

## Программа учебной дисциплины «Программирование на Python и работа с Linux»

Утверждена  
Академическим советом ООП  
Протокол № 2.03-09/2706-01  
от «27» июня 2018г.

Автор	Черепанов Ярослав Андреевич, yaroslav.cherepanov@gmail.com Свешников Михаил, mike0sv@gmail.com
Число кредитов	6
Контактная работа (час.)	32
Самостоятельная работа (час.)	196
Курс	1
Формат изучения дисциплины	без использования онлайн курса

### I ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ

Целями освоения дисциплины является формирование у студентов комплекса теоретических знаний и методологических основ в области операционных систем семейства Linux и языка программирования Python, а также практических навыков, необходимых для практического использования этих инструментов.

В результате освоения дисциплины студент должен:

- обладать базовым уровнем программирования на языке Python;
- иметь навыки сбора и обработки данных на языке Python;
- уметь писать bash команды в операционной системе Linux;  
уметь подключаться к облачным ресурсам машин по SSH.

В рамках магистерской программы «Финансовые технологии и анализ данных» настоящая дисциплина является дисциплиной по выбору. Данный курс является адаптационным.

Для освоения учебной дисциплины, студенты должны знать основные классы ИТ-систем, лучшие практики и современные стандарты в сфере информационных технологий.

Также студенты должны владеть навыками проектирования информационных систем, уметь систематизировать и обобщать информацию, разрабатывать конкретные предложения по результатам исследований, готовить справочно-аналитические материалы для принятия решений в сфере информационных технологий.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

- «Банковская ИТ-инфраструктура»;
- «Современные методы принятия решения: Алгоритмы обработки больших данных»;

## **II СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

### **Тема 1. Основы Linux**

Структура курса, контрольные мероприятия, оценка за работу в семестре. Краткая история Linux. Командный процессор Bash. Сетевой протокол SSH. Работа с директориями. Работа с файлами. Текстовый редактор vim. Переменные среды. Пайпы. Пакетные менеджеры.

*Литература:*

Основная литература: [1].

### **Тема 2. Введение в Python**

Zen of Python. Руководство по написанию кода на Python (PEP8). IDE для разработки. Интерпретируемость языка. Переменные, условия, циклы. Динамическая типизация. Ввод-вывод. Примитивы, работа со строками, стандартные структуры данных (tuple, list, dict, set), iterables, slices, list comprehensions, генераторы. Функции и области видимости.

*Литература:*

Основная литература: [2], [3], [4]

### **Тема 3. Файлы и модули**

Работа с файлами. Работа с встроенными модулями: sys, os, shutil, collections, itertools, time, datetime, re, json, requests. Работа с внешними компонентами: pip, venv, conda, ipython, jupyter.

*Литература:*

Основная литература: [2], [3], [4]

### **Тема 4. Классы**

Конструкторы, магические методы, статические методы, наследование.

*Литература:*

Основная литература: [2], [3], [4]

### **Тема 5. Объекты**

Утиная типизация, лямбды, map, reduce, декораторы, контекст-менеджеры, exceptions, try-catch-finally, monkey patching, секретные атрибуты

*Литература:*

Основная литература: [2], [3], [4]

## **Тема 6. Параллельные вычисления**

Subprocess, GIL, Threading, Multiprocessing

*Литература:*

Основная литература: [5]

### **III. ОЦЕНИВАНИЕ**

Студент должен продемонстрировать знание разделов дисциплины и способность представить результаты выполнения домашнего задания в соответствии с требуемыми компетенциями.

Оценки по всем формам текущего контроля выставляются по 10-ти балльной шкале.

Накопленная оценка по дисциплине рассчитывается по формуле:

$$O_{\text{накопленная}} = (1/3) * O_{\text{дз1}} + (1/3) * O_{\text{дз2}} + (1/3) * O_{\text{ауд работа (посещаемость)}}.$$

Накопленная и итоговая оценки округляются арифметически.

