ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Микро- и наноэлектроника»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

Б1.В.01 «Практика программирования с использованием Python»

Направление подготовки

03.03.01 «Прикладные математика и физика»

Направленность (профиль) подготовки

Электроника, квантовые системы и нанотехнологии

Уровень подготовки

Академический бакалавриат

Квалификация выпускника – бакалавр

Формы обучения – очная

Рязань, 2026 г.

1. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Оценочные материалы – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общекультурных и профессиональных компетенций, приобретаемых обучающимся в соответствии с этими требованиями.

ПК-1.1: Проводит моделирование и исследования функциональных, статических, динамических, временных, частотных характеристик приборов, схем, устройств и установок электроники и наноэлектроники различного функционального назначения;

ПК-1.2: Изучает стандарты, методы и методики исследования и контроля параметров электронных материалов и структур, технологических операций электроники и наноэлектроники.

Контроль знаний обучающихся проводится в форме текущего контроля и промежуточной аттестации.

При оценивании (определении) результатов освоения дисциплины применяется традиционная система (зачтено, незачтено).

Паспорт фонда оценочных средств по дисциплине

№ п/п	№ раздела	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Этап формирования контролируемой компетенции (или её части)	Вид, метод, форма оценочного средства
1	1	Знакомство с Python	ПК-1.1, ПК-1.2	Лекционные, лабораторные и самостоятельные занятия обучающихся в течение учебного семестра	Аналитический отчет, зачет
2	2	Ввод-вывод данных	ПК-1.1, ПК-1.2	Лекционные, лабораторные и самостоятельные занятия обучающихся в течение учебного семестра	Аналитический отчет, отчеты по лабораторным работам, зачет
3	3	Выбор из двух. Логические операции. Вложенные и каскадные условия	ПК-1.1, ПК-1.2	Лекционные, лабораторные и самостоятельные занятия обучающихся в течение учебного семестра	Аналитический отчет, зачет

4	4	Числовые и строковые типы данных	ПК-1.1, ПК-1.2	Лекционные, самостоятельные занятия обучающихся в течение учебного семестра	Аналитический отчет, отчеты по лабораторным работам, зачет
5	5	Цикл for. Цикл while	ПК-1.1, ПК-1.2	Лекционные, самостоятельные занятия обучающихся в течение учебного семестра	Аналитический отчет, отчеты по лабораторным работам, зачет
6	6	Введение в списки. Вывод элементов списка	ПК-1.1, ПК-1.2	Лекционные, самостоятельные занятия обучающихся в течение учебного семестра	Аналитический отчет, отчеты по лабораторным работам, зачет
7	7	Функции без параметров. Функции с параметром	ПК-1.1, ПК-1.2	Лекционные, самостоятельные занятия обучающихся в течение учебного семестра	Аналитический отчет, отчеты по лабораторным работам, зачет
8	8	Введение в кортежи	ПК-1.1, ПК-1.2	Лекционные, самостоятельные занятия обучающихся в течение учебного семестра	Аналитический отчет, отчеты по лабораторным работам, зачет

2 Формы текущего контроля

Текущий контроль успеваемости проводится с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся и принятия необходимых мер по совершенствованию методики преподавания учебной дисциплины, организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков обучающихся: на занятиях, по результатам выполнения обучающимися индивидуальных заданий, проверки качества конспектов лекций и иных материалов.

Текущий контроль по дисциплине «Практика программирования с использованием Python» проводится в виде тестовых опросов по отдельным темам дисциплины, проверки заданий, выполняемых самостоятельно и на лабораторных занятиях, а также экспресс-опросов и заданий по лекционным материалам и лабораторным работам. Учебные пособия, рекомендуемые для самостоятельной работы и подготовки к лабораторным занятиям обучающихся по дисциплине «Практика программирования с использованием Python», содержат необходимый теоретический материал. Результаты самостоятельной работы контролируются преподавателем.

3 Формы промежуточного контроля

Формой промежуточного контроля по дисциплине является экзамен. Форма проведения экзамена – устный ответ, по утвержденным экзаменационным билетам, сформулированным с учетом содержания учебной дисциплины.

Целью проведения промежуточной аттестации является проверка компетенций, приобретенных студентом при освоении дисциплины.

