

Практическая занятие №16

Тема: Составление программ с классами и с ООП и пикл в IDE PyCharm Community.

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

Задание 1

Постановка задачи.

Создайте класс «Счетчик», который имеет атрибут текущего значения и методы для инкремента и декремента значения.

Текст программы:

```
class Counter:

    def __init__(self, znachenie):
        self.znachenie = znachenie

    def increment(self):
        self.znachenie += 1

    def uncriment(self):
        self.znachenie -= 1

Counter = Counter(10)
print(Counter.znachenie)

Counter.increment()
print(Counter.znachenie)

Counter.uncriment()
print(Counter.znachenie)
```

Протокол работы программы:

10
11
10

Process finished with exit code 0

Задание 2

Постановка задачи.

Создание базового класса "Работник" и его наследование для создания классов "Менеджер" и "Инженер".

В классе "Работник" будут общие методы, такие как "работать" и "получать зарплату",

а классы-наследники будут иметь свои уникальные методы и свойства, такие как "управлять командой" и "проектировать системы".

Текст программы:

```
class Rabotnik:
    def __init__(self, name, age, zp):
        self.name = name
        self.age = age
        self.zp = zp

    def rabotat(self):
        print(f"{self.name} работает")

    def get_zp(self):
        print(f"{self.name} получает зарплату в размере {self.zp}")

class Meneger(Rabotnik):
    def ypravlit_comandai(self):
        print(f"{self.name} управляет командой")

class Inginer(Rabotnik):
    def proectirovat_sistemy(self):
        print(f"{self.name} проектирует системы")

# Пример использования классов
Rabotnik1 = Rabotnik("Иван", 30, 50000)
Rabotnik1.rabotat()
Rabotnik1.get_zp()

Meneger1 = Meneger("Мария", 35, 60000)
Meneger1.rabotat()
Meneger1.get_zp()
Meneger1.ypravlit_comandai()

Inginer1 = Inginer("Петр", 28, 55000)
Inginer1.rabotat()
Inginer1.get_zp()
```

```
Inginer1.proectirovat_sistemy()
```

Протокол работы программы:

Иван работает

Иван получает зарплату в размере 50000

Мария работает

Мария получает зарплату в размере 60000

Мария управляет командой

Петр работает

Петр получает зарплату в размере 55000

Петр проектирует системы

Process finished with exit code 0

Задание 3

Постановка задачи.

```
"""
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.
"""
```

Текст программы:

```
import pickle

class Counter:
    def __init__(self, znachenie):
        self.znachenie = znachenie

    def increment(self):
        self.znachenie += 1

    def uncriment(self):
        self.znachenie -= 1

def save_def(obj, filename):
```

```
with open(filename, 'wb') as file:
    pickle.dump(obj, file)

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

counter1 = Counter(10)
counter2 = Counter(5)
counter3 = Counter(20)

counter1.incriment()
counter2.incriment()
counter3.incriment()

counter1.uncriment()
counter2.uncriment()
counter3.uncriment()

save_def(counter1, 'counter1.pkl')
save_def(counter2, 'counter2.pkl')
save_def(counter3, 'counter3.pkl')

loaded_counter1 = load_def('counter1.pkl')
loaded_counter2 = load_def('counter2.pkl')
loaded_counter3 = load_def('counter3.pkl')

print(loaded_counter1.znachenie)
print(loaded_counter2.znachenie)
print(loaded_counter3.znachenie)
```

Протокол работы программы:

10

5

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с классами и с ООП и пикл в IDE PyCharm Community.

Готовые программные коды выложены на Github.