

РОСЖЕЛДОР

**Федеральное государственное бюджетное
образовательное учреждение высшего образования
"Ростовский государственный университет путей сообщения"
(ФГБОУ ВО РГУПС)**

УТВЕРЖДАЮ
Первый проректор М.А. Кравченко

Кафедра "Информатика"

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ (МЕЖДИСЦИПЛИНАРНОМУ КУРСУ)**

ОП.03 «Основы алгоритмизации и программирования»

по Учебному плану

специальности среднего профессионального образования
09.02.09 Веб-разработка

Квалификация специалиста среднего звена "Разработчик веб-приложений"

Ростов-на-Дону
2025

Содержание

1. Результаты обучения дисциплины (модуля)	3
2. Компетенции, формируемые в процессе изучения дисциплины (модуля)	5
3. Оценочные средства для оценки успеваемости студентов	6
4. Шкала оценивания в зависимости от уровня сформированности компетенций	8

1. Результаты обучения дисциплины (модуля)

В результате освоения дисциплины (модуля) обучающийся должен овладеть следующими результатами:

Код и наименование компетенции выпускника	Формулировка требований к степени сформированности компетенции
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.	Умения:
	Анализировать постановку вычислительной задачи и определять наиболее подходящий тип алгоритма;
	Выбирать оптимальные структуры данных для хранения и обработки информации в зависимости от условий конкретной задачи;
	Сравнивать эффективность различных алгоритмов решения одной задачи и обосновывать выбор;
	Выбирать парадигму программирования в зависимости от сложности и масштаба разрабатываемой программы.
	Знания:
	Классификации алгоритмов и их свойств;
	Базовых принципов оценки сложности алгоритмов;
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.	Умения:
	Использовать официальную документацию Python, справочные системы IDE и ресурсы для поиска информации о синтаксисе, стандартных библиотеках и методах решения типовых проблем программирования;
	Применять интегрированные среды разработки со всеми их инструментами для повышения эффективности разработки;
	Анализировать и интерпретировать сообщения об ошибках для локализации и устранения дефектов в коде.
	Знания:
	Структуры и правил работы с официальной технической документацией по языку программирования;
	Основных возможностей современных сред разработки для python;
	Базовых принципов визуализации данных в контексте программной

	разработки.
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.	Умения:
	<p>Читать и понимать фрагменты технических заданий, описаний алгоритмов и спецификаций на русском языке для корректной реализации программы;</p> <p>Использовать англоязычную документацию к стандартным библиотекам python;</p> <p>Понимать ключевую профессиональную лексику на английском языке, сообщениях об ошибках, названиях функций и методов.</p>
	Знания:
	<p>Базовой профессиональной терминологии в области алгоритмизации и программирования на русском и английском языках;</p> <p>Структуры и условных обозначений, принятых в технической документации по программированию.</p>
ПК 1.1. Проектировать информационные ресурсы.	Умения:
	<p>Анализировать постановку задачи и выделять основные сущности и процессы предметной области;</p> <p>Разрабатывать алгоритмы решения типовых вычислительных задач с использованием основных алгоритмических конструкций;</p> <p>Выбирать оптимальные структуры данных для хранения и обработки информации в соответствии с условиями задачи;</p> <p>Проектировать структуру программы, выделяя логические блоки и определяя взаимосвязи между ними.</p>
	Знания:
	<p>Основных свойств алгоритмов (дискретность, понятность, определенность, результативность, массовость) и способов их описания;</p> <p>Основных алгоритмических конструкций и принципов структурного программирования;</p> <p>Характеристик, областей применения и ограничений базовых структур данных языка python;</p> <p>Этапов жизненного цикла программного обеспечения.</p>
ПК 1.2. Разрабатывать интерфейсы пользователя.	Умения:
	<p>Организовывать консольный ввод данных от пользователя с использованием функции input() и преобразованием типов;</p> <p>Форматировать вывод результатов работы программы для удобства восприятия;</p>

