Практическая №16 Кайханиди Виктор ИС-27

Тема: составленеи 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 блока задания №1:
Создай класс "Человек" с атрибутами "имя", "возраст" и "пол".
Напиши метод который выводит информацию о человеке в формате Имя: имя, Возраст: возраст, Пол: пол".

Текст программы:

```
class Human:
    def __init__(self, name, age, gender):
        self.name = name
        self.age = age
        self.gender = gender

    def display_info(self):
        return f'Имя: {self.name}, Возраст: {self.age}, Пол: {self.gender}'


person = Human("Alex", 30, "Мужской")

print(person.display_info())
```

Протокол работы программы:

```
Имя: Alex, Возраст: 30, Пол: Мужской

Process finished with exit code 0
```

Постановка задачи блока задания №2:
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, name):
        self.name = name

    def breathe(self):
        return f'{self.name} дышит'

    def eat(self):
        return f'{self.name} питается'


class Dog(Animal):
    def bark(self):
        return f'{self.name} гавкает'
```

```
class Cat(Animal):
    def purr(self):
        return f'{self.name} мурлыкает'

dog = Dog('Шарик')
print(dog.breathe())
print(dog.bark())

cat = Cat('Мурзик')
print(cat.eat())
print(cat.purr())
```

Протокол работы программы:
Шарик дышит
Шарик гавкает
Мурзик питается
Мурзик мурлыкает

Process finished with exit code 0

Постановка задачи блока #3:
Для задачи из блока 1 создать две функции save_def и load_def, которые позволяют сохранять информацию из экземпляров класса (3 шт.) n фал и загружать её обратно. Использовать модуль pickle сериализации и дееариализации объектов Python в бинарном формате.

Текст программы:

```
import pickle
class MyClass:
    def __init__(self, name):
        self.name = name
def save_def(obj, filename):
    with open(filename, 'wb') as output:  # Overwrites any existing file.
        pickle.dump(obj, output, pickle.HIGHEST_PROTOCOL)
def load_def(filename):
    with open(filename, 'rb') as input:
        obj = pickle.load(input)
    return obj
# создание экземпляров класса
obj1 = MyClass('object1')
obj2 = MyClass('object2')
obj3 = MyClass('object3')
# сохранение объектов
```

```
save_def([obj1, obj2, obj3], 'objs.pkl')
# загрузка объектов
loaded_objs = load_def('objs.pkl')
for obj in loaded_objs:
    print(obj.name)
```

Постановка работы программы:
object1
object2
object3

Process finished with exit code 0

Вывод: я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрёл навыки составления программ с ООП в IDE PyCharm Community.