Студентка группы ИС-22 Калашникова А. Е.

Практическое занятие №16

Тема:составление программ с использованием ООП.

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

Постановка задачи.

Задание 1.

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.

```
# 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".

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

    def get_info(self):
        return f"Марка: {self.marca}, Процессор: {self.processor}, Оперативная память: {self.ram}"


computer = Computer("Dell", "Intel Core i7", "16GB")
print(computer.get_info())
```

Протокол работы программы.

/usr/bin/python3.9 /home/student/Документы/clone/PZ16/pz16_1.py

Марка: Dell, Процессор: Intel Core i7, Оперативная память: 16GB

Process finished with exit code 0

Задание 2.

Создайте базовый класс "Человек" со свойствами "имя", "возраст" и "пол". От этого класса унаследуйте классы "Мужчина" и "Женщина" и добавьте в них свойства, связанные с социальным положением (например, "семейное положение", "количество детей" и т.д.).

Тип алгоритма.

Линейный.

Текст программы.

```
# Создайте базовый класс "Человек" со свойствами "имя", "возраст" и "пол". От этого
# класса унаследуйте классы "Мужчина" и "Женщина" и добавьте в них свойства,
# связанные с социальным положением (например, "семейное положение",
# "количество детей" и т.д.).
class Person:
    def __init__(self, name, age, gender):
        self.name = name
        self.age = age
        self.gender = gender

class Man(Person):
    def __init__(self, name, age, gender, marital_status, children_count):
        super().__init__(name, age, gender)
        self.marital_status = marital_status
        self.children_count = children_count

class Woman(Person):
    def __init__(self, name, age, gender, marital_status, children_count):
        super().__init__(name, age, gender)
        self.marital_status = marital_status
        self.children_count = children_count

# Example usage
man = Man("John", 30, "Male", "Married", 2)
woman = Woman("Alice", 28, "Female", "Single", 0)

print(f"Name: {man.name}, Age: {man.age}, Gender: {man.gender}, Marital Status: {man.marital_status}, Children Count: {man.children_count}")
print(f"Name: {woman.name}, Age: {woman.age}, Gender: {woman.gender}, Marital Status: {woman.marital_status}, Children Count:
{woman.children_count}")
```

Протокол работы программы.

/usr/bin/python3.9 /home/student/Документы/clone/PZ16/pz16_2.py

Name: John, Age: 30, Gender: Male, Marital Status: Married, Children Count: 2

Name: Alice, Age: 28, Gender: Female, Marital Status: Single, Children Count: 0

Process finished with exit code 0

Задание 3.

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.

```
# 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.
import pickle


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

    def get_info(self):
        return f"Марка: {self.marca}, Процессор: {self.processor}, Оперативная память: {self.ram}"


def save_def(computers):
    with open('computers_data.pkl', 'wb') as file:
        pickle.dump(computers, file)

def load_def():
    with open('computers_data.pkl', 'rb') as file:
        return pickle.load(file)

computer1 = Computer("HP", "Intel Core i5", "8GB")
computer2 = Computer("Dell", "AMD Ryzen 7", "16GB")
computer3 = Computer("Apple", "Apple M1", "32GB")

computers_list = [computer1, computer2, computer3]
save_def(computers_list)

loaded_computers = load_def()

for computer in loaded_computers:
    print(computer.get_info())
```

Протокол работы программы.

/usr/bin/python3.9 /home/student/Документы/clone/PZ16/pz16_3.py

Марка: HP, Процессор: Intel Core i5, Оперативная память: 8GB

Марка: Dell, Процессор: AMD Ryzen 7, Оперативная память: 16GB

Марка: Apple, Процессор: Apple M1, Оперативная память: 32GB

Process finished with exit code 0

Вывод: закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ с ООП в IDE PyCharm Community.