

Практическое занятие № 16

Тема: составление программ с использованием ООП в IDE PyCharm Community

Цели практического занятия: Цели практического занятия: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с ООП в IDE PyCharm Community

Постановка задачи №1.

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

```
import pickle

class Counter:

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

    @staticmethod
    def incr(val):
        val += 1
        return val

    @staticmethod
    def decr(val):
        val -= 1
        return val

def save_def(val, file):
    with open(file, 'wb') as f:
        pickle.dump(val, f)

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

a = Counter(0)
```

```
b = Counter(1)
c = Counter(2)

val_info = [a, b, c]

for val in val_info:
    save_def(val, 'values.pkl')
    values = load_def('values.pkl')
    print(values.incr(values.curr_val))
    print(values.decr(values.curr_val), '\n')
```

Протокол работы программы:

1
-1

2
0

3
1

Process finished with exit code 0

Постановка задачи №2.

Создайте класс "Автомобиль", который содержит информацию о марке, модели и годе выпуска. Создайте класс "Грузовик", который наследуется от класса "Автомобиль" и содержит информацию о грузоподъемности. Создайте класс "Легковой автомобиль", который наследуется от класса "Автомобиль" и содержит информацию о количестве пассажиров.

Тип алгоритма: Линейный

Текст программы:

```
class Auto:
    def __init__(self, manuf, model, year):
        self.manuf = manuf
        self.model = model
        self.year = year

mashina = Auto('Koenigsegg', 'Agera', 2011)

class Truck(Auto):
    def __init__(self, manuf, model, year, lift):
        super().__init__(manuf, model, year)
        self.lift = lift

truck = Truck('Kamaz', 'Kakoito', 2000, '1500kg')

class Car(Auto):
    def __init__(self, manuf, model, year, passan):
        super().__init__(manuf, model, year)
        self.passan = passan

car = Car('Lada', 'Priora', 2005, 4)
```

Протокол работы программы:

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с регулярными выражениями в IDE PyCharm Community. Были использованы языковые конструкции `open`, `print`, `for`.

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

Готовые программные коды выложены на [GitHub](#).