые задания письменные задания устный опрос
2	Темы 2.5-2.8		IV семестр экзамен

### 3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

Текущий контроль знаний, умений и навыков осуществляется в ходе образовательного процесса по всем изучаемым междисциплинарным курсам в формах, определяемых преподавателем в рабочей программе учебной дисциплины. Он может проводиться в ходе всех видов занятий в форме, избранной преподавателем.

Текущий контроль успеваемости может проводиться в следующих формах:

- устная (устный опрос, защита письменной работы, доклад или реферат по результатам самостоятельной работы и т. д.);
- письменная (письменный опрос, выполнение контрольных и других проверочных работ и т. д.);
- тестовая (устное, письменное, компьютерное тестирование).

Возможны и другие виды текущего контроля успеваемости, которые определяются преподавателями.

<b>ФОРМА КОНТРОЛЯ</b>	<b>тестовая</b>	<b>письменная</b>	<b>устная</b>
<b>ТЕМЫ УЧЕБНОЙ ДИСЦИПЛИНЫ</b>	тема 1.1-2.2 2.5-2.8	темы 1.1-2.8	темы 1.1-2.8

Система текущего контроля успеваемости предусматривает разнообразные по форме и содержанию контрольные мероприятия, учитывающие все виды аудиторной и самостоятельной учебной деятельности обучающегося.

Данные текущего контроля используются преподавателями для анализа (мониторинга) качества освоения обучающимися основной образовательной программы среднего образования, для организации консультаций.

#### 3.1 Тестовая форма контроля

Тестовая форма контроля осуществляется посредством использования электронной образовательной платформы Юрайт – Академия.

Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 227 с. — (Профессиональное образование). — ISBN 978-5-534-17319-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. с. 174 — URL: <https://urait.ru/bcode/539652/p.174>.

**Тема 1.1. Алгоритмизация**

<https://urait.ru/quiz/run-test/BADAC700-F311-40AD-A385-FA2075522733/B044CE82-28E6-450A-932D-3166BD589348>

**Тема 1.2. Основы технологии программирования**

<https://urait.ru/quiz/run-test/9CE29268-ED4A-4B52-B6DB-F4549FACD2B1/CA6024D4-ECE0-4B1F-929C-AC6DFC1E4DA3>

**Тема 2.1. Алфавит языка программирования. Типы данных**

<https://urait.ru/quiz/run-test/8CFE1E30-9C59-4911-B732-CB9C700DB112/4924975E-535E-44B9-ADDE-55E1BE013D47>

**Тема 2.2. Операторы языка**

<https://urait.ru/quiz/run-test/C9C4DB51-B9C2-463D-988C-1FDB576B95E5/8792A9E2-4194-49DD-87DF-87F8354710BE>  
<https://urait.ru/quiz/run-test/F544155E-B537-4BE1-B27D-D550F7A9AB03/467E9A98-9BEE-4703-A549-7968106F9C9F>

**Тема 2.5. Пользовательские типы данных**

<https://urait.ru/quiz/run-test/A1C48662-B22F-4B93-B641-2B7ABD63954F/ED760669-5483-4829-B94B-F0C55FE9BAC5>

**Тема 2.6. Функции**

<https://urait.ru/quiz/run-test/0168C8B8-1630-4439-A86D-3F700C57C839/25E8E2C1-6A66-43F8-B197-DF888584364E>

**Тема 2.7. Работа с файлами**

<https://urait.ru/quiz/run-test/DD5623BC-01C9-404D-91D1-801ECD3B852C/69E9283C-5499-49A7-A41D-B268CBD37DC5>

**Тема 2.8. Динамические структуры данных**

<https://urait.ru/quiz/run-final-test/C7CE975C-8593-4F47-965C-6B34D1C57A67>

### 3.2 Критерии оценки результатов выполнения тестовых заданий

В качестве критерия освоения материала учебной дисциплины берётся выполнение 50% заданий от их общего числа в тест-задании.

Для интерпретации результатов тестирования используется переводная шкала, с помощью которой результаты теста преобразуют в оценку знаний студентов по пятибалльной системе.

Переводная шкала для интерпретации результатов выполнения тестов

<b>% ошибочных ответов</b>	<b>Интерпретация результатов тестирования по 5-ой системе</b>
до 10%	Отлично
от 15% до 30%	Хорошо
от 35% до 50%	Удовлетворительно
более 50%	Неудовлетворительно