	<p>Реализовывать простое интерактивное консольное меню выбора действий;</p> <p>Обеспечивать «дружелюбность» интерфейса путем обработки некорректного ввода пользователя с использованием механизма исключений;</p>
	Знания:
	<p>Принципов организации ввода-вывода в консольных приложениях;</p> <p>Методов форматирования строк в python;</p> <p>Базовых принципов создания удобного интерфейса и обработки ошибок ввода.</p>
<p>ПК 1.3. Интегрировать программный код в соответствующую инфраструктуру.</p>	Умения:
	<p>Создавать пользовательские функции и классы для организации многофайлового проекта;</p> <p>Импортировать и использовать собственные и стандартные модули python;</p> <p>Интегрировать отдельные программные модули в единую исполняемую программу;</p> <p>Организовывать чтение исходных данных из внешних файлов и запись итоговых результатов в файлы;</p> <p>Использовать виртуальные окружения для управления зависимостями проекта.</p>
	Знания:
	<p>Принципов модульности и повторного использования кода;</p> <p>Синтаксиса и семантики оператора импорта модулей в python;</p> <p>Основных форматов файлов для обмена данными и способов работы с ними;</p> <p>Назначения и основ использования виртуальных окружений.</p>
<p>ПК 1.4. Использовать систему контроля версий в процессе коллективной (параллельной) разработки.</p>	Умения:
	<p>Создавать локальный и удаленный репозиторий для учебного программного проекта;</p> <p>Выполнять основные операции с репозиторием;</p> <p>Организовывать работу с удаленным репозиторием;</p> <p>Выполнять слияние веток и разрешать простые конфликты слияния;</p> <p>Анализировать историю коммитов.</p>
	Знания:
	Назначения, преимущества и базовые принципы работы распределенных

	<p>систем контроля версий;</p> <p>Инициировать репозиторий для учебного проекта и подключать удаленный репозиторий;</p> <p>Корректно фиксировать изменения в коде с информативными комментариями;</p> <p>Создавать отдельные ветки для разработки новых функций и сливать их с основной веткой.</p>
ПК 1.5. Выполнять процедуры тестирования программного кода.	Умения:
	<p>Идентифицировать типы ошибок в программе;</p> <p>Применять основные методы отладки;</p> <p>Разрабатывать и применять простые тестовые наборы данных;</p> <p>Проводить ручное функциональное тестирование программы на соответствие техническому заданию.</p>
	Знания:
	<p>Классификации ошибок программирования;</p> <p>Основных стратегий и методов тестирования программного обеспечения (отладка, модульное тестирование);</p> <p>Основных типов исключений в python и принципов их обработки.</p>

2. Компетенции, формируемые в процессе изучения дисциплины (модуля)

Индекс и Наименование компетенции	Признаки проявления компетенции в соответствии с уровнем формирования в процессе освоения дисциплины
ПК 1.1. Проектировать информационные ресурсы.	<p>недостаточный уровень: Компетенции не сформированы. Знания отсутствуют, умения и навыки не сформированы.</p> <p>пороговый уровень: Компетенции сформированы. Сформированы базовые структуры знаний. Умения фрагментарны и носят репродуктивный характер. Демонстрируется низкий уровень самостоятельности практического навыка.</p> <p>продвинутый уровень: Компетенции сформированы. Знания обширные, системные. Умения носят репродуктивный характер, применяются к решению типовых заданий. Демонстрируется достаточный уровень самостоятельности устойчивого практического навыка.</p> <p>высокий уровень: Компетенции сформированы.</p>
ПК 1.2. Разрабатывать интерфейсы пользователя.	
ПК 1.3. Интегрировать программный код в соответствующую инфраструктуру.	
ПК 1.4. Использовать систему контроля версий в процессе коллективной (параллельной) разработки	
ПК 1.5. Выполнять процедуры	

тестирования программного кода.	Знания твердые, аргументированные, всесторонние. Умения успешно применяются к решению как типовых, так и нестандартных творческих заданий. Демонстрируется высокий уровень самостоятельности, высокая адаптивность практического навыка.
---------------------------------	--

3. Оценочные средства для оценки успеваемости студентов

1

!*ВВ

Какое свойство алгоритма гарантирует, что он завершится за конечное число шагов?

