ПРИЛОЖЕНИЕ 1

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА
Кафедра радиотехнических систем

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
по дисциплине
Б1.О.13.03 «Информатика (Лабораторный практикум)»
Направление подготовки – 11.03.01 «Радиотехника»
ОПОП бакалавриата
«Радиотехнические системы локации, навигации и телевидения»
Квалификация (степень) выпускника – бакалавр
Форма обучения – очная

Рязань 2026 г.

Критерии оценивания уровня сформированности компетенций в процессе выполнения практических занятий:
- 41%-60% правильных ответов соответствует пороговому уровню сформированности компетенции на данном этапе ее формирования;
- 61%-80% правильных ответов соответствует продвинутому уровню сформированности компетенции на данном этапе ее формирования;
- 81%-100% правильных ответов соответствует эталонному уровню сформированности компетенции на данном этапе ее формирования.
Сформированность уровня компетенций не ниже порогового является основанием для допуска обучающегося к промежуточной аттестации по данной дисциплине. Формой промежуточной аттестации по данной дисциплине является экзамен, оцениваемый по принятой в ФГБОУ ВО «РГРТУ» четырехбалльной системе: «неудовлетворительно», «удовлетворительно», «хорошо» и «отлично». Критерии оценивания промежуточной аттестации представлены в таблице 1.

Таблица 1 - Критерии оценивания промежуточной аттестации (практические занятия, экзамен)

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. Оценка «не зачтено» также ставится студентам, которые не выполнили и защитили лабораторные работы и практические занятия, предусмотренные рабочей программой.
«отлично»	«Отлично» заслуживает студент, обнаруживший всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной программой. Как правило, оценка «отлично» выставляется студентам, усвоившим взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.
«хорошо»	«Хорошо» заслуживает студент, обнаруживший полное знание

	учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка «хорошо» выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.
«удовлетворительно»	«Удовлетворительно» заслуживает студент, обнаруживший знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по специальности, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка «удовлетворительно» выставляется студентам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.
«неудовлетворительно»	«Неудовлетворительно» выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине.

Вопросы к экзамену

1. Формальные и естественные языки. Интерпретируемые и компилируемые языки программирования. Интерпретатор командной строки, работа в IDLE
2. Основы языка. Переменные, оператор присвоения. Арифметические операции. Строковый и численные типы данных. Приведения типов
3. Функции, аргументы и возвращаемые значения, значения по умолчанию. Функции print и input
4. Условный оператор. Тернарный условный оператор. Циклы while, for. Ключевые слова range, break, continue
5. Управление потоком выполнения
6. Списки и кортежи. Mutable и immutable объекты. Хешируемость. Множества и словари. Сортировка
7. Структуры данных в Python
8. Работа в интегрированной среде разработки
9. Интегрированная среда разработки
10. Работа с файлами. Менеджеры контекста
11. Работа с текстовыми файлами и файлами pickle, JSON. Автоматизация обработки данных
12. Пакет numpy для вычислений. Пакет sympy для символьных преобразований
13. Модули numpy, sympy
14. Рисование графиков с помощью пакета matplotlib

15. Модуль matplotlib.pyplot
16. Пакет pandas для обработки данных
17. Использование модуля pandas для работы с электронными таблицами
18. Классы и объекты. Атрибуты и методы. Полиморфизм. Методы объекта, класса и статические. Композиция. Перегрузка операторов. Наследование
19. Объектно-ориентированное программирование
20. Принципы построения графических интерфейсов. Qt Designer. Widget. Layout. Button, Edit, Label, Double Spin Box, Check Box, Radio Button. Рисование графиков в PyQt
21. Графический интерфейс пользователя в PyQt5
22. Перспективы использования Python в инженерной практике

Примеры заданий для проведения практических занятий

1. Напишите программу для ввода с клавиатуры имени пользователя. В ответ программа здоровается с пользователем.

2. Напишите функцию `myfun`, рассчитывающую значение `y` по известному `x` в соответствии с математической функцией. Программа запрашивает у пользователя значение x, использует функцию `myfun` для расчёта значения `y` и выводит результат работы на экран.

