

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. В.Ф. УТКИНА**

Кафедра «Автоматики и информационных технологий в управлении»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДИСЦИПЛИНЫ
ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Специальность 12.05.01

«Электронные и оптико-электронные приборы
и системы специального назначения»

ОПОП

«Оптико-электронные информационно-измерительные приборы и системы»

Квалификация выпускника – инженер

Формы обучения – очная

Рязань 2021 г.

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 сформированной общекультурных и профессиональных компетенций, приобретаемых обучающимися в соответствии с этими требованиями.

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

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 практических занятиях и лабораторных работах; по результатам выполнения контрольной работы; по результатам выполнения обучающимися индивидуальных заданий; по результатам проверки качества конспектов лекций и иных материалов.

Количество лабораторных и практических работ и их тематика определена рабочей программой дисциплины, утвержденной заведующим кафедрой. В качестве оценочных средств на протяжении семестра используется контрольные работы. При оценивании результатов освоения практических занятий и лабораторных работ, а также контрольных работ применяется шкала оценки «зачтено – не зачтено».

Промежуточный контроль по дисциплине осуществляется проведением зачета. При оценивании результатов освоения дисциплины на зачете применяется шкала оценки «зачтено – не зачтено».

Форма проведения зачета устный ответ, по утвержденным билетам к зачету, сформулированным с учетом содержания учебной дисциплины. В билет к зачету включается два теоретических вопроса по темам курса. После выполнения письменной работы обучаемого производится ее оценка преподавателем и, при необходимости, проводится теоретическая беседа с обучаемым для уточнения экзаменационной оценки.

Паспорт фонда оценочных средств по дисциплине

Модуль 1 (раздел 1)

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
1	2	3	4
1	Введение в языки высокого уровня	ПК-1.1-3 ПК-1.1-У ПК-1.1-В ПК-1.2-3 ПК-1.2-У ПК-1.2-В ПК-2.1-3 ПК-2.1-У ПК-2.1-В ПК-2.2-3 ПК-2.2-У ПК-2.2-В ПК-3.1-3 ПК-3.1-У ПК-3.1-В ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Зачет, Отчет о выполнении лабораторной работы
2	Основные конструкции языка Python	ПК-2.2-У ПК-2.2-В ПК-1.2-3 ПК-1.2-У ПК-1.2-В ПК-2.2-3 ПК-3.1-3 ПК-3.1-У ПК-3.1-В	Зачет, Отчет о выполнении лабораторной работы
3	Основные модули языка Python	ПК-1.1-3 ПК-1.1-У ПК-1.1-В ПК-1.2-3 ПК-1.2-У ПК-1.2-В ПК-2.1-3 ПК-2.1-У ПК-2.1-В ПК-2.2-3 ПК-2.2-У ПК-2.2-В ПК-3.1-3 ПК-3.1-У ПК-3.1-В ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Зачет, Отчет о выполнении лабораторной работы
4	Создание приложений с графическим интерфейсом пользователя на языке Python	ПК-3.1-3 ПК-3.1-У ПК-3.1-В	Зачет, Отчет о выполнении лабораторной работы
5	Объектно-ориентированное программирование на языке Python	ПК-1.1-3 ПК-1.1-У ПК-1.1-В ПК-1.2-3 ПК-1.2-У ПК-1.2-В ПК-2.1-3 ПК-2.1-У ПК-2.1-В ПК-2.2-3 ПК-2.2-У ПК-2.2-В ПК-3.1-3 ПК-3.1-У ПК-3.1-В ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Зачет, Отчет о выполнении лабораторной работы
6	Выполнение математических вычислений на языке Python	ПК-1.1-3 ПК-1.1-У ПК-1.1-В ПК-1.2-3 ПК-1.2-У ПК-1.2-В ПК-2.1-3 ПК-2.1-У ПК-2.1-В ПК-2.2-3	Зачет, Отчет о выполнении лабораторной работы

		ПК-2.2-У ПК-2.2-В ПК-3.1-З ПК-3.1-У ПК-3.1-В ПК-3.2-З ПК-3.2-У ПК-3.2-В	
7	Работа с математической и инженерной графикой на языке Python	ПК-1.2-З ПК-1.2-У ПК-1.2-В ПК-3.2-З ПК-3.2-У ПК-3.2-В	Зачет, Отчет о выполнении лабораторной работы
8	Функциональное программирование на языке Python	ПК-1.1-З ПК-1.1-У ПК-1.1-В ПК-1.2-З ПК-1.2-У ПК-1.2-В ПК-2.1-З ПК-2.1-У ПК-2.1-В ПК-2.2-З ПК-2.2-У ПК-2.2-В ПК-3.1-З ПК-3.1-У ПК-3.1-В ПК-3.2-З ПК-3.2-У ПК-3.2-В	Зачет, Отчет о выполнении лабораторной работы
9	Разработка сетевых приложений на языке Python	ПК-1.1-З ПК-1.1-У ПК-1.1-В ПК-1.2-З ПК-1.2-У ПК-1.2-В ПК-2.1-З ПК-2.1-У ПК-2.1-В ПК-2.2-З ПК-2.2-У ПК-2.2-В ПК-3.1-З ПК-3.1-У ПК-3.1-В ПК-3.2-З ПК-3.2-У ПК-3.2-В	Зачет, Отчет о выполнении лабораторной работы

Критерии оценивания уровня сформированности компетенций

1. Уровень усвоения материала, предусмотренного программой.
2. Умение анализировать материал, устанавливать причинно-следственные связи.
3. Ответы на вопросы: полнота, аргументированность, убежденность, умение
4. Качество ответа (его общая композиция, логичность, убежденность, общая эрудиция)
5. Использование дополнительной литературы при подготовке ответов.