!*-

Детерминированность

!*-

Массовость

!*+

Конечность

!*-

Результативность

2

!*ВВ

Какой тип цикла в Python оптимален, когда известно точное количество повторений?

!*-

while

!*-

do...while

!*+

for

!*-

repeat...until

3

!*ВК

Как называется функция, вызывающая саму себя? - ...

!*-

рекурсивная

4

!*ВК

Структура данных, работающая по принципу «последним пришёл — первым вышел»? – это ...

!*-

стек

5

!*ВК

Какой командой можно загрузить и установить библиотеку в среду выполнения Python ...

!*-

pip

6

!*ВК

Какой тип данных возвращает команда input() ...

!*
строковый

7

!*ВК

С какого номера начинается индекс списка в Python...

!*
0

8

!*ВК

Если не узнавать длину списка, по какому индексу можно адресовать последний элемент...

!*
-1

9

!*ВВ

Какой тип данных в Python является неизменяемым?

!*+
Кортеж

!*-
Список

!*-
Словарь

!*+
Строка

10

!*ВВ

Чтобы создать кортеж нужно использовать символы ...

!*+
()

!*-
[]

!*-
{ }

!*-
" "

11

!*ВВ

12

!*ВВ

13

!*ВВ

14

!*ВВ

15

!*ВВ

16

!*ВВ

17

!*ВВ

18

!*ВВ

19

!*ВВ

20

!*ВВ

21

!*ВВ

22

!*ВВ

23

!*ВВ

Перечень вопросов для устного опроса:

- 1) Основные понятия и определения в области алгоритмизации и программирования.
- 2) Приемы описания алгоритмов и этапы разработки программ.
- 3) Приемы использования базовых конструкций языка Python.
- 4) Принципы организации программы с использованием функций и модулей.
- 5) Принципы анализа и отладки программного кода.

Перечень вопросов для самоподготовки:

- 1) Особенности работы со средами разработки для Python.
- 2) Особенности реализации алгоритмов сортировки и поиска.
- 3) Сложность алгоритмов: временная и пространственная
- 4) Администрирование структур данных.
- 5) Классификация ошибок в программах и методы их устранения.

Перечни сопоставленных с ожидаемыми результатами освоения дисциплины вопросов (задач):

Зачет. Семестр. № 3

Вопросы для оценки результата освоения "Знать":

- 1) Основные свойства алгоритмов и способы их описания. Сравнение типов алгоритмов: линейных, разветвляющихся, циклических, рекурсивных.
- 2) Основные алгоритмические конструкции и их графическое представление в виде блок-схем.
- 3) Принципы структурного, модульного и объектно-ориентированного программирования и их отличия.
- 4) Правила именования идентификаторов в Python. Ключевые слова языка и основные типы данных.
- 5) Классификация операторов Python: арифметические, сравнения, логические, присваивания, ввода-вывода.
- 6) Синтаксис и семантика условных операторов и операторов циклов. Понятие вложенных циклов.
- 7) Понятие структурированного типа данных. Организация работы с одномерными и двумерными массивами.
- 8) Характеристики и области применения основных структур данных Python: списков, кортежей, множеств и словарей.
- 9) Методы работы со строками. Базовые понятия регулярных выражений.

Вопросы для оценки результата освоения "Уметь":

- 1) Анализировать условие задачи и разрабатывать для нее алгоритм в виде блок-схемы или словесного описания.
- 2) Устанавливать интерпретатор Python и настраивать среду разработки для выполнения учебных проектов.
- 3) Разрабатывать программы линейной структуры с использованием операторов ввода, присваивания и вывода.
- 4) Разрабатывать программы разветвляющейся структуры для реализации логики принятия решений.
- 5) Разрабатывать программы циклической структуры для обработки последовательностей данных и реализации итеративных алгоритмов.
- 6) Создавать и обрабатывать одномерные и двумерные массивы (списки): выполнять поиск, сортировку, нахождение суммы, минимума/максимума.
- 7) Применять различные структуры данных для эффективного решения типовых задач.
- 8) Выполнять обработку строк: поиск и замена подстрок, работа с регулярными выражениями для валидации данных.

