

## Практическое занятие № 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 задачи.

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

Создайте класс "Человек", который содержит информацию о имени, возрасте и поле. Создайте классы "Мужчина" и "Женщина", которые наследуются от класса "Человек". Каждый класс должен иметь метод, который выводит информацию о поле объекта.

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:

```
# Создайте класс «Студент», который имеет атрибуты имя, фамилия и оценки.
# Добавьте методы для вычисления среднего балла и определения, является ли
студент
# отличником.

class Student:
    def __init__(self, name, surname, grades):
        self.name = name
        self.surname = surname
        self.grades = grades

    def sr_ball(self):
        sred = sum(self.grades) / len(self.grades)
        print("Средний балл студента равен ", round(sred, 2))
        return sred

    def sucess(self):
        if self.sr_ball() == 5:
            print("Студент отличник")
        else:
            print("Студент не отличник")

student