

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ПО ДИСЦИПЛИНЕ**
Программирование на Python

Код модуля
1154965(1)

Модуль
Операционные системы и языки
программирования

Екатеринбург

Оценочные материалы составлены автором(ами):

№ п/п	Фамилия, имя, отчество	Ученая степень, ученое звание	Должность	Подразделение
1	Борисов Василий Ильич	кандидат технических наук, без ученого звания	Доцент	радиоэлектроники и телекоммуникаций
2	Созыкин Андрей Владимирович	кандидат технических наук, без ученого звания	Доцент	информационных технологий и систем управления

Согласовано:

Управление образовательных программ

Т.Г. Комарова

Авторы:

- **Борисов Василий Ильич, Доцент, радиоэлектроники и телекоммуникаций**
- **Созыкин Андрей Владимирович, Доцент, информационных технологий и систем управления**

1. СТРУКТУРА И ОБЪЕМ ДИСЦИПЛИНЫ Программирование на Python

1.	Объем дисциплины в зачетных единицах	3	
2.	Виды аудиторных занятий	Лекции Практические/семинарские занятия	
3.	Промежуточная аттестация	Экзамен	
4.	Текущая аттестация	Контрольная работа	1
		Домашняя работа	1

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ (ИНДИКАТОРЫ) ПО ДИСЦИПЛИНЕ МОДУЛЯ Программирование на Python

Индикатор – это признак / сигнал/ маркер, который показывает, на каком уровне обучающийся должен освоить результаты обучения и их предъявление должно подтвердить факт освоения предметного содержания данной дисциплины, указанного в табл. 1.3 РПМ-РПД.

Таблица 1

Код и наименование компетенции	Планируемые результаты обучения (индикаторы)	Контрольно-оценочные средства для оценивания достижения результата обучения по дисциплине
1	2	3
ПК-1 -Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования	П-1 - Разрабатывать и оптимизировать программный код для решения задач обработки информации П-2 - Составлять техническую документацию по использованию и настройке компонентов программно-аппаратного комплекса У-1 - Анализировать техническое задание	Домашняя работа Контрольная работа Лекции Практические/семинарские занятия Экзамен

3. ПРОЦЕДУРЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ В РАМКАХ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ МОДУЛЯ В БАЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ (ТЕХНОЛОГИЧЕСКАЯ КАРТА БРС)

3.1. Процедуры текущей и промежуточной аттестации по дисциплине

1. Лекции: коэффициент значимости совокупных результатов лекционных занятий – 0.5		
Текущая аттестация на лекциях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>контрольная работа</i>	1,18	100
Весовой коэффициент значимости результатов текущей аттестации по лекциям – 0.5		
Промежуточная аттестация по лекциям – экзамен		
Весовой коэффициент значимости результатов промежуточной аттестации по лекциям – 0.5		
2. Практические/семинарские занятия: коэффициент значимости совокупных результатов практических/семинарских занятий – 0.5		
Текущая аттестация на практических/семинарских занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>домашняя работа</i>	1,18	100
Весовой коэффициент значимости результатов текущей аттестации по практическим/семинарским занятиям – 1		
Промежуточная аттестация по практическим/семинарским занятиям – нет		
Весовой коэффициент значимости результатов промежуточной аттестации по практическим/семинарским занятиям – не предусмотрено		
3. Лабораторные занятия: коэффициент значимости совокупных результатов лабораторных занятий – не предусмотрено		
Текущая аттестация на лабораторных занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по лабораторным занятиям – не предусмотрено		
Промежуточная аттестация по лабораторным занятиям – нет		
Весовой коэффициент значимости результатов промежуточной аттестации по лабораторным занятиям – не предусмотрено		
4. Онлайн-занятия: коэффициент значимости совокупных результатов онлайн-занятий – не предусмотрено		
Текущая аттестация на онлайн-занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по онлайн-занятиям – не предусмотрено		
Промежуточная аттестация по онлайн-занятиям – нет		
Весовой коэффициент значимости результатов промежуточной аттестации по онлайн-занятиям – не предусмотрено		

3.2. Процедуры текущей и промежуточной аттестации курсовой работы/проекта

Текущая аттестация выполнения курсовой работы/проекта	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент текущей аттестации выполнения курсовой работы/проекта– не предусмотрено		
Весовой коэффициент промежуточной аттестации выполнения курсовой работы/проекта– защиты – не предусмотрено		

4. КРИТЕРИИ И УРОВНИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ МОДУЛЯ

4.1. В рамках БРС применяются утвержденные на кафедре/институте критерии (признаки) оценивания достижений студентов по дисциплине модуля (табл. 4) в рамках контрольно-оценочных мероприятий на соответствие указанным в табл.1 результатам обучения (индикаторам).

