УНИВЕРСИТЕТ СИНЕРГИЯ

ВСПОМИНАЕМ ОСНОВЫ

Продвинутый Python

Андрей Вильмов

О СПИКЕРЕ

Продвинутый Python

Андрей Вильмов

- Опыт программирования на Python более 6 лет
- Спикер крупнейшей профессиональной конференции для разработчиков высоконагруженных систем HighLoad++ 2023
- Проекты: разработка микросервисов при переходе с монолита, построение DataLake и DWH для компаний, разработка BI отчетности

СОДЕРЖАНИЕ ТЕМЫ

01 Функции. Методы. Лямбда функции. Как и для чего их использовать?

02 ООП. Классы. Наследование

03 Абстрактные классы

04 Принципы SOLID

05 Магические методы и шаблоны проектирования

ФУНКЦИЯ

это блоки кода, выполняющие определенные операции. Если требуется, функция может определять входные параметры, позволяющие вызывающим объектам передавать ей аргументы

ФУНКЦИИ И МЕТОДЫ

Функция создается через ключевое слово **def**

```
def название_функции(аргументы):
 тело функции
```

Телом функции называется код, который **выполняет функция**

Для того чтобы вернуть значение из функции, используется ключевое слово **return**

```
def my_sum(a, b):
 return a+b
```

ФУНКЦИИ И МЕТОДЫ

Чтобы вызвать функцию, достаточно написать название функции и передать параметры

```
def my_sum(a, b):
 return a+b

my_sum(2,3)//Вернет 5
my_sum(2) //Вернет ошибку
```

ФУНКЦИИ И МЕТОДЫ

Также, можно создать функцию, которая ничего не возвращает. Для этого просто не пишем слово **return**

```
•••
def my_print(print_str):
 print(print_str)

my_print(‘Hello World’) //Напечатает Hello World
```

ФУНКЦИИ И МЕТОДЫ

В функции можно **указать необязательные параметры**

```
•••
def my_print(print_str = ‘Hello’):
 print(print_str)

my_print()//Напечатает Hello
my_print(‘Hello World’) //Напечатает Hello World
```

ФУНКЦИИ И МЕТОДЫ

В качестве аргументов функция может принимать практически **все что угодно**

Например другую функцию

```
•••
def print_hello():
 return “Hello”

def print_bye():
 return “Bye”

def my_print(print_func):
 print(print_func())

my_print(print_hello)//Напечатает Hello
my_print(print_bye) //Напечатает Bye
```

Возвращаем строку

Обращаемся к переданной в параметры функции, чтобы ее вызвать

ФУНКЦИИ И МЕТОДЫ

Функцию можно **сохранить в переменную**

```
def my_print(print_str = ‘Hello’):
 print(print_str)

my_var = my_print
my_var ()//Напечатает Hello
my_var (‘Hello World’) //Напечатает Hello World
```

! **ВАЖНО**
Когда мы сохраняем функцию в переменную, после названия функции не нужно ставить круглые скобки

ЛЯМБДА ФУНКЦИЯ В PYTHON

это такие функции, которые не имеют названия. Их также называют анонимными. Определяется лямбда функции служебным словом lambda

ЛЯМБДА ФУНКЦИИ

Создание lambda функции

```
•••
lambda_func = lambda x: x + 1

lambda_func(5) //Вернет 6
```

ЛЯМБДА ФУНКЦИИ

Чаще всего lambda функции используют для фильтрации данных либо в методах обработки данных

```
employee = [
 {
    “Age”: 20,
    “Name”: “Станислав”
  },
  {
    “Age”: 5,
    “Name”: “Виталий”
   },
]
filter_ = lambda x: x[‘Age’] > 10
print(
   [
     item
     for item in employee
     if filter_(item)
   ]
) // Выведет [{'Age': 20, 'Name': 'Станислав'}]
```

ПОДВЕДЁМ ИТОГИ

- **Вы узнали** что такое функции, методы, ООП, классы и наследование
- **Познакомились** с лямбда функцией
- **Научились** работать с абстрактными классами
- **Разобрались** с принципами SOLID, магическими методами и шаблонами проектирования