

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.**Постановка цели:**

- 1) Создать класс «Студент», который имеет атрибуты имя, фамилия и оценки. Добавьте методы для вычисления среднего балла и определения, является ли студент отличником.

Текст программы:

```

1 3 usages new *
4 class Student:
5     new *
6     def __init__(self, name, surname, evaluations):
7         self.name = name
8         self.surname = surname
9         self.evaluations = evaluations
10
11 3 usages new *
12
13 3 usages new *
14 def average_evaluations(self):
15     return sum(self.evaluations) / len(self.evaluations)
16
17 3 usages new *
18 def excellent_student(self):
19     return all(evaluations >= 5 for evaluations in self.evaluations)
20
21 st1 = Student( name: "Юлия", surname: "Сухачёва", evaluations: [5, 5, 5, 5, 5])
22 st2 = Student( name: "Данил", surname: "Рычков", evaluations: [5, 5, 4, 5, 4])
23 st3 = Student( name: "Анастасия", surname: "Лещенко", evaluations: [4, 4, 4, 5, 5])
24
25 print(st1.__dict__)
26 print(st2.__dict__)
27 print(st3.__dict__)
28
29 print("Средний балл студента:", st1.average_evaluations())
30 print("Средний балл студента:", st2.average_evaluations())
31 print("Средний балл студента:", st3.average_evaluations())
32
33 print("Студент отличник?", st1.excellent_student())
34 print("Студент отличник?", st2.excellent_student())
  
```

Протокол программы:

```
{'name': 'Юлия', 'surname': 'Сухачёва', 'evaluations': [5, 5, 5, 5, 5]}
```

```
{'name': 'Данил', 'surname': 'Рычков', 'evaluations': [5, 5, 4, 5, 4]}
```

```
{'name': 'Анастасия', 'surname': 'Лещенко', 'evaluations': [4, 4, 4, 5, 5]}
```

Средний балл студента: 5.0

Средний балл студента: 4.6

Средний балл студента: 4.4

Студент отличник? True

Студент отличник? False

Студент отличник? False

Process finished with exit code 0

- 2) Создать 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:

```
pz_16_1.py pz_16_2.py pz_16_3.py STUDENT.bin

2 usages new *
7 class Avto:
    new *
8     def __init__(self, brand, model, year):
9         self.brand = brand
10        self.model = model
11        self.year = year
12
13 usage new *
13 class Truck(Avto):
14     new *
15     def __init__(self, brand, model, year, capacity):
16         super().__init__(brand, model, year)
17         self.capacity = capacity
18
19 usage new *
18 class Passenger(Avto):
19     new *
20     def __init__(self, brand, model, year, passengers):
21         super().__init__(brand, model, year)
22         self.passengers = passengers
23
24 car1 = Truck(brand="Tesla", model="Tesla Semi", year=2017, capacity=37000)
25 car2 = Passenger(brand="BMW Motorsport", model="BMW M5", year=2021, passengers=5)
26
27 print("Грузовик:", car1.brand, car1.model, car1.year, "\nГрузоподъемность:", car1.capacity, "\n")
28 print("Легковой автомобиль:", car2.brand, car2.model, car2.year, "\nКоличество пассажиров:", car2.passengers)
```

Протокол программы:

Грузовик: Tesla Tesla Semi 2017

Грузоподъемность: 37000

Имя: Анастасия, Фамилия: Лещенко, Оценки: [4, 4, 4, 5, 5]

Process finished with exit code 0

Вывод: закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с ООП в IDE PyCharm Community.