Таблица 4

Критерии оценивания учебных достижений обучающихся

Результаты обучения	Критерии оценивания учебных достижений, обучающихся на соответствие результатам обучения/индикаторам
Знания	Студент демонстрирует знания и понимание в области изучения на уровне указанных индикаторов и необходимые для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Умения	Студент может применять свои знания и понимание в контекстах, представленных в оценочных заданиях, демонстрирует освоение умений на уровне указанных индикаторов и необходимых для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Опыт /владение	Студент демонстрирует опыт в области изучения на уровне указанных индикаторов.
Другие результаты	Студент демонстрирует ответственность в освоении результатов обучения на уровне запланированных индикаторов. Студент способен выносить суждения, делать оценки и формулировать выводы в области изучения. Студент может сообщать преподавателю и коллегам своего уровня собственное понимание и умения в области изучения.

4.2 Для оценивания уровня выполнения критериев (уровня достижений обучающихся при проведении контрольно-оценочных мероприятий по дисциплине модуля) используется универсальная шкала (табл. 5).

Таблица 5

Шкала оценивания достижения результатов обучения (индикаторов) по уровням

Характеристика уровней достижения результатов обучения (индикаторов)		
№ п/п	Содержание уровня выполнения критерия оценивания результатов	Шкала оценивания

	обучения (выполненное оценочное задание)	Традиционная характеристика уровня		Качественная характеристика уровня
1.	Результаты обучения (индикаторы) достигнуты в полном объеме, замечаний нет	Отлично (80-100 баллов)	Зачтено	Высокий (В)
2.	Результаты обучения (индикаторы) в целом достигнуты, имеются замечания, которые не требуют обязательного устранения	Хорошо (60-79 баллов)		Средний (С)
3.	Результаты обучения (индикаторы) достигнуты не в полной мере, есть замечания	Удовлетворительно (40-59 баллов)		Пороговый (П)
4.	Освоение результатов обучения не соответствует индикаторам, имеются существенные ошибки и замечания, требуется доработка	Неудовлетворительно но (менее 40 баллов)	Не зачтено	Недостаточный (Н)
5.	Результат обучения не достигнут, задание не выполнено	Недостаточно свидетельств для оценивания		Нет результата

5. СОДЕРЖАНИЕ КОНТРОЛЬНО-ОЦЕНОЧНЫХ МЕРОПРИЯТИЙ ПО ДИСЦИПЛИНЕ МОДУЛЯ

5.1. Описание аудиторных контрольно-оценочных мероприятий по дисциплине модуля

5.1.1. Лекции

Самостоятельное изучение теоретического материала по темам/разделам лекций в соответствии с содержанием дисциплины (п. 1.2. РПД)

5.1.2. Практические/семинарские занятия

Примерный перечень тем

1. Элементы функционального программирования в Python.
2. Объектно-ориентированное программирование в Python.
3. Библиотеки и репозитории в Python.
4. Инструменты разработчика в Python.
5. Code Review. Стили программирования.
6. Работа с текстовыми файлами в Python.
7. Библиотеки для анализа данных в Python.
8. Интеграция с базами данных в Python.
9. Веб-программирование на Python.

Примерные задания

Создание Web-приложений в Python.

Работа с базами данных в Python.

LMS-платформа

1. <https://lms.skillfactory.ru/courses/course-v1:SkillFactory+URFUML22p1s+SEP2022/course/>

5.2. Описание внеаудиторных контрольно-оценочных мероприятий и средств текущего контроля по дисциплине модуля

Разноуровневое (дифференцированное) обучение.

Базовый

5.2.1. Контрольная работа

Примерный перечень тем

1. Инструменты разработчика на Python

Примерные задания

Выберите библиотеки, которые позволяют работать с операционной системой:

- numpy
- flask
- sys
- os
- keras

Выберите библиотеки для создания web приложений:

- selenium
- flask
- scrapy
- django
- requests

Выберите библиотеки для построения интеллектуальных систем:

- openCV
- pytorch
- tornado
- tensorflow

В каких целях файл с программным кодом разделяется на логические модули:

- чтобы стало легче масштабировать и поддерживать программу
- чтобы стало сложнее ориентироваться в программе
- чтобы стало больше файлов

Чтобы python смог отличить обычный пакет от каталога нужно:

- нужно назвать каталог особым образом
- ничего не нужно Python сам все найдет
- нужно добавить в каталог файл `__init__.py`

В каком файле находится информация, необходимая для установки пакета:

- README.md

- setup.py
- main.py
- MANIFEST.in

Какая команда git отправляет изменения из локального репозитория в удаленный?

- add
- push
- commit t

Что такое разрешение конфликтов в git:

- исправление ошибок при слиянии веток (например, после выполнения pull)
- исправление ошибок при коммите
- исправление ошибок при создании веток

Что НЕ может делать отладчик?