3. Написать программу, генерирующую случайное число от 0 до 100. Для генерации случайных чисел использовать функцию `randint(a, b)` модуля random. Функция формирует целые числа в диапазоне от `a` до `b`. Пользователь, пока не угадает число, должен вводить с клавиатуры значения, на которые программа отвечает: «больше», «меньше», «угадали». Вывести на экран количество шагов, которое понадобилось пользователю, чтобы угадать искомое число.

4. Написать программу, запрашивающую у пользователя число (тип «`int`») и выводящую двоичное его значение (строкой — «`str`»). Реализовать перевод встроенными средствами Python и переводом в цикле.

5. Разработать функцию, аргументом принимающую список целых чисел и возвращающую кортеж уникальных значений, отсортированных в обратном порядке.

6. Разработать функцию, принимающую кортеж `a` целых чисел и целое значение `x`. Функция возвращает кортеж, включающий в себя фрагмент включительно от первого и до последнего вхождения `x` в кортеж `a`. Если вхождение одно, выборка происходит до конца. Если вхождений нет, программа возвращает пустой кортеж.

7. Разработать функцию, принимающую кортеж a целых чисел и целое значение `x`. Функция возвращает кортеж, отличающийся от `a` отсутствием в нём первого вхождения значения `x`.

8. Создать список `lst` из 5 словарей с информацией о студентах. Каждый словарь содержит поля `name` (имя), `age` (возраст), `grade` (оценка). Функция `good_students(lst)` возвращает список словарей с информацией о студентах, оценка которых выше средней.

9. В IDE Spyder сформировать набор инструментов среды и сохранить его с названием вида «гр. 000, Фамилия», указав номер группы и свою фамилию.

10. Разработать рекурсивную и нерекурсивную реализации функции расчёта факториала целого положительного числа.

11. Разработать рекурсивную и нерекурсивную реализации функции расчёта значения в треугольнике Паскаля.

12. Разработать рекурсивную и нерекурсивную реализации функции расчёта значения последовательности чисел Фибоначчи.

13. Реализовать рекурсивную реализации функции расчёта наибольшего общего делителя.

14. Для одних исходных данных выполнить вызов разработанных рекурсивных и нерекурсивных реализаций функций в разных ячейках программного кода в IDE Spyder. Измерить время выполнения, используя профайлер среды и функции `getTickCounter` и `getTickFrequency` модуля `cv2`.

15. Открыть файл «grades.json» в блокноте с кодировкой «UTF-8», исследовать его структуру.

16. Загрузить данные из файла «grades.json», используя импортируемый модуль json. Исследовать используемые типы данных, параметры структур. Сохранить данные в файл «grades0.pickle» с аргументом `protocol=0` и «grades0.pickle» с аргументом `protocol=4`. Сравнить размеры файлов «grades0.pickle», «grades4.pickle» и «grades.json» (в количестве байт). Прочитать данные из файла «grades4.pickle», убедиться в отсутствии ошибок чтения.

17. Из json-файла прочитать информацию, используя модуль json, и вывести список структур, фильтруемых по определённому параметру. Отсортировать список и вывести его в формате таблицы. Сохранить сформированный список в отдельный JSON-файл. Открыть его в блокноте. Сравнить размеры исходного и результирующего файлов (в количестве байт).

18. Прочитать координаты точек из файла `dots.pickle` в список `A`. Для каждого кортежа `a` в списке `A`: преобразовать кортеж в массив numpy; создать третью компоненту вектора, значение которой задать равным 1; выполнить матричное умножение заданной преподавателем матрицы на вектор. Разделить вектор b на значение его третьей компоненты, после чего удалить её. Сформировать двумерный массив `B`, столбцами которого являются сформированные на предыдущем шаге вектора `b`. Графически отобразить положение точек отдельных сформированных векторов на плоскости.

19. Задать одномерный массив `numpy x` значений, линейно меняющихся в заданном диапазоне с шагом `dx`. Задать численную функцию, эквивалентную аналитической, заданной преподавателем. Рассчитать значения массива `y` применением функции `fun` к значениям массива `x`.

