Тема занятия: Введение в язык программирования Python

1. Введение в Python

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Основные характеристики:

1) высокоуровневость
2) неявная строгая динамическая типизация
3) автоматическое управление памятью
4) интерпретируемость
5) поддержка ООП
6) функциональный
7) многопроцессорность, многопоточность, асинхронность

Детальнее о каждом пункте:

1) высокоуровневость

Категории языков программирования:

- К *высокоуровневым* относят языки программирования, которые созданы с расчетом на то, что их должны понимать люди. Они независимы: программистам не нужно знать, на каком оборудовании будет запускаться программа.
- *Среднеуровневые* языки служат как бы связующим звеном между аппаратной и программной частью компьютера. Они действуют на уровне абстракции.
- *Низкоуровневые* языки, в свою очередь, созданы для удовлетворения нужд конкретной компьютерной архитектуры и учитывают требования «железа».

2) неявная строгая динамическая типизация

Типизация – способ, с помощью которого язык программирования распознает типы переменных.

Базовые типы данных, которые есть во всех языках:

- целые числа
- числа с запятой
- строки
- булевые значения

Категории типизации:

- *Строгая (сильная)* типизация отличается тем, что в выражениях нельзя смешивать типы данных, неявного преобразования не происходит
- при *явной* типизации тип переменной, параметров функции надо указывать явно, при объявлении
- при *динамической* типизации тип переменной определяется во время выполнения программы
- *нестрогая (слабая)* допускает “автоматическое” преобразование типов, что сказывается на точности вычислений.
- при *неявной* типизации тип переменной, параметров функции не нужно указывать явно, при объявлении
- при *статической* типизации тип переменной известен до выполнения программы

3) автоматическое управление памятью

4) интерпретируемость

Компилятор – это компьютерная программа, которая переводит компьютерный код с одного языка программирования на другой. Компилятор берет программу целиком и преобразует ее в исполняемый компьютерный код. Для этого требуется целая программа, так как компьютер понимает только то, что написано двоичным кодом. Задача

компилятора — преобразовать исполняемую программу в машинный код, который и распознается компьютером.

Интерпретатор – это компьютерная программа, которая преобразует каждый программный оператор высокого уровня в машинный код. Сюда входят исходный код, предварительно скомпилированный код и сценарии. Интерпретатор представляет собой машинную программу, которая непосредственно выполняет набор инструкций без их компиляции.

5) поддержка ООП

ООП – объектно-ориентированное программирование – методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования. Данная методология включает в себя парадигмы программирования, благодаря которым можно легко структурировать и переиспользовать код.

6) функциональный

Функциональное программирование — парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних. Противопоставляется парадигме императивного программирования, которая описывает процесс вычислений как последовательное изменение состояний.

7) многопроцессорность, многопоточность, асинхронность

2. Установка интерпретатора Python

3. Установка среды разработки PyCharm

IDE (Integrated development Environment) – это настольный графический пользовательский интерфейс, который позволяет редактировать, выполнять, просматривать и отлаживать программы Python, используя единый интерфейс.

4. Pip

pip — система управления пакетами, которая используется для установки и управления программными пакетами, написанными на Python.

Python Package Index (PyPI) – это репозиторий, открытый для всех Python разработчиков, в нем вы можете найти пакеты для решения практически любых задач. Там также есть возможность выкладывать свои пакеты. Для скачивания и установки используется pip.

Основные команды (запускаются в командной строке):
pip help – получение справочной информации
pip install <название пакета> – установка определенного пакета
pip uninstall <название пакета> – удаление определенного пакета
pip freeze – получение списка всех установленных пакетов

5. Виртуальное окружение (venv)

Виртуальное окружение – своего рода песочниц, в рамках которых запускается приложение со своими библиотеками, обновление и изменение которых не затронет другие приложение, использующие те же библиотеки.

Инструменты для создания виртуальных окружений:

- *virtualenv*
- *venv*

Детальнее про venv:

1. Создание виртуального окружения
   python -m venv <название окружения>
2. Активация виртуального окружения
   source <название окружения>/bin/activate (Linux)
   <название окружения>\Scripts\activate.bat (Windows)
3. Деактивация виртуального окружения
   deactivate

6. Синтаксис

Основные моменты:

1. Первое, что, как правило, бросается в глаза, если говорить о синтаксисе в Python, это то, что *отступы имеют значение:*

- они определяют, какой код попадает в блок;
- когда блок кода начинается и заканчивается.

В качестве отступов могут использоваться табы или пробелы (лучше использовать пробелы, а точнее, настроить редактор так, чтобы таб был равен 4 пробелам – тогда при использовании клавиши табуляции будут ставиться 4 пробела, вместо 1 знака табуляции).

2. Названии переменных, функций и классов должно начинаться с алфавитного символа или со знака подчеркивания и может содержать алфавитно-цифровые символы и знак подчеркивания

3. Python - регистрозависимый язык, поэтому выражения print и Print или PRINT представляют разные выражения.

4. Для названия переменных и функций применяется стиль snake_case, для констант – UPPER_CASE_SNAKE_CASE, а для классов – PascalCase.

Стили наименования:

snake_case – чтобы писать в данном стиле, нужно просто заменить пробелы знаками подчеркивания. Все слова при этом пишутся строчными буквами (используется для переменных, функций).

PascalCase – каждое слово начинается с заглавной буквы (в отличие от camelCase, где первое слово начинается со строчной) (используется для классов).

UPPER_CASE_SNAKE_CASE – все слова пишутся заглавными буквами, а пробелы заменяются знаками подчеркивания (используется для констант).