

Федеральное государственное автономное образовательное учреждение  
высшего образования

«ОМСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Прикладная математика и фундаментальная информатика»

Практическое занятие №10\_1

по дисциплине «Практикум по программированию»

на тему: «Основы ООП в Python»

Вариант №14

Выполнил

Студент гр. **ФИТ-212**

группа

**Курпенов К.И.**

Фамилия И.О. студента

подпись

Принял:

Преподаватель

**Моисеева Н.А.**

Фамилия И.О. преподавателя

дата, подпись

Омск 2022

## ЗАДАНИЕ 1.

### Условие:

#### ЧАСТЬ I Графическая библиотека + функция

Капли из ландышей очень способствуют успокоению коротышек, они перестают драться по любому поводу. Знайка предложил доктору Пилюлькину поставить их производство на поток. Для этого нужно перепрограммировать машину, собирающую землянику, на сбор ландышей. К счастью, для этого нужно только показать машине образец цвета ландыша, а она определит направление движения самостоятельно по следующему алгоритму: в каждом столбце пикселей изображения вычисляется их количество заданного цвета, затем выбирается столбец с самым большим количеством пикселей цвета ландыша (если таких несколько, выбирается наименьший по номеру) и вычисляется значение смещения по формуле:

$$X * 1001 // width - 500$$
 (X — номер выбранного столбца, а width — ширина изображения).

Напишите функцию `search_for_lilies(image, color)`, принимающую объект изображения и цвет ландышей, а возвращающую вычисленное по формуле смещение.

### Решение:

```
1 from PIL import Image
2
3
4 def search_for_lilies(path: str, color: tuple) -> int:
5     with Image.open(path) as image:
6         pixels = image.load()
7         x, y = image.size
8
9         colors = [0 for _ in range(x)]
10
11         for i in range(x):
12             for j in range(y):
13                 rgb = pixels[i, j]
14                 if rgb == color:
15                     colors[i] += 1
16
17         index = colors.index(max(colors))
18
19         return index * 1001 // x - 500
20
21
22 if __name__ == "__main__":
23     path = input("> Enter path to image: ")
24     color = tuple(map(int, input("> Enter RGB-form color: ").split()))
25
26     print(search_for_lilies(path, color))
27
```

## ЗАДАНИЕ 2.

### Задача 2.1.

#### Условие:

Необходимо реализовать код, показанный в примерах выше. Создать класс Rectangle и класс ToyClass. Для ToyClass необходимо добавить три атрибута, и метод который устанавливает их.

#### Решение:

```
1 class ToyClass:
2     def __init__(self, p1: float, p2: list, p3: str) -> None:
3         self.parameter_1 = p1
4         self.parameter_2 = p2
5         self.parameter_3 = p3
6
7     def show(self) -> None:
8         print(self.parameter_1)
9         print(self.parameter_2)
10        print(self.parameter_3)
11
12    def instance_method(self) -> str:
13        return f"[+] Instance method called: {self}"
14
15    @classmethod
16    def class_method(cls) -> str:
17        return f"[+] Class method called: {cls}"
18
19    @staticmethod
20    def static_method():
21        return f"[+] Static method called"
22
23
24 if __name__ == "__main__":
25     toy = ToyClass(3.1415926, ["List"], "String")
26     toy.show()
27
```

## Задача 2.2.

### Условие:

Напишите программу с классом Student, в котором есть три атрибута: name, groupNumber и age. По умолчанию name = Ivan, age = 18, groupNumber = 10A. Необходимо создать пять методов: getName, getAge, getGroupNumber, setNameAge, setGroupNumber. Метод getName нужен для получения данных об имени конкретного студента, метод getAge нужен для получения данных о возрасте конкретного студента, метод setGroupNumber нужен для получения данных о номере группы конкретного студента. Метод setNameAge позволяет изменить данные атрибутов установленных по умолчанию, метод setGroupNumber позволяет изменить номер группы установленный по умолчанию. В программе необходимо создать пять экземпляров класса Student, установить им разные имена, возраст и номер группы.

### Решение:

```
1 class Student:
2     def __init__(self, name="Ivan", group="10a", age=18) -> None:
3         self.name = name
4         self.group = group
5         self.age = age
6
7     def getName(self) -> str:
8         return self.name
9
10    def getGroup(self) -> str:
11        return self.group
12
13    def getAge(self) -> int:
14        return self.age
15
16    def setName(self, name: str) -> None:
17        self.name = name
18
19    def setGroup(self, group: str) -> None:
20        self.group = group
21
22    def setAge(self, age: int) -> None:
23        self.age = age
24
25
26 if __name__ == "__main__":
27     student_1 = Student("Kuat", "FIT-212", 18)
28     student_2 = Student("Alexey", "FIT-212", 18)
29     student_3 = Student("Vladimir", "FIT-212", 18)
30     student_4 = Student("Ivan", "FIT-212", 18)
31     student_5 = Student("Artyom", "IVT-211", 18)
32
```

### Задача 2.3.

#### Условие:

Напишите программу с классом Math. Создайте два атрибута — a и b. Напишите методы addition — сложение, multiplication — умножение, division — деление, subtraction — вычитание. При передаче в методы параметров a и b с ними нужно производить соответствующие действия и печатать ответ.

#### Решение:

```
1 class Math:
2     def __init__(self, a: float, b: float) -> None:
3         self.a = a
4         self.b = b
5
6     def addition(self) -> float:
7         return self.a + self.b
8
9     def subtraction(self) -> float:
10        return self.a - self.b
11
12    def multiplication(self) -> float:
13        return self.a * self.b
14
15    def division(self) -> float:
16        return self.a / self.b
17
18
19 if __name__ == "__main__":
20     math = Math(8, 2)
21
22     print(math.addition())
23     print(math.subtraction())
24     print(math.multiplication())
25     print(math.division())
26
```

## Задача 2.4.

### Условие:

Напишите программу с классом Car. Создайте конструктор класса Car. Создайте атрибуты класса Car — color (цвет), type (тип), year (год). Напишите пять методов. Первый — запуск автомобиля, при его вызове выводится сообщение «Автомобиль заведен». Второй — отключение автомобиля — выводит сообщение «Автомобиль заглушен». Третий — присвоение автомобилю года выпуска. Четвертый метод — присвоение автомобилю типа. Пятый — присвоение автомобилю цвета.

### Решение:

```
1 class Car:
2     def __init__(self, color: str, t: str, year: int) -> None:
3         self.color = color
4         self.type = t
5         self.year = year
6
7     def start(self) -> None:
8         print("[+] Car was started")
9
10    def mute(self) -> None:
11        print("[-] Car was muted")
12
13    def setYear(self, year: int) -> None:
14        self.year = year
15
16    def setType(self, t: str) -> None:
17        self.type = t
18
19    def setColor(self, color: str) -> None:
20        self.color = color
21
22
23 if __name__ == "__main__":
24     car = Car("UAZ Patriot", "Auto", 2018)
25
```