

Отчет о практическом занятии

Практическое занятие №16

Тема: Составление программ с использованием ООП в IDE PyCharm Community

Цель: Закрепить полученные знания , понятия , алгоритмы, основные принципы составления программ, приобрести навыки составления программ с 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: циклический

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: циклический

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 десериализации объектов python в бинарном формате

Тип алгоритма: циклический

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
десериализации 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 использованием ООП в IDE PyCharm Community. Были использованы языковые конструкции, `int()`, `print`, – вывод полученного значения, метод `pickle`, работа с наследованием и элементами ООП

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

Готовые программные коды выложены на GitHub.