Итоговая оценка рассчитывается по формуле:

$$O_{\text{итоговая}} = O_{\text{накопленная}} * 0,6 + O_{\text{экзамен}} * 0,4.$$

## **IV. ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ**

### **10.1 Содержание заданий текущего контроля**

Выполнение домашнего задания предусматривает написание кода на языке программирования Python с использованием встроенных и сторонних библиотек для работы с аргументами командной строки, протоколом http/https, синтаксическим парсером HTML/XML, а также написание кода в парадигме ООП.

### **10.2 Вопросы для оценки качества освоения дисциплины**

#### **Вопросы к Разделу 1. ОСНОВЫ LINUX**

1. Что такое SSH? Зачем используется?
2. Команды bash для работы с директориями
3. Команды bash для работы с файлами
4. Для чего нужны переменные среды?
5. Использование пайпов

#### **Вопросы к Разделу 2. ВВЕДЕНИЕ В PYTHON**

1. Назовите парадигмы программирования, которым следует Python. Объясните, что они значат.

2. Что такое интерпретируемый язык программирования (которым является Python)?
3. Что означает динамическая типизация в Python?
4. Какие есть встроенные коллекции в Python. Напишите примеры их использования
5. Что такое mutable и immutable объекты
6. Понятие list, dict, set comprehension, их пример
7. Что такое итератор?
8. Зачем нужны функции? Как они задаются? Опишите все способы передачи аргументов.
9. В чем отличие генератора от функции?

### **Вопросы к Разделу 3. ФАЙЛЫ И МОДУЛИ**

1. Открытие и чтение файлов в Python
2. Использование модулей, управление ими, менеджеры пакетов
3. Работа с командной строкой из Python
4. Перечислите коллекции из модуля collections в дополнение к встроенным

### **Вопросы к Разделу 4. КЛАССЫ**

1. Назовите 3 принципа ООП. Поясните, что они значат.
2. Напишите своими словами, зачем нужно ООП.
3. Что такое класс и чем он отличается от экземпляра.
4. Как в питоне реализована инкапсуляция?
5. Чем обычные методы класса отличаются от статических и классовых? Опишите примеры их использования.
6. Что такое магические методы? Перечислите несколько и опишите как они могли бы использоваться
7. Что такое метакласс?
8. Что такое декоратор? Опишите принцип работы и приведите пример использования

### **Вопросы к Разделу 5. ОБЪЕКТЫ**

1. Что такое утиная типизация? Как её можно использовать?
2. Как написать свой контекстный менеджер? Опишите 2 способа.
3. Что такое анонимная функция? Каким образом они представлены в Python?
4. Как обрабатываются исключения в Python?
5. Что делают функции map и reduce? Приведите примеры.

## Вопросы к Разделу 6. ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ

1. Чем параллельные вычисления отличаются от конкурентных?
2. Могут ли конкурентные вычисления давать прирост производительности при одном физическом ядре? Если могут, то каким образом?
3. Чем процесс отличается от потока(треда)
4. Что такое GIL? Какие его плюсы и минусы?
5. Какую проблему решают локи? Как это работает?
6. Какое принципиальное отличие модулей threading и multiprocessing (имеется в виду то, что вызвано GIL-ом)?

### 10.3 Пример экзаменационных задач

1. Написать функцию, делающую из списка, состоящего из вложенных в него списков и других объектов, список состоящий только из объектов с сохранением порядка;
2. Переписать код с циклами с помощью функций map, reduce, filter.

## V. РЕСУРСЫ

### 12.1 Основная литература

1. Cobbaut, Paul. «Mastering Linux: Fundamentals» - Samurai Media Limited, 2016. – с. 366
2. Мэтиз Э. «Изучаем Python. Программирование игр, визуализация данных, веб-приложения». – Питер, 2018 – с. 496
3. Любанович Б. «Простой Python. Современный стиль программирования». – Питер, 2016 – с. 480
4. Лутц М. «Изучаем Python» - Символ-Плюс, 2011 – с. 1280
5. Мартелли А., Рейвенскрофт А., Холден С. «Python. Справочник. Полное описание языка». – Вильямс, 2018 – с. 896

### Программное обеспечение

№ п/п	Наименование	Условия доступа
1.	Интерпретатор python 3.5+	Свободно распространяемое ПО