20. Используя модуль `numpy`, выполнить численное интегрирование заданной преподавателем функции с шагом `dx` в некоторых пределах. Графически отобразить функцию, пределы интегрирования и вывести численное значение интеграла.

21. Создать символьную переменную `X`. Задать аналитическую функцию `F` в соответствии с номером варианта. Выполнить в аналитическом виде интегрирование и дифференцирование функции `F`. Получить численные реализации (`lambdify`) функций `Flam`, `dFlam` (производная) `iFlam` (интеграл) с использованием модуля `numpy`, построить их графики зависимостей. Рассчитать значение интеграла в пределах от 1 до 3. Сравнить результат с результатом численного интегрирования. Объяснить отличия.

22. Задать в программном коде список, фиксирующий экспериментальную зависимость Y(X). Отобразить зависимость курсивной кривой, сплошной кривой показать теоретическую зависимость Y(X). Вывести легенду графика. Оформить график согласно требованиям ГОСТ (шрифт, подписи осей, обозначения переменных, шаг сетки графика).

23. Задать в программном коде список, фиксирующий экспериментальную зависимость Y(X, k) где Y(X) - функция зависимости, k - параметр функции, численное значение которого неизвестно. Используя функцию curve_fit, оценить значение неизвестного параметра. Отобразить экспериментальную зависимость курсивной кривой, сплошной кривой показать теоретическую зависимость Y(X, k). Вывести легенду графика. Оформить график согласно требованиям ГОСТ (шрифт, подписи осей, обозначения переменных, шаг сетки графика).

24. Задать в программном коде список, фиксирующий семейство исследуемых зависимостей. Отобразить зависимости различными кривыми. Вывести легенду графика. Оформить график согласно требованиям ГОСТ (шрифт, подписи осей, обозначения переменных, шаг сетки графика).

25. Осуществить вывод статистики из файла «grades.json» за определённый месяц в соответствующий файл с расширением «xlsx». По строкам — различные студенты, по столбцам — различные дисциплины (пронумеровать). Список соответствий «номер-дисциплина» вывести на отдельный лист Excel-файла.

26. Осуществить вывод статистики из файла «grades.json» для определённого студента в соответствующий файл с расширением «xlsx». По строкам — различные дисциплины, по столбцам — дата (например для сентября 2023 г. — «2023 г., 9»).

27. Осуществить вывод статистики по каждой дисциплине на соответствующую вкладку xlsx-файла. По строкам — различные студенты, по столбцам — дата (например для сентября 2023 г. — «2023 г., 9»).

28. Разработать класс `CameraPTZ` PTZ-камеры Axis P-1355 с методами `set_pan` и `set_tilt` (задают азимутальный и угломестный углы поворота камеры), `set_zoom` (задаёт фокусное расстояние), `set_iris` (меняет выдержку матрицы), `enable_autoiris` (включает либо выключает автоматическую подстройку выдержки под освещённость сцены в зависимости от аргумента `enable`), `get_image` и `get_params` (возвращают массив

`numpy` с изображением с камеры и словарь с численными параметрами `pan`, `tilt`, `zoom`, `iris`, описывающими состояние камеры).

29. Разработать оконную форму управления PTZ-камерой Axis: `Line Edit` (3 шт.) предназначены для ввода IP-адреса, логина и пароля (в QtDesigner задать значения по умолчанию), кнопка `Push Button` — для подключения к камере; `Double Spin Box` с шагом 0,1 задают углы поворота, горизонтальные слайдеры (`Horizontal Slider`) — зум и выдержку; галочка (`Check Box`) включает автоматическую подстройку выдержки; `QWidget`, преобразованный в `ImageView` библиотеки `pyqtgraph`, отображает изображение с камеры. Разработать класс для управления объектами формы окна.

Составил
к.т.н., доцент кафедры РТС / А.С. Сычев /

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

ПОДПИСАНО **ФГБОУ ВО "РГРТУ", РГРТУ,** Кошелев Виталий Иванович, Заведующий кафедрой РТС **09.10.26** 14:28 (MSK) Простая подпись