Программа курса
Python Basic

Версия 2.0.0

Продолжительность курса: 84 пары (42 дня)

Цель курса

Обучить слушателя основам программирования на Python.

По окончании курса слушатель будет:

- разбираться в типах данных Python;
- понимать тонкости работы с переменными;
- применять механизмы преобразования типов данных;
- использовать условия и циклы;
- взаимодействовать со строками;
- создавать функции;
- понимать особенности реализации механизмов ООП в Python;
- понимать принципы функционального программирования;
- уметь пользоваться системой контроля версий;
- применять паттерны проектирования;
- использовать юнит-тестирование.

По окончании данного курса студент сдает практическое задание и теоретический экзамен по материалам курса. Для допуска к экзамену должны быть сданы все домашние и практические задания. Практическое задание должно охватывать максимум материала из различных разделов курса.

Перед началом данного предмета необходимо предоставить студентам доступ к следующему курсу Microsoft Imagine Academy: Introduction to Python.
Этот курс является подготовкой к экзамену MTA 98-381: Introduction to Programming Using Python.

Тематический план

Модуль 1

Введение в Python

1. **Обзор языков программирования.**
   - Знакомство основными парадигмами программирования.
   - Обзор современных языков программирования.
   - Понятие алгоритма.
   - Знакомство с языком Python, сферы применения.
2. **Введение в Python. Интерпретатор Python и его окружение.**
   - Введение в Python.
   - Понятие интерпретатора и порядок установки.
   - Знакомство со средами программирования:
     - стандартный пакет программирования IDLE и Python Shell;
     - IDE PyCharm, Spyder, Visual Studio, Visual Studio Code;
     - Atom.
3. **Типы данных, переменные и синтаксические конструкции.**
   - Тип и значения.
   - Переменные.
   - Имена переменных и зарезервированные слова.
   - Инструкции.
   - Операторы и операнды.
   - Приоритеты операторов.
   - Операции над переменными.
   - Порядок выполнения программы.
   - Ввод/вывод.
   - Преобразование типов.
   - Ошибки синтаксические и логические, работа с ними.

Модуль 2

Операторы ветвлений, циклы

1. **Условные инструкции и их синтаксис.**
   - Понятие «блока» выполнения.
   - Логические выражения и операторы.
   - Операторы ветвления if ... else, if elif else.
   - Вложенные конструкции.

2. **Циклы.**
   - Понятие итерации.
   - Цикл while.
   - Бесконечные циклы.
   - Управляющие операторы continue, break и else.
   - Цикл for.
   - Локальные и глобальные переменные.

Модуль 3

Строки, списки

1. **Строки.**
   - Кодировка ASCII, Unicode, UTF-8, Byte-code.
   - Строка – неизменяемая последовательность символов.
   - Методы строк.
   - Особенности работы со строками.
   - Срез строки.
   - Экранированные последовательности.
   - «Сырые строки».
   - Форматированный вывод.
   - Модуль string.
   - Байты и кодировки.
   - Регулярные выражения, модуль re.
2. **Списки.**
   - Понятие классического массива.
   - Понятие коллекции объектов.
   - Ссылочный тип данных list.
   - Создание списков.
   - Генераторы списков.
   - Работа со списками.
   - Методы списков.
   - Оператор принадлежности in.
   - Особенности списков, ссылки и клонирование.
   - Поиск элемента.
   - Матрицы.

Модуль 4

Сортировка, поиск

1. **Сортировка.**
   - Оптимальность.
   - Сортировка пузырьком.
   - Сортировка слиянием.
   - Сортировка Шелла.
   - Пирамидальная сортировка.
   - Быстрая сортировка.
2. **Поиск.**
   - Линейный поиск.
   - Бинарный поиск.

Модуль 5

Кортежи, множества, словари

1. **Кортежи.**
   - Коллекции неизменяемых объектов.
   - Применение и особенности кортежа.
2. **Множества.**
   - Математическое понятие множеств.
   - Тип данных set(), frozenset().
   - Операции над множествами.
   - Применение множеств.
3. **Словари.**
   - Ассоциативные массивы.
   - Хеш-таблицы.
   - Создание словаря.
   - Методы словаря.
   - Понятие разреженной матрицы.
4. **Практические примеры использования.**

Модуль 6

Функции, модули

1. **Функции и модули.**
   - Что такое функция?
   - Цели и задачи функции.
   - Встроенные функции.
   - Математические функции и случайные числа.
   - Синтаксис объявления функций.
   - Аргументы и возвращаемые значения.
2. **Расширенные приемы по работе с функциями.**
   - Распаковка и упаковка аргументов.
   - Аргументы по умолчанию, аргументы-ключи.
   - Область видимости, правило LEGB.
   - Локальные и глобальные переменные в функциях.
   - Функции как объекты первого класса.
   - Рекурсия.
3. **Функциональное программирование.**
   - Что такое функциональное программирование?
   - Анонимные функции lambda.
   - Модуль functools.
   - Функции map(), reduce(), filter(), zip().
   - Функции высших порядков.
   - Замыкание.
4. **Замыкание.**
5. **Карринг.**
6. **Декораторы.**

Модуль 7

Исключения

1. **Что такое исключение?**
2. **Типы исключений.**
3. **Конструкция try except finally.**
   - Что такое try except finally?