Уровень освоения сформированности знаний, умений и навыков по дисциплине оценивается в форме двухбалльной отметки (зачтено/ не зачтено):

Оценка **«зачтено»** выставляется студенту, который прочно усвоил предусмотренный программный материал; правильно, аргументировано ответил на все вопросы, с приведением примеров; показал глубокие систематизированные знания, владеет приемами рассуждения и

сопоставляет материал из разных источников: теорию связывает с практикой, другими темами данного курса, других изучаемых предметов; без ошибок выполнил практическое задание.

Обязательным условием выставленной оценки является правильная речь в быстром или умеренном темпе. Дополнительным условием получения оценки «зачтено» могут стать хорошие успехи при выполнении самостоятельной работы, а также уверенная защита лабораторных работ.

Оценка «не зачтено» выставляется студенту, который не справился с 50% вопросов и заданий билета, в ответах на другие вопросы допустил существенные ошибки. Не может ответить на дополнительные вопросы, предложенные преподавателем. Целостного представления об изучаемой дисциплине у студента нет. Оценивается качество устной речи и изложение письменного материала, как и при выставлении положительной оценки.

Типовые контрольные задания или иные материалы

Вопросы к экзамену по дисциплине

Раздел 1

1. Языки высокого уровня. Основы программирования на языках высокого уровня: Структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование.

2. Python3. Динамическая типизация. Автоматическое управление памятью. Интроспекция. Понятие программы. Основные алгоритмические конструкции. Последовательность операторов.

3. Python3. Управление потоками: условия, циклы. Функции. Исключения. Встроенные типы данных. Выражения. Имена. Соглашения.

4. Python3. Понятие модуля. Пакеты. Встроенные функции.

5. Дополнительные модули Python3. Модули сервисов периода выполнения. Модули поддержки цикла разработки. Модули взаимодействие с ОС (файлы, процессы). Модули обработка текстов.

6. Дополнительные модули Python3. Модули многопоточных вычислениях. Хранение данных. Архивация. Платформено-зависимые модули.

7. Графический интерфейс пользователя. Пакет Tkinter. Основные активные элементы. Схемы размещения активных элементов. Обработка событий.

8. Основные понятия ООП. Абстракция и декомпозиция. Объекты. Типы и классы. Инкапсуляция. Полиморфизм. Наследование.

9. Основные понятия ООП. Агрегация. Методы. Статические методы. Метаклассы. Мультиметоды. Устойчивые объекты.

10. Выполнение математических вычислений на языке Python. Модуль Numeric. Функции модуля Numeric. Функции для работы с массивами. Модуль LinearAlgebra. Модуль RandomArray.

11. Работа с математической и инженерной графикой на языке Python. Модуль NumPlotLib. Построение двумерных графиков. Форматирование двумерных графиков. Построение трехмерных графиков

12. Понятие функционального программирования. Функциональная программа. Функция: определение и вызов. Рекурсия. Функции как параметры и результат. Обработка последовательностей. Итераторы. Простые генераторы. Карринг.

13. Работа в сети. Сокеты. Модуль smtpplib. Модуль poplib. Модули для работы с WWW. XML-PRC сервер.

Лабораторный практикум

Раздел 1

№ п/п	№ раздела (темы) Дисциплины	Наименование лабораторной работы	Трудоемкость, час
1	1(2-3)	Введение в программирование на языке Python	4
2	1(4-5)	Создание приложений с GUI в Python	4
3	1(6-8)	Работа с матрицами на языке Python. Построение графиков в языке Python	4
4	1(8-9)	Разработка сетевых приложений на языке Python	4

Планы практических занятий

Раздел 1

Тема 2. Основные конструкции языка Python

- 1) Знакомство с языком Python
- 2) Знакомство со средой программирования PyCharm (community edition)

Тема 3. Основные модули языка Python

- 1) Настройка и установка модулей Python
- 2) Подключение модулей Python к программе

Тема 4. Создание приложений с графическим интерфейсом пользователя на языке Python

- 1) Создания простейших приложений GUI в Python
- 2) Изучение менеджеров размещения в Python

Тема 5. Объектно-ориентированное программирование на языке Python

- 1) Создания приложений с соблюдением основных принципов ООП в Python
- 2) Работа с исключениями в Python.

Тема 6. Выполнение математических вычислений на языке Python

- 1) Изучение возможностей библиотеки NumPy.
- 2) Решение математических задач с использованием Python.

Тема 7. Работа с математической и инженерной графикой на языке Python

- 1) Изучение возможностей библиотеки Matplotlib
- 2) Построение двухмерных графиков с использованием Python.
- 3) Построение трехмерных графиков с использованием Python.

Тема 8. Функциональное программирование на языке Python

- 1) Создания приложений с соблюдением основных принципов функционального программирования в Python.
- 2) Использование рекурсии.

Тема 9. Разработка сетевых приложений на языке Python

- 1) Изучение возможностей библиотек Socket, Pickle.
- 2) Создание простейших сетевых приложений.