

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 задачи.

Создайте класс "Человек" с атрибутами "имя", "возраст" и "пол". Напишите метод, который выводит информацию о человеке в формате "Имя: имя, Возраст: возраст, Пол: пол".

Текст программы:

```
class Human:
    def __init__(self, name: str, age: int, gender: str) -> None:
        self.name = name
        self.age = age
        self.gender = gender

    def info(self) -> None:
        print(f"Имя: {self.name}, Возраст: {self.age}, Пол: {self.gender}")

test_human = Human(name="Test", age=100500, gender="F")
test_human.info()
```

Протокол работы программы:

Имя: Test, Возраст: 100500, Пол: F

Process finished with exit code 0

Постановка задачи.

Создание базового класса "Животное" и его наследование для создания классов "Собака" и "Кошка". В классе "Животное" будут общие методы, такие как "дышать" и "питаться", а классы-наследники будут иметь свои уникальные методы и свойства, такие как "гавкать" и "мурлыкать".

Текст программы:

```
class Animal:
    def __init__(self, nickname: str, sex: str, old: int):
        self.nickname = nickname
        self.sex = sex
        self.old = old

    def breathe(self):
        print(f'{self.nickname} дышит')

    def eat(self):
        print(f'{self.nickname} питается')

    def get_info(self):
        print(f'Кличка: {self.nickname}\nПол: {self.sex}\nВозраст: {str(self.old)}')

class Dog(Animal):
    def __init__(self, nickname: str, sex: str, old: int):
        super().__init__(nickname, sex, old)

    def bark(self):
        print(f'{self.nickname} гавкает')

class Cat(Animal):
    def __init__(self, nickname: str, sex: str, old: int):
        super().__init__(nickname, sex, old)

    def purr(self):
        print(f'{self.nickname} мурлычет')

if __name__ == '__main__':
    my_dog = Dog(nickname='Шарик', sex='Самец', old=8)
    my_cat = Cat(nickname='Матроскин', sex='Самец', old=5)

    my_cat.get_info()
    print()
    my_dog.get_info()
    print()
```

```
my_dog.breathe()
my_cat.purr()
my_dog.bark()
my_cat.eat()
```

Протокол работы программы:

Кличка: Матроскин

Пол: Самец

Возраст: 5

Кличка: Шарик

Пол: Самец

Возраст: 8

Шарик дышит

Матроскин мурлычет

Шарик гавкает

Матроскин питается

Process finished with exit code 0

Постановка 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.

Текст программы:

```
import pickle

class Human:
    def __init__(self, name: str, age: int, gender: str) -> None:
        self.name = name
        self.age = age
        self.gender = gender

    def info(self) -> None:
        print(f"Имя: {self.name}, Возраст: {self.age}, Пол: {self.gender}")
```

```
def save_def(instance: Human, filepath: str) -> None:
    with open(filepath, "wb") as file:
        pickle.dump(obj=instance, file=file)

def load_def(filepath: str) -> Human:
    with open(filepath, "rb") as file:
        human_instance = pickle.load(file=file)
    return human_instance

if __name__ == '__main__':
    test_human = Human(name="Test", age=100500, gender="F")
    vasya = Human(name="Василий", age=25, gender="M")
    ann = Human(name="Анна", age=19, gender="Ж")

    save_def(instance=test_human, filepath="./saves/test_human.pickle")
    save_def(instance=vasya, filepath="./saves/vasya.pickle")
    save_def(instance=ann, filepath="./saves/ann.pickle")

    test_human_l = load_def(filepath="./saves/test_human.pickle")
    vasya_l = load_def(filepath="./saves/vasya.pickle")
    ann_l = load_def(filepath="./saves/ann.pickle")

    test_human_l.info()
    vasya_l.info()
    ann_l.info()
```

Протокол работы программы:

Имя: Test, Возраст: 100500, Пол: F

Имя: Василий, Возраст: 25, Пол: M

Имя: Анна, Возраст: 19, Пол: Ж

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрёл навыки составления программ с ООП в IDE PyCharm Community. Были использованы языковые конструкции class, def. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.