### 3.3 Письменная форма контроля

<b>ТЕМА</b>	<b>НАЗВАНИЕ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ РАБОТ</b>	<b>САМОСТОЯТЕЛЬНАЯ РАБОТА</b>
<b>Тема 1.1. Алгоритмизация</b>	Практическое занятие №1. Разработка алгоритмов линейной, разветвляющейся и циклической структуры	Самостоятельная работа №1. Реферат « Основы алгоритмизации и технологии программирования»
<b>Тема 1.2. Основы технологии программирования</b>		
<b>Тема 2.1. Алфавит языка программирования. Типы данных</b>		Самостоятельная работа №2. Реферат «Основы программирования»
<b>Тема 2.2. Операторы языка</b>	Практическое занятие № 2. Разработка программ линейной структуры Практическое занятие № 3. Разработка программ разветвляющейся структуры Практическое занятие № 4. Разработка программ циклической структуры	Самостоятельная работа №3. Тест №1. Основы алгоритмизации и программирования
<b>Тема 2.3. Массивы</b>	Практическое занятие № 5. Разработка программ с использованием одномерных	Самостоятельная работа №4. Тест №2. Основы программирования

	<p>массивов</p> <p>Практическое занятие № 6. Разработка программ с использованием двумерных массивов</p>	
<b>Тема 2.4. Строки</b>	<p>Практическое занятие № 7. Разработка программ с использованием стандартных функций для работы со строками и массивами</p>	-
<b>Тема 2.5. Пользовательские типы данных</b>	<p>Практическое занятие № 8. Разработка программ с использованием пользовательских типов данных</p>	-
<b>Тема 2.6. Функции</b>	<p>Практическое занятие № 9. Разработка функций с использованием одномерных массивов</p> <p>Практическое занятие № 10. Разработка функций с использованием двумерных массивов</p> <p>Практическое занятие № 11. Разработка программ с использованием рекурсии</p> <p>Практическое занятие № 12. Разработка функций с использованием данных строкового типа</p>	-
<b>Тема 2.7. Работа с файлами</b>	<p>Практическое занятие № 13. Создание файла. Чтение из файла. Изменение данных в файле</p>	
<b>Тема 2.8. Динамические структуры данных</b>	<p>Практическое занятие № 14. Разработка программ с использованием однонаправленных списков типа «стек»</p> <p>Практическое занятие № 15. Разработка программ с использованием однонаправленных списков типа «очередь».</p> <p>Практическое занятие № 16. Разработка программ с использованием двусвязных списков</p> <p>Практическое занятие № 17. Разработка алгоритмов с использованием динамических структур данных: очередей и стеков.</p>	

### **3.4. Критерии оценки результатов выполнения письменных заданий**

Оценка «5» (отлично) – краткая, ясная и четкая описание решения задания, верно, на высоком профессиональном уровне, с правильными пояснениями. Обоснованность ответа. Необходимо мотивировать выбранный курс действий, приводящих к разрешению ситуации, и объяснить причины и рациональность его выбора.

Оценка «4» (хорошо) – задание выполнено в целом верно, но с небольшими неточностями, имеются неточности в пояснении.

Оценка «3» (удовлетворительно) – задание выполнено верно, но не доведен до конца, либо в нем имеются ошибки, которые, однако, не приводят к принципиально неверному решению.

Оценка «2» (неудовлетворительно) – задание не выполнено или выполнено не верно

### **3.5 Устная форма контроля**

**Примерный список вопросов для устного ответа по учебной дисциплине.**

1. Опишите сильные и слабые стороны языка программирования Python. Ответ обоснуйте
2. Расскажите, какие цели преследовал Гвидо в процессе разработки языка Python?
3. Перечислите основные этапы разработки программ.
4. Перечислите основные области применения язык Python. Приведите примеры программ.
5. Опишите схему запуска программы на языке Python.
6. Опишите правила наименования переменных в Python.
7. Дайте развернутое описание модели памяти Python при работе с переменными.
8. Опишите процесс создания функций в Python.
9. Объясните, в чем состоят различия между выполнением команд из файла и в интерактивном режиме.

10. Раскройте смысл понятия области видимости переменных, опишите возможности создания локальных и глобальных переменных в Python, ключевые слова, определяющие область видимости переменных, .
11. Опишите схему передачи чисел в качестве аргументов функции.
12. Перечислите операции над строками, существующие в языке Python, опишите их назначение. Приведите примеры escape-последовательностей.
13. Опишите операторы отношений, существующие в Python. Перечислите правила логических операций над объектами.
14. Приведите примеры случаев, когда применяется условная инструкция if.
15. Приведите пример «оператора моржа».
16. Опишите, как формируются строки документации.
17. Объясните, что такое модуль в Python. Приведите пример.
18. Объясните, что такое пакет в Python. Приведите пример.
19. Опишите процесс создания собственных модулей в Python.
20. Объясните, что такое разработка через тестирование. Приведите пример.
21. Опишите алгоритм генерации случайных чисел в Python.
22. Опишите строковые методы, существующие в Python. Объясните отличия методов от функций.
23. Опишите процесс создания списка. Перечислите основные операции над списками в Python.
24. Приведите примеры преобразования типов в Python (списки, строки).
25. Охарактеризуйте виды циклов, существующие в Python. Приведите примеры применения циклов for, while.
26. Опишите возможности применения вложенных списков в Python.
27. Объясните, что такое множество. Опишите операции над множествами, существующие в Python.

