

Практическое занятие № 16

Тема: составление программ с использованием ООП.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с ООП в IDE PyCharm Community.

Постановка 1 задачи:

Создайте класс "Компьютер" с атрибутами "марка", "процессор" и "оперативная память". Напишите метод, который выводит информацию о компьютере в формате "Марка: марка, Процессор: процессор, Оперативная память: память".

Тип алгоритма: линейный

Текст программы:

```
import pickle

class Computer:
    def __init__(self, brand, processor, ram):
        self.brand = brand
        self.processor = processor
        self.ram = ram

    def __str__(self):
        return self.get_info()

    def get_info(self):
        return f'Марка {self.brand}, процессор {self.processor}, оперативная память {self.ram}'
```

Постановка 2 задачи:

Создайте класс "Человек", который содержит информацию о имени, возрасте и поле. Создайте классы "Мужчина" и "Женщина", которые наследуются от класса "Человек". Каждый класс должен иметь метод, который выводит информацию о поле объекта.

Тип алгоритма: линейный

Текст программы:

```
class Human:
    def __init__(self, name, age, gender):
        if gender not in ('Мужчина', 'Женщина'):
            raise ValueError('gender must be "Мужчина" or "Женщина"')
        self.name = name
        self.age = age
        self.gender = gender

    def __str__(self):
        return self.get_info()

    def get_info(self):
        return f'{self.name}, {self.age} лет, {self.gender}'
```

```
class Man(Human):
    def __init__(self, name, age):
        super().__init__(name, age, 'Мужчина')
```

```
class Woman(Human):
    def __init__(self, name, age):
        super().__init__(name, age, 'Женщина')
```

Постановка 3 задачи:

Для задачи из блока 1 создать две функции, `save_def` и `load_def`, которые позволяют сохранять информацию из экземпляров класса (3 шт.) в файл и загружать ее обратно. Использовать модуль `pickle` для сериализации и десериализации объектов Python в бинарном формате.

Тип алгоритма: линейный

Текст программы:

```
import pickle
```

```
class Computer:
    def __init__(self, brand, processor, ram):
        self.brand = brand
        self.processor = processor
        self.ram = ram

    def __str__(self):
        return self.get_info()

    def get_info(self):
        return f'Марка {self.brand}, процессор {self.processor}, оперативная память {self.ram}'

    def save(self, filename):
        with open(filename, 'wb') as f:
            pickle.dump(self, f)

    @classmethod
    def load(cls, filename):
        with open(filename, 'rb') as f:
            instance = pickle.load(f)
        return instance
```

Вывод:

В процессе выполнения практического задания я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с ООП в IDE PyCharm Community.