

Практическая работа № 16.

Тема: составление программ с использованием ООП.

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

Задание 1.

Постановка задачи.

Создайте класс "Машина" с атрибутами "марка", "модель" и "год выпуска".

Напишите метод, который выводит информацию о машине в формате "Марка: марка, Модель: модель, Год выпуска: год".

Текст программы:

```
class Car:
    def __init__(self, mark, model, year):
        self.mark = mark
        self.model = model
        self.year = year

    def info(self):
        print(f"Марка: {self.mark}, Модель: {self.model}, Год выпуска: {self.year}")

car1 = Car('Lada', 'Granta', 2012)
car1.info()
```

Протокол работы программы:

"Марка: Lada, Модель: Granta, Год выпуска: 2012

Process finished with exit code 0

Задание 2.

Постановка задачи.

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.

Текст программы:

```
class Avtomobil:
    def __init__(self, mark, model, year):
        self.mark = mark
        self.model = model
        self.year = year

class Gruzovik(Avtomobil):
    def __init__(self, mark, model, year, груз_pod):
        Avtomobil.__init__(self, mark, model, year)
        self.груз_pod = груз_pod

class Legkovoi(Avtomobil):
    def __init__(self, mark, model, year, кол_pas):
        Avtomobil.__init__(self, mark, model, year)
        self.кол_pas = кол_pas

avto1 = Avtomobil('Lada', 'Granta', 2013)
```

```
avto2 = Gruzovik('ЗИЛ', 130, 2010, 7500)
avto3 = Legkovoi('BMW', 'X5', 2017, 5)

print(avto1.__dict__)
print(avto2.__dict__)
print(avto3.__dict__)
```

Протокол работы программы:

```
{'mark': 'Lada', 'model': 'Granta', 'year': 2013}
{'mark': 'ЗИЛ', 'model': 130, 'year': 2010, 'gruz_pod': 7500}
{'mark': 'BMW', 'model': 'X5', 'year': 2017, 'col_pas': 5}
```

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 Car:
    def __init__(self, mark, model, year):
        self.mark = mark
        self.model = model
        self.year = year

def save_def(info_cars, fname):
    with open(fname, 'wb') as f:
        pickle.dump(info_cars, f)

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

    def info(self):
        print(f"Марка: {self.mark}, Модель: {self.model}, Год выпуска: {self.year}")

car1 = Car('Toyota', 'Corolla', 2015)
car2 = Car('Honda', 'Civic', 2018)
car3 = Car('Ford', 'Focus', 2012)

info_cars = [car1, car2, car3]
save_def(info_cars, 'cars.pkl')

loaded = load_def('cars.pkl')
for i in loaded:
    print(f"Mark: {i.mark}, Model: {i.model}, Year: {i.year}")
```

Протокол работы программы:

Mark: Toyota, Model: Corolla, Year: 2015
Mark: Honda, Model: Civic, Year: 2018

Mark: Ford, Model: Focus, Year: 2012

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал(а) навыки составления программ с использованием ООП в IDE PyCharm Community. Были использованы языковые конструкции `__dict__`, `class`.

Готовые программные коды выложены на GitHub.