28. Объясните, что такое кортеж. Опишите операции над кортежами, существующие в Python.
29. Объясните, что такое словарь. Опишите операции над словарями, существующие в Python.
30. Объясните, что такое стек и как работает стековая машина языка Python.
31. Расскажите, как происходит обработка исключений в Python.
32. Объясните, как работает менеджер контекста при обработке файлов.
33. Опишите операции над байтовым типом данных в Python.
34. Объясните, что такое регулярные выражения, приведите примеры их применения.
35. Опишите преимущества использования формата JSON.
36. Опишите особенности объектно-ориентированного программирования в Python. Что такое классы, объекты?
37. Дайте определение наследования в ООП. Приведите примеры его использования на Python.
38. Объясните, что такое итераторы и приведите пример создания итератора.
39. Опишите структуру оконного приложения на примере модуля tkinter.
40. Расскажите, что такое генератор и как он связан с итератором.
41. Объясните, что такое декоратор, и приведите примеры декораторов.
42. Опишите работу шаблона «модель — вид — контроллер» (на примере модуля tkinter).

### **3.6 Критерии оценки результатов выполнения устных заданий**

Оценка «5» (отлично) предполагает грамотное и логичное изложение ответа в устной форме, на практикоориентированные вопросы; обоснование собственного ответа с точки зрения известных теоретических положений.

Оценка «4» (хорошо) – если студент полно освоил учебный материал, владеет научнопонятийным аппаратом, ориентируется в изученном материале, осознанно применяет теоретические знания на практике, грамотно излагает ответ (в устной или письменной форме), но содержание и форма ответа имеют отдельные неточности.

Оценка «3» (удовлетворительно) – если студент обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, непоследовательно, допускает неточности в определении понятий, в применении теоретических знаний при ответе на практико-ориентированные вопросы;

Оценка «2» (неудовлетворительно) – если студент имеет разрозненные, бессистемные знания, допускает ошибки в определении базовых понятий, искажает их смысл; не может практически применять теоретические знания.

## **4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

Промежуточная аттестация по учебной дисциплине IV семестр – экзамен.

### **4.1 Критерии оценивания знаний, обучающихся на экзамене.**

В результате аттестации дисциплине комплексная проверка профессиональных и общих компетенций осуществляется в форме оценки качества выполнения заданий на экзамене:

«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 учебные задания содержат грубые ошибки.

### **Пример типового экзаменационного задания:**

#### **Задание 1.**

Постановка задачи: Оценки каждого из 12 учеников по трем предметам представлены в виде таблицы:

Ученик	Предмет		
	1	2	3
1	4	4	5
2	3	4	3
–			
12	5	4	4

Необходимо ввести в программу каждую из оценок и найти их сумму.

Задачу решить в двух вариантах:

- 1) ввод оценок осуществляется по строкам;
- 2) ввод оценок осуществляется по столбцам

Необходимо представить на проверку:

Блок-схему программы

Текст программы должен содержать комментарии.

## Задание 2

Для разработанной программы создать:

сценарий и набор для тестирования;

провести анализ эффективности разработанного кода.

### Варианты постановок задач для разработки программ:

1. Разработать программу для игры. «Описание игры. На экране появляется вопрос: «Чет (введите 2) или нечет (введите 1)?» Играющий должен ответить, какое число – четное или нечетное – выберет компьютер, и ввести соответственно 2 или 1. После этого компьютер случайным образом генерирует одно из них. Результат сравнения ответа играющего с числом компьютера выводится на экран.»

2. Разработать игру, моделирующую бросание игрового кубика каждым из двух участников, после чего определяется у кого, выпало больше очков.

3. Разработать игру «Описание игры. Компьютер генерирует случайное целое число, большее 0 и меньше 101. Играющий пытается отгадать это число, делая несколько попыток. В случае несовпадения «задуманного» компьютером числа и числа-ответа на экран выводится

сообщение о том, какое из них больше, после чего играющий вновь вводит число, и т. д. до отгадывания»

4. Разработать игру «Игра моделирует выбор каждым из двух играющих «наугад» по од-ной карте из полного набора игральных карт, включающего четыремасти («пики», «трефы», «бубны» и «червы») и по 9 достоинств картв каждой масти («шестерка», «семерка», «восьмерка», «девятка», «десятка», «валет», «дама», «король», «туз»), и определение того изучастников игры, у которого выбранная карта «старше». При этом условимся, что приведенный выше перечень мастей и карт одной масти дан в порядке увеличения их «старшинства» (например, любая картамасти «бубны» старше любой карты масти «пики», а «валет червей» старше «десятки червей»))»

5. Разработать игру «Описание игры. На столе выложены  $n$  предметов (спичек, монет, ка-мешков и т. п.). Играют двое. Они поочередно забирают несколько предметов, причем заранее договорено, что число забранных предметов не превышает  $k$  ( $1 < k \leq n$ ). Проигрывает тот, кто своим ходом вынужден забрать последний предмет»