4 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, шкал оценивания

Оценка степени формирования указанных выше (п. 1) контролируемых компетенций у обучающихся на различных этапах их формирования проводится преподавателем во время лекций, консультаций, практических и лабораторных занятий по шкале оценок «зачтено» – «не зачтено». Текущий контроль по дисциплине проводится в виде тестовых опросов по отдельным темам дисциплины, проверки заданий, выполняемых самостоятельно, и на лабораторных и практических занятиях, а также экспресс- опросов и заданий по лекционным материалам и лабораторным работам. Формирование у обучающихся во время обучения в семестре указанных выше компетенций на этапах лабораторных занятий и самостоятельной работы оценивается по критериям шкалы оценок - «зачтено» – «не зачтено». Освоение материала дисциплины и достаточно высокая степень формирования контролируемых компетенций обучающегося (своевременные выполнение и защита отчетов по лабораторным работам служат) основанием для допуска обучающегося к этапу промежуточной аттестации - экзамену.

Уровень теоретической подготовки студента определяется составом и степенью формирования приобретенных компетенций, усвоенных теоретических знаний и методов, а также умением осознанно, эффективно применять их при решении задач целенаправленного применения различных групп материалов в электронной технике.

Применяется двухбалльная шкала оценок: "зачтено", "не зачтено", что соответствует шкале "компетенции студента полностью соответствуют требованиям ФГОС ВО", "компетенции студента соответствуют требованиям ФГОС ВО", "компетенции студента не соответствуют требованиям ФГОС ВО".

Целью проведения промежуточной аттестации (зачета) является проверка общекультурных, общепрофессиональных и профессиональных компетенций, приобретенных студентом при изучении дисциплины «Практика программирования с использованием Python».

Зачет организуется и осуществляется, как правило, в форме собеседования. Средством, определяющим содержание собеседования студента с экзаменатором, является утвержденный список вопросов для тестирования.

Применяются следующие критерии оценивания компетенций (результатов):

-уровень усвоения материала, предусмотренного программой;

-умение анализировать материал, устанавливать причинно-следственные связи;

- полнота, аргументированность, убежденность ответов на вопросы;

- качество ответа (общая композиция, логичность, убежденность, общая эрудиция);

- использование дополнительной литературы при подготовке к этапу промежуточной аттестации.

5. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, шкал оценивания

К оценке уровня знаний и практических умений и навыков рекомендуется предъявлять следующие общие требования.

«Зачтено»:

знание основного программного материала дисциплины, понимание сущности и взаимосвязи основных рассматриваемых явлений (процессов):

понимание сущности обсуждаемых вопросов, правильные, без грубых ошибок ответы на поставленные вопросы, несущественные ошибки в ответах на дополнительные вопросы.

«Не зачтено»:

отсутствие знаний значительной части программного материала дисциплины; неправильный ответ хотя бы на один из вопросов, существенные и грубые ошибки в ответах на дополнительные вопросы, недопонимание сущности излагаемых вопросов, неумение применять теоретические знания при решении практических задач, отсутствие навыков в обосновании выдвигаемых предложений и принимаемых решений.

6. Критерии оценивания промежуточной аттестации

Таблица 1 - Критерии оценивания промежуточной аттестации

Шкала оценивания	Критерии оценивания
«зачтено»	**студент должен:** продемонстрировать общее знание изучаемого материала; знать основную рекомендуемую программой дисциплины учебную литературу; уметь строить ответ в соответствии со структурой излагаемого вопроса; показать общее владение понятийным аппаратом дисциплины; уметь устранить допущенные погрешности в ответе на теоретические вопросы и/или при выполнении практических заданий под руководством преподавателя, либо (при неправильном выполнении практического задания) по указанию преподавателя выполнить другие практические задания того же раздела дисциплины.
«не зачтено»	**ставится в случае:** незнания значительной части программного материала; не владения понятийным аппаратом дисциплины; существенных ошибок при изложении учебного материала; неумения строить ответ в соответствии со структурой излагаемого вопроса; неумения делать выводы по излагаемому материалу. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение по образовательной программе без дополнительных занятий по соответствующей дисциплине (формирования и развития компетенций, закрепленных за данной дисциплиной). Оценка «неудовлетворительно» выставляется также, если студент после начала экзамена отказался его сдавать или нарушил правила сдачи экзамена (списывал, подсказывал, обманом пытался получить более высокую оценку и т.д.).

7 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы *(для всех видов проводимых занятий или самостоятельных работ необходимо предусмотреть материалы для проверки знаний, умений и владений навыками)*

Типовые задания в рамках самостоятельной работы студентов для укрепления теоретических знаний, развития умений и навыков, предусмотренных компетенциями, закрепленными за дисциплиной.

Знакомство с Python: установка, запуск, первая программа print("Hello, World!")

— Работа в интерпретаторе и простом редакторе (например, IDLE, Thonny, VS Code).