- Цели и задачи try except finally.
- Блок try.
- Блок except.
- Блок finally.
- Примеры использования.

4. **Базовые типы исключений.**
   - BaseException.
   - Exception.
   - ArithmeticError.
   - BufferError.
   - LookupError.
   - IndexError.
   - KeyError.
   - OverflowError.
   - Другие типы исключений.
5. **Генерация исключений.**
   - Зачем генерировать исключение?
   - Ключевое слово raise.
   - Примеры использования.

Модуль 8

Файлы

1. **Файлы.**
   - Файловая система, особенности реализации форматов.
   - Работа с файлами:
     - открытие;
     - закрытие;
     - чтение;
     - запись.
2. **Менеджер контекста.**
3. **Типы файлов, текстовые и бинарные.**
4. **Практические примеры использования.**

Модуль 9

Системы контроля версий

1. **Что такое контроль версий?**
2. **Зачем нужен контроль версий.**
3. **Обзор систем контроля версий.**
   - CVS.
   - SVN.
   - Git.
   - Другие системы контроля версий.
4. **Git.**
   - Что такое Git?
   - Цели и задачи Git?
   - Основные термины:
     - репозиторий;
     - коммит;
     - ветка;
     - рабочий каталог.
   - Операции с Git:
     - установка;
     - создание репозитория;
     - добавление файла в репозиторий;
     - запись коммита в репозиторий;
     - получение текущего состояния рабочего каталога;
     - отображение веток;
     - операции с накопительным буфером;
     - git remote;
     - git push;
     - git pull;
     - другие операции.
5. **Использование внешних сервисов (github).**

Модуль 10

ООП

1. **Введение в ООП.**
   - Понятие ООП.
   - Инкапсуляция.
   - Наследование.
   - Полиморфизм.
   - Особенности реализации ООП в Python, «утиная типизация».
2. **Типы данных, определяемые пользователем.**
   - Экземпляр класса.
   - Классы и объекты.
   - Атрибуты, поля (свойства), методы класса.
   - Перегрузка методов.
   - Magic-методы, конструкторы.
   - Статические методы и методы класса.
3. **Наследование и инкапсуляция.**
   - Общедоступный, внутренний и приватный метод.
   - Множественное наследование и MRO (порядок разрешения методов).
4. **Полиморфизм.**
   - Перегрузка операторов.
   - Реализация магических методов.
5. **Создание и управление поведением экземпляров класса.**
   - Функторы.
   - Декораторы.
   - Управляемые атрибуты.
   - Свойства.
   - Дескрипторы.
6. **Метаклассы.**
   - Модель метаклассов.
   - Метод конструктор __new__().
   - Протокол инструкции class.

Модуль 11

Модульное тестирование

1. **Что такое модульное тестирование?**
2. **Цели и задачи модульного тестирования.**
3. **Необходимость модульного тестирования.**
4. **Обзор инструментов для модульного тестирования.**
5. **Инструмент для модульного тестирования Python-приложений.**

Модуль 12

Структуры данных

1. **Связанные списки.**
   - Что такое список?
   - Односвязный и двусвязный список.
   - Практические примеры использования.
2. **Стек.**
   - Что такое стек?
   - Принцип LIFO.
   - Практические примеры использования.
3. **Очередь.**
   - Что такое очередь?
   - Виды очередей.
   - Практические примеры использования.
4. **Деревья.**
   - Что такое дерево?
   - Виды деревьев.
   - Практические примеры использования.

