

Отчёт

Практическое занятие № 16

Тема: составление программ для работы с текстовыми файлами в IDE PyCharm Community.

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

Постановка задачи 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 1: с использованием ООП

Текст программы 1:

```
class Computer:
    def __init__(self, brand, processor, ram):
        self.brand=brand
        self.processor=processor
        self.ram=ram
    def vivod(self):
        return (f'Компьютер: {self.brand}, Процессор: {self.processor},
Оперативная память: {self.ram}')
comp1=Computer('lenovo','intel','8ГБ')
comp2=Computer('apple','intel','10ГБ')
comp3=Computer('hp','proc','6ГБ')

print(comp1.vivod())
```

Протокол работы программы 1:

Компьютер: lenovo, Процессор: intel, Оперативная память: 8ГБ

Process finished with exit code 0

Постановка задачи 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

Тип алгоритма 2: с использованием ООП

Текст программы 2:

```
class Human:
    def __init__(self, name, age, sex):
        self.name=name
        self.age=age
        self.sex=sex
class Man(Human):
    def __init__(self, name, age, sex, married, kids, work):
        Human.__init__(self, name, age, sex)
        self.married=married
        self.kids=kids
        self.work=work
class Woman(Human):
    def __init__(self, name, age, sex, married, kids, work):
        Human.__init__(self, name, age, sex)
        self.married=married
        self.kids=kids
        self.work=work
katya=Human('Катя', 22, 'жен')
ivan=Man('Иван', 34, 'муж', True, 2, 'продавец')
print(katya.__dict__)
print(ivan.__dict__)
```

Протокол работы программы 2:

{'name': 'Катя', 'age': 22, 'sex': 'жен'}

{'name': 'Иван', 'age': 34, 'sex': 'муж', 'married': True, 'kids': 2, 'work': 'продавец'}

Process finished with exit code 0

Постановка задачи 3

Для задачи из блока 1 создать две функции, `safe_def` и `load_def`, которые позволяют сохранять информацию из экземпляров класса (3 шт.) в файл и загружать её обратно. Использовать модуль `pickle` для серелизации и десерелизации объектов Python в бинарном формате.

Тип алгоритма 3: с использованием ООП

Текст программы 3:

```
import pickle
from PZ_16_1 import comp1, comp2, comp3
```

```
info=[comp1.__dict__,comp2.__dict__,comp3.__dict__]
def save_def():
    with open ('Computer.bin','wb') as file:
        pickle.dump(info,file)
def load_def():
    with open ('Computer.bin','rb') as file:
        pickle.load(file)

save_def()
load_def()
```

Протокол работы программы 3:

Компьютер: lenovo, Процессор: intel, Оперативная память: 8ГБ

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ с использованием ООП в IDE PyCharm Community. Были использованы операторы with as, def, import, классы и модуль pickle ,была проведена проверка на исключения.

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

Готовые программные коды выложены на GitHub.