- Пошаговое выполнение кода
- Просмотр стека вызовов функций
- Автоматически проверять результат работы функции на различных значениях
- Выполнение кода с заходом или пропуском функций

Какие подходы используются для тестирования программы?

- Ручное тестирование
- Тестирование с помощью функций
- Автоматическое тестирование

Какой компонент Python подходит и для написания тестов, и для создания документации?

- Docstring
- Pytest
- Pdb

Профилирование кода — это поиск неэффективных участков кода с помощью...?

- синтаксического анализа кода
- определения времени выполнения различных участков кода
- проверки результатов работы различных участков кода

Что позволяет сделать IDE при рефакторинге кода?

- Извлечение методов, переменных и констант из существующего кода
- Извлечение классов из существующих сигнатур классов
- Извлечение строк комментариев из кода программы

Стандартом, описывающим правила оформления кода на Python является:

- ISO 9001-2015
- ГОСТ 19.506-79
- MISRA
- PEP8

В каких случаях требуется рефакторинг:

- вы работаете над кодом в команде
- вы дорабатываете и поддерживаете уже существующий код
- вы работаете над кодом в одиночку
- отдаете первую рабочую версию кода и больше не редактируете ее
- время жизни проектов велико (годы, десятки лет)
- код пишется с нуля, а не редактируется версия, написанная другим человеком
- версии программы меняются очень часто (недели, месяц)

LMS-платформа

1. <https://lms.skillfactory.ru/courses/course-v1:SkillFactory+URFUML22p1s+SEP2022/course/>

5.2.2. Домашняя работа

Примерный перечень тем

1. Работа с базами данных в Python.
2. Создание Web-приложений в Python.

Примерные задания

Выполните анализ готовых наборов данных Atlas в MongoDB (<https://developer.mongodb.com/article/atlas-sample-datasets>). Допускается как развернуть MongoDB локально, скачав набор данных, так и воспользоваться облаком. Для решения задач необходимо установить модуль pymongo.

Задания для анализа:

- Найдите количество записей в `sample_weatherdata`, в которых давление меньше 1000. В ответе укажите одно число.
- В `sample_restaurants` найдите, сколько ресторанов в Бронксе содержат в названии «Food» (например, F & B Foods Llc).
- В `sample_supplies` найти минимальный и максимальный возраст клиентов. В ответе записать числа через запятую без пробелов. Пример: 28,64.

На порту 27017 хоста 127.0.0.1 запущен сервер MongoDB с базой данных `db`. В этой базе есть коллекция `users`, в которой пользователи хранятся в таком виде: `{ "name": "Kenneth", "surname": "Cruz", "age": 30, "rating": 5.0, "trip_count": 2478.0, "registered_at": "2010-04-25", "avg_trip_km": 6.99, "favorite_music": "Cream", }`. Реализуйте функцию `get_ages_sum`, возвращающую сумму возрастов всех пользователей в данной коллекции. Для связи с сервером MongoDB необходимо использовать модуль `pymongo`.

С помощью фреймворка `flask` разработайте сайт с двумя страничками:

- `/`, которая бы возвращала текст `hello`
- `/counter`, которая бы возвращала число, количество посетителей, зашедших на данную страничку. Т.е. у первого посетителя - число 1, у второго - число 2 и т.д.

LMS-платформа

1. <https://lms.skillfactory.ru/courses/course-v1:SkillFactory+URFUML22p1s+SEP2022/course/>

5.3. Описание контрольно-оценочных мероприятий промежуточного контроля по дисциплине модуля

5.3.1. Экзамен

Список примерных вопросов

1. Функциональное программирование в Python.
 2. Объектно-ориентированное программирование в Python.
 3. Создание модулей и пакетов в Python.
 4. Создание библиотек в Python.
 5. Командная разработка в Python. Использование git.
 6. Отладка программ на Python.
 7. Рефакторинг кода на Python.
 8. Code Review на Python.
 9. Среды разработки (IDE) на Python.
 10. Работа с текстовыми файлами различных форматов на Python.
 11. Работа с базами данных SQL на Python.
 12. Работа с базой sqlite на Python.
 13. Работа с базой данных MongoDB на Python.
 14. Создание web-приложений на Python.
 15. Использование библиотеки Flask на Python.
 16. Использование библиотеки Django на Python.
 17. Многопоточное программирование в Python.
 18. Библиотеки для многопоточного программирования в Python.
 19. Библиотеки для взаимодействия с операционной системой в Python.
 20. Библиотеки для анализа данных в Python.
- LMS-платформа – не предусмотрена

5.4 Содержание контрольно-оценочных мероприятий по направлениям воспитательной деятельности

Направления воспитательной деятельности сопрягаются со всеми результатами обучения компетенций по образовательной программе, их освоение обеспечивается содержанием всех дисциплин модулей.