Модуль 13

Введение в паттерны проектирования

1. **Что такое паттерны проектирования.**
2. **Причины возникновения паттернов проектирования.**
3. **Понятие паттерна проектирования.**

4. **Принципы применения паттернов проектирования.**
5. **Принципы выбора паттернов проектирования.**
6. **Принципы разделения паттернов на категории.**
7. **Введение в UML.**
   - Диаграмма классов.
   - Диаграмма объектов.
   - Диаграмма взаимодействия.
8. **Использование UML при анализе паттернов проектирования.**
   - Диаграмма классов.
   - Диаграмма объектов.
   - Диаграмма взаимодействия.
9. **Порождающие паттерны.**
   - Что такое порождающий паттерн?
   - Цели и задачи порождающих паттернов.
   - Обзор порождающих паттернов.
   - Разбор порождающих паттернов:
     - Abstract Factory:
       - цель паттерна;
       - причины возникновения паттерна;
       - структура паттерна;
       - результаты использования паттерна;
       - практический пример использования паттерна.
     - Builder:
       - цель паттерна;
       - причины возникновения паттерна;
       - структура паттерна;
       - результаты использования паттерна;
       - практический пример использования паттерна.
     - Factory Method:
       - цель паттерна;
       - причины возникновения паттерна;
       - структура паттерна;
       - результаты использования паттерна;
       - практический пример использования паттерна.
     - Prototype:
       - цель паттерна;
       - причины возникновения паттерна;

        - структура паттерна;
        - результаты использования паттерна;
        - практический пример использования паттерна.
    - Singleton:
        - цель паттерна;
        - причины возникновения паттерна;
        - структура паттерна;
        - результаты использования паттерна;
        - практический пример использования паттерна.

10. Структурные паттерны.

- Что такое структурный паттерн?
- Цели и задачи структурных паттернов.
- Обзор структурных паттернов.
- Разбор структурных паттернов:
    - Adapter;
    - Composite;
    - Facade;
    - Proxy;
    - другие структурные паттерны.

11. Паттерны поведения.

- Что такое паттерны поведения?
- Цели и задачи паттернов поведения.
- Обзор паттернов поведения.
- Разбор паттернов поведения:
    - Command;
    - Iterator;
    - Observer;
    - Strategy;
    - другие структурные паттерны.

Модуль 14

Принципы проектирования классов SOLID

1. **Обзор проблем, встречающихся при проектировании и разработке классов.**
2. **Принципы проектирования классов SOLID.**
    - Принцип единственности ответственности (The Single Responsibility Principle).

- Принцип открытости/закрытости (The Open Closed Principle).
- Принцип подстановки Барбары Лисков (The Liskov Substitution Principle).
- Принцип разделения интерфейса (The Interface Segregation Principle).
- Принцип инверсии зависимостей (The Dependency Inversion Principle).

3. **Примеры использования принципов SOLID.**

Модуль 15

Работа в команде, управление программными проектами

1. **Что такое управление программными проектами?**
2. **Причины возникновения дисциплины управление программными проектами.**
3. **Диаграммы Ганта.**
4. **Важные вопросы по управлению программными проектами.**
   - Что такое проект и программный проект?
   - Что такое жизненный цикл процесса разработки программного обеспечения?
   - Что такое управление проектами?
   - Что такое одиночная разработка?
   - Что такое командная разработка?
   - Анализ проблем одиночной и командной разработки программного обеспечения.
5. **Анализ терминов предметной области.**
   - Процесс.
   - Проект.
   - Персонал.
   - Продукт.
   - Качество.
6. **Характеристики проекта.**
   - Тип проекта.
   - Цель проекта.
   - Требования к качеству.
   - Требования к бюджету.
   - Требования по срокам завершения.

7. **Расходы, связанные с проектом.**
   - Прямые.
   - Непрямые.
8. **Общий обзор моделей и методологий процесса разработки.**
   - Фазы процесса:
     - определение требований;
     - проектирование;
     - конструирование («реализация», «кодирование»);
     - интеграция;
     - тестирование и отладка («верификация»);
     - инсталляция;
     - поддержка.
   - Водопадная модель.
   - Спиральная модель.
   - Итеративная модель:
     - Agile;
     - Scrum;
     - XP.
   - RUP.
   - MSF.
   - Анализ существующих моделей и методов.
9. **Подробнее о Scrum.**
   - Что такое Scrum?
   - Причины возникновения Scrum.
   - Роли в Scrum:
     - владелец продукта;
     - команда;
     - scrum-мастер.
   - Бэклог продукта:
     - что такое бэклог продукта;
     - как создавать бэклог;
     - как оценивать задачи в бэклоге;
     - что такое scrum-доска;
     - примеры создания бэклога.
   - Спринт:
     - что такое спринт;
     - планирование спринтов;

- ежедневный скрам;
- обзор спринта;
- ретроспективное собрание.

Практическое задание

Необходимо провести симуляцию работы команды по методологии Scrum. Например, это может быть так называемое скрам-лего.

Подробно здесь: [Scrum Simulation with LEGO Bricks](#).

Модуль 16

Экзамен