Экзамен. Семестр № 4

Вопросы для оценки результата освоения "Знать":

- 1) Понятие пользовательского типа данных. Основные действия: создание, инкапсуляция данных, определение методов.
- 2) Сущность и назначение функции в программе. Принципы организации кода с использованием функций.
- 3) Способы передачи аргументов в пользовательские. Различия между изменяемыми и неизменяемыми аргументами.
- 4) Понятие рекурсии. Условия корректного завершения рекурсивного алгоритма. Примеры классических рекурсивных задач.
- 5) Типы файлов. Основные операции при работе с файлами.
- 6) Принципы организации и основные операции для абстрактных типов данных.
- 7) Основные принципы объектно-ориентированного программирования.

Вопросы для оценки результата освоения "Уметь":

- 1) Создавать пользовательские типы данных (классы) для описания сущностей предметной области, инкапсулируя данные и методы работы с ними.
- 2) Разделять программу на логические блоки, оформляя их в виде отдельных функций с четко определенным интерфейсом.
- 3) Разрабатывать функции с передачей аргументов различных типов, в том числе произвольного количества.
- 4) Разрабатывать и применять рекурсивные алгоритмы для решения задач.
- 5) Организовывать работу с файлами различных форматов: чтение данных, их обработка и запись результатов.
- 6) Реализовывать базовые операции над динамическими структурами данных.
- 7) Реализовывать алгоритмы работы с бинарными деревьями.
- 8) Создавать иерархии классов, используя механизм наследования для расширения и специализации поведения.
- 9) Применять полиморфизм через переопределение методов и создание функций, работающих с объектами разных классов.
- 10) Выполнять процедуры тестирования и отладки программного кода: выявлять синтаксические и логические ошибки, использовать отладчик и инструменты трассировки.
- 11) Интегрировать отдельные программные модули (функции, классы) в единую работающую программу, решающую учебную задачу.

4. Шкала оценивания в зависимости от уровня сформированности компетенций

Описание шкал оценивания компетенций

Значение оценки	Уровень освоения компетенции	Шкала оценивания (для аттестационной ведомости, зачетной книжки, документа об образовании)	Шкала оценивания (процент верных при проведении тестирования)
Балльная оценка - "удовлетворительно".	Пороговый	Оценка «удовлетворительно» выставляется обучающемуся, который имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения последовательности изложения программного материала и испытывает трудности в выполнении практических навыков.	От 40% до 59%
Балльная оценка - "хорошо".	Базовый	Оценка «хорошо» выставляется обучающемуся, твердо знающему программный материал, грамотно и по существу его излагающему, который не допускает существенных неточностей в ответе, правильно применяет теоретические положения при решении практических работ и задач, владеет необходимыми навыками и приемами их выполнения.	От 60% до 84%
Балльная оценка - "отлично".	Высокий	Оценка «отлично» выставляется обучающемуся, глубоко и прочно усвоившему программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагающему, в ответе которого тесно увязываются теория с практикой. При этом обучающийся не затрудняется с ответом при видоизменении задания, показывает знакомство с литературой, правильно	От 85% до 100%

Значение оценки	Уровень освоения компетенции	Шкала оценивания (для аттестационной ведомости, зачетной книжки, документа об образовании)	Шкала оценивания (процент верных при проведении тестирования)
		обосновывает ответ, владеет разносторонними навыками и приемами практического выполнения практических работ.	
Дуальная оценка - "зачтено".	Пороговый, Базовый, Высокий	Оценка « зачтено » выставляется обучающемуся, который имеет знания, умения и навыки, не ниже знания только основного материала, может не освоить его детали, допускать неточности, недостаточно правильные формулировки, нарушения последовательности изложения программного материала и испытывает трудности в выполнении практических навыков.	От 40% до 100%
Балльная оценка - "неудовлетворительно", Дуальная оценка - "не зачтено".	Не достигнут	Оценка « неудовлетворительно, не зачтено » выставляется обучающемуся, который не знает значительной части программного материала, допускает ошибки, неуверенно выполняет или не выполняет практические работы.	От 0% до 39%