Переменные и типы данных: int, float, str, bool

— Присваивание значений, вывод информации, понимание типов.

Арифметические операции: +, -, *, /, //, %, **

— Вычисления, приоритет операций, работа с числами.

Ввод данных с клавиатуры: функция input()

— Сбор данных от пользователя, преобразование строк в числа (int(input()), float(input())).

Логические выражения и операторы сравнения: ==, !=, <, >, <=, >=

— Сравнение чисел и строк, получение значений True/False.

Условный оператор if: простое ветвление

— Примеры: "если число положительное", "если пароль верный".

Конструкция if-else: два варианта действий

— Задачи: проверка возраста, определение чётности числа.

Конструкция if-elif-else: выбор из нескольких вариантов

— Оценка по баллам, выбор действия по номеру.

Цикл while: повторение действий при условии

— Счётчики, простые игры ("угадай число").

Цикл for с использованием range()

— Вывод чисел, сумма чисел от 1 до N, таблица умножения.

Работа со строками: длина, доступ к символам, конкатенация

— Примеры: "первая буква", "обратный порядок", "объединение имён".

Основные методы строк: .upper(), .lower(), .strip(), .replace()

— Форматирование текста, исправление опечаток.

Списки: создание, доступ к элементам, изменение значений

— Хранение нескольких значений (например, имена друзей, оценки).

Методы списков: .append(), .remove(), .pop(), .sort()

— Добавление и удаление элементов, сортировка.

Обход списков с помощью цикла for

— Вывод всех элементов, подсчёт количества, сумма значений.

Функции: определение с def, вызов, передача аргументов

— Простые функции: "приветствие", "вычисление площади", "проверка на чётность".

Функции, возвращающие значение: return

— Создание функций, которые возвращают результат (например, максимум из двух чисел).

Работа с файлами: чтение из текстового файла (txt)

— Открытие файла, построчное чтение, подсчёт строк.

Запись данных в текстовый файл

— Сохранение результатов программы (например, список покупок, лог действий).

Создание простой программы: калькулятор, викторина, список дел

— Объединение пройденных тем в одно приложение (в консоли).

Список **типовых контрольных вопросов** для оценки уровня сформированности знаний, умений и навыков, предусмотренных компетенциями, закрепленными за дисциплиной.

1. Какой оператор используется для вывода информации на экран в Python?
2. Что делает функция input()?
3. Как в Python ввести с клавиатуры целое число?
4. Чем отличается print(5 + 3) от print("5 + 3")?
5. Какие основные типы данных есть в Python? Назовите четыре.
6. Что такое переменная? Приведите пример её объявления.
7. Как проверить тип переменной в Python?
8. Что означает оператор + при работе со строками?
9. Как объединить две строки в одну?
10. Как получить длину строки?

11. Как получить первый символ строки?
12. Что делает оператор ==?
13. Чем отличается = от == в Python?
14. Как записывается условие "если число больше 10"?
15. Для чего используется ключевое слово else?
16. Что будет, если не поставить двоеточие после if?
17. Какой цикл используется для повторения действий определённое количество раз?
18. Какой цикл используется, когда условие проверяется перед каждым шагом?
19. Что делает функция range(1, 6)?
20. Как написать цикл, который выводит числа от 1 до 5?
21. Что такое список в Python? Приведите пример.
22. Как получить третий элемент списка?
23. Как добавить элемент в конец списка?
24. Как удалить элемент из списка по значению?
25. Что делает метод .sort() для списка?
26. Что такое функция в Python?
27. Как объявить функцию с именем greet?
28. Что делает ключевое слово return в функции?
29. Как открыть файл для чтения в Python?
30. Как записать текст в файл с помощью Python?

Типовые тестовые задания для укрепления и проверки теоретических знаний, развития умений и навыков, предусмотренных компетенциями, закрепленными за дисциплиной.

1. Какой оператор используется для вывода данных на экран в Python?
a) input()
b) print()
c) write()
d) output()

2. Что делает функция input()?
a) Выводит текст на экран
b) Считывает данные, введённые пользователем
c) Ожидает нажатия клавиши
d) Записывает данные в файл

3. Какой тип данных будет у результата input()?
a) int
b) float
c) str
d) bool

4. Как преобразовать строку "123" в целое число?
a) str(123)
b) int("123")
c) float("123")
d) bool("123")

5. Что выведет следующий код?
print(7 % 3)

a) 1
b) 2
c) 3
d) 0

6. Какой оператор используется для целочисленного деления?
a) /
b) //
c) %
d) **

7. Что делает оператор ** в Python?
a) Умножение
b) Деление
c) Возведение в степень
d) Извлечение корня

8. Как записать условие "если x не равен y"?
a) x != y
b) x == y
c) x > y
d) x < y

9. Что будет выведено, если выполнить этот код?
```
x = 5
if x > 3:
    print("Да")
else:
    print("Нет")
```
a) Да
b) Нет
c) Ошибка
d) Ничего

10. Как правильно записать условие "если число чётное"?
a) if num % 2 == 0:
b) if num // 2 == 0:
c) if num / 2 == 0:
d) if num % 2 != 0:

11. Что выведет следующий код?
```
for i in range(3):
    print(i)
```
a) 1 2 3
b) 0 1 2
c) 3 2 1
d) 0 1 2 3

12. Сколько раз выполнится цикл for i in range(5)?
a) 4 раза

b) 5 раз
c) 6 раз
d) 1 раз

13. Что делает функция len()?
a) Возвращает тип данных
b) Возвращает длину строки или списка
c) Преобразует в число
d) Проверяет, чётное ли значение

14. Как получить последний символ строки text?
a) text[0]
b) text[1]
c) text[-1]
d) text[last]

15. Как сделать все буквы строки заглавными?
a) text.lower()
b) text.upper()
c) text.title()
d) text.capitalize()

16. Что делает метод .strip()?
a) Разбивает строку на части
b) Удаляет пробелы в начале и в конце
c) Заменяет символы
d) Считает количество символов

17. Как создать пустой список?
a) list = ()
b) list = {}
c) list = []
d) list = ""

18. Как добавить элемент "банан" в конец списка fruits?
a) fruits.add("банан")
b) fruits.append("банан")
c) fruits.insert("банан")
d) fruits.push("банан")

19. Как удалить элемент "яблоко" из списка fruits?
a) fruits.remove("яблоко")
b) fruits.delete("яблоко")
c) fruits.pop("яблоко")
d) fruits.drop("яблоко")

20. Что делает метод .sort() для списка?
a) Перемешивает элементы
b) Сортирует элементы по алфавиту или по возрастанию
c) Удаляет дубликаты

d) Возвращает длину списка

21. Как правильно объявить функцию с именем hello?
a) def hello:
b) function hello():
c) def hello():
d) hello():

22. Что делает ключевое слово return в функции?
a) Выводит результат на экран
b) Возвращает значение из функции
c) Останавливает программу
d) Начинает новую функцию

23. Что выведет следующий код?
```
def add(a, b):
    return a + b
print(add(2, 3))
```
a) 2
b) 3
c) 5
d) 6

24. Как открыть файл data.txt для чтения?
a) open("data.txt", "w")
b) open("data.txt", "r")
c) read("data.txt")
d) file.open("data.txt")

25. Какой режим используется для записи в файл?
a) "r"
b) "a"
c) "w"
d) "w" или "a"

26. Что делает следующий код?
```
f = open("test.txt", "w")
f.write("Привет!")
f.close()
```
a) Читает файл
b) Создаёт файл и записывает в него текст
c) Удаляет файл
d) Печатает содержимое файла

27. Что произойдёт, если попытаться открыть несуществующий файл в режиме чтения?
a) Файл автоматически создастся
b) Программа выведет пустое значение
c) Возникнет ошибка
d) Программа проигнорирует ошибку

28. Какой из следующих фрагментов кода содержит синтаксическую ошибку?
a) print("Hello")
b) if x == 5:
c) for i in range(5)
d) name = "Анна"

29. Что делает цикл while?
a) Выполняется один раз
b) Выполняется, пока условие истинно
c) Выполняется заданное количество раз
d) Пропускает шаги

30. Что выведет следующий код?
s = "Привет"
print(s[1])
a) П
b) р
c) и
d) и (с ошибкой)

Фонд оценочных средств входит в состав рабочей программы дисциплины «Практика программирования с использованием Python», направление подготовки – 03.03.01 «Прикладные математика и физика», ОПОП «Электроника, квантовые системы и нанотехнологии».

Составил
доцент кафедры
микро- и наноэлектроники Ермачихин А.В.

Зав. кафедрой
микро- и наноэлектроники,
д.ф.-м.н., доцент Литвинов В.Г.

Оператор ЭДО ООО "Компания "Тензор"
ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
ПОДПИСАНО **ФГБОУ ВО "РГРТУ", РГРТУ,** Литвинов Владимир Георгиевич, Заведующий кафедрой МНЭЛ **03.11.26** 18:29 (MSK) Простая подпись