Студентка группы ИС-25 Полькина.В

**Отчет о практическом занятии**

**Практическое занятие №16**

**Тема:** Составление программ c использованием ООП в IDE PyCharm Community

**Цель:** Закрепить полученные знания , понятия , алгоритмы, основные принципы составления программ, приобрести навыки составления программ с c использованием ООП в IDE PyCharm Community

**Постановка 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: оперативная память"

**Тип алгоритма:** цикличный

**Текст программы:**

# 24 вариант  
# 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: оперативная память"  
  
class Computer:  
 def \_\_init\_\_(self, brand, processor, ram):  
 self.brand = brand  
 self.processor = processor  
 self.ram = ram  
  
 def display\_info(self):  
 print(f"Марка: {self.brand}, Процессор: {self.processor}, Оперативная память: {self.ram}")  
  
  
my\_computer = Computer("HP", "Intel i5", "16GB")  
my\_computer.display\_info()

**Протокол работы программы:**

**C:\Users\Viktoria\AppData\Local\Programs\Python\Python311\python.exe "C:\POLKINA Viktoria\clone25\PZ\_16\pz16.1.py"**

**Марка: HP, Процессор: Intel i5, Оперативная память: 16GB**

Process finished with exit code 0

**Постановка задачи2 :**

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 субъекта.

**Тип алгоритма:** цикличный

**Текст программы:**

# 24  
# 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 субъекта.  
  
class Person:  
 def \_\_init\_\_(self, name, age, gender):  
 self.name = name  
 self.age = age  
 self.gender = gender  
  
 def display\_gender(self):  
 print(f"The person's gender is {self.gender}")  
  
  
class Man(Person):  
 def \_\_init\_\_(self, name, age):  
 super().\_\_init\_\_(name, age, "Male")  
  
 def display\_gender(self):  
 print(f"The man's gender is {self.gender}")  
  
  
class Woman(Person):  
 def \_\_init\_\_(self, name, age):  
 super().\_\_init\_\_(name, age, "Female")  
  
 def display\_gender(self):  
 print(f"The woman's gender is {self.gender}")  
  
  
john = Man("John", 30)  
jane = Woman("Jane", 25)  
  
john.display\_gender()  
jane.display\_gender()

**Протокол работы программы:**

C:\Users\Viktoria\AppData\Local\Programs\Python\Python311\python.exe "C:\POLKINA Viktoria\clone25\PZ\_16\pz16.2.py"

The man's gender is Male

The woman's gender is Female

Process finished with exit code 0

**Постановка задачи3:**

Для задачи из блока 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: оперативная память") создать 2 функции , save\_def и load\_def , которые позволяют сохранять информацию из экземпляров класса (3шт) в файл и загружать ее обратно. Использовать модуль pickle для сериализации и дессериализации объектов python в бинарном формате

**Тип алгоритма:** цикличный

**Текст программы:**

# 24  
# Для задачи из блока 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: оперативная память") создать 2 функции ,  
# save\_def и load\_def , которые позволяют сохранять информацию из экземпляров класса (3шт) в файл  
# и загружать ее обратно. Использовать модуль pickle для сериализации и дессериализации объектов python  
# в бинарном формате  
  
import pickle  
  
  
class Computer:  
 def \_\_init\_\_(self, brand, processor, ram):  
 self.brand = brand  
 self.processor = processor  
 self.ram = ram  
  
 def display\_info(self):  
 print(f"Марка: {self.brand}, Процессор: {self.processor}, Оперативная память: {self.ram}")  
  
  
def save\_def(computers):  
 with open('computers.pkl', 'wb') as file:  
 pickle.dump(computers, file)  
  
  
def load\_def():  
 try:  
 with open('computers.pkl', 'rb') as file:  
 return pickle.load(file)  
 except FileNotFoundError:  
 return []  
  
  
computer1 = Computer("Lenovo", "Intel Core i5", "8GB")  
computer2 = Computer("HP", "Intel Core i5", "16GB")  
computer3 = Computer("Dell", "Intel Core i7", "12GB")  
  
  
computers\_list = [computer1, computer2, computer3]  
save\_def(computers\_list)  
  
  
loaded\_computers = load\_def()  
  
  
for computer in loaded\_computers:  
 computer.display\_info()

**Протокол работы программы:**

C:\Users\Viktoria\AppData\Local\Programs\Python\Python311\python.exe "C:\POLKINA Viktoria\clone25\PZ\_16\pz16.3.py"

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

Марка: HP, Процессор: Intel Core i5, Оперативная память: 16GB

Марка: Dell, Процессор: Intel Core i7, Оперативная память: 12GB

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятия выработала навыки составления программ с использование ООП в IDE PyCharm Community. Были использованы языковые конструкци, int(), print ,– вывод полученного значения, метод pickle, работа с наследованием и элементами OOP

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub.