

# Python Базовый

Инкапсуляция


# Python Базовый

## Introduction



**Бондаренко Кирилл**

Senior Data scientist, CreatorIQ

 [profile.php?id=100011447245832](https://www.facebook.com/profile.php?id=100011447245832)

 [kirill-bond/](https://www.linkedin.com/in/kirill-bond/)

 [@bond.kirill.alexandrovich](https://www.telegram.com/@bond.kirill.alexandrovich)



# Python Базовый

Тема урока

Инкапсуляция

# Python Базовый

## План урока

1. Что такое инкапсуляция
2. Как на практике она применима
3. Решение задач

# Python Базовый

## Инкапсуляция (Encapsulation)

**Инкапсуляция** - принцип, при котором пользователь использует только публичную или "интерфейсную" часть класса и не вникает в его внутреннюю реализацию.

Таким образом программист скрывает определенные данные, методы, атрибуты от последующих пользователей.



# Python Базовый

## Инкапсуляция в Python

```
1 class Worker:
2     RIGHTS = "Every worker has right to get payed" \
3         "for work he/she has done."
4
5     def __init__(self, work_class):
6         self.__class_salary_map = {
7             "A": 100,
8             "B": 200,
9             "C": 500,
10            "D": 1000
11        }
12        self.__salary = self.__calculate_salary(work_class)
13
14    def __calculate_salary(self, work_class):
15        return self.__class_salary_map.get(work_class, 0)
16
17    @property
18    def salary(self):
19        if not self.__salary:
20            return "Undefined"
21        return self.__salary
22
```

```
worker1 = Worker(work_class="A")
print(worker1.salary) # 100
worker2 = Worker(work_class="B")
print(worker2.salary) # 200
print(worker1.RIGHTS == worker2.RIGHTS) # True
print(Worker.RIGHTS) # Every worker ...
worker3 = Worker(work_class="X")
print(worker3.salary) # Undefined
```

```
100
200
True
Every worker has right to get payedfor work he/she has done.
Undefined
```

# Python Базовый

## Задачи

1. Написать класс `TextProcessor` для обработки текстовых данных. Класс должен иметь публичный метод `get_clean_string`, который удалит все знаки препинания из строки, которую в него передают аргументом и приватный метод `is_punctuation`, который непосредственно проверяет символ на равенство со знаками пунктуации и возвращает `True/False`, которые в свою очередь являются приватным или защищенным атрибутом класса.
2. Написать класс `TextLoader`, который имеет приватным атрибутом `text_processor` объект класса, что был создан в задаче 1. Новый класс будет иметь приватный атрибут `clean_string` и публичный метод `set_clean_text`, который будет вызывать метод класса `TextProcessor` через свой атрибут `text_processor` и записывать значение в `clean_string`. Сам же атрибут `clean_string` будет иметь `property` с дополнительным выводом в консоль того, что выводится уже очищенная строка.
3. Написать класс `DataInterface`, который будет иметь своим защищенным атрибутом объект класса `TextLoader` и публичный метод `process_texts`, который будет принимать список строк, в цикле обрабатывает каждую и выведет ее значение в консоль.

# Python Базовый

## Решение

```
1 class TextProcessor:
2     def __init__(self):
3         self._punks = ",.?!:;'&"
4
5     def __is_punkt(self, char):
6         return char in self._punks
7
8     def get_clean_string(self, string):
9         result = ""
10        for char in string:
11            if self.__is_punkt(char):
12                continue
13            result += char
14        return result
15
16
17 class TextLoader:
18     def __init__(self):
19         self.__text_processor = TextProcessor()
20         self.__clean_string = None
21
22     def set_clean_text(self, text):
23         self.__clean_string = self.__text_processor.get_clean_string(text)
24
25     @property
26     def clean_string(self):
27         print("Clean text is:")
28         return self.__clean_string
29
```

```
30 class DataInterface:
31     def __init__(self):
32         self.__loader = TextLoader()
33
34     def process_texts(self, list_of_texts):
35         for text in list_of_texts:
36             self.__loader.set_clean_text(text)
37             print(self.__loader.clean_string)
38
39
40 if __name__ == "__main__":
41     interface = DataInterface()
42     data = ["Hello, world !", "Hi! My name is John."]
43     interface.process_texts(data)
44
```



# Информационный видеосервис для разработчиков программного обеспечения



# Проверка знаний

TestProvider.com



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://TestProvider.com)

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

# Python Базовый

Спасибо за внимание! До новых встреч!



Бондаренко Кирилл  
Senior Data scientist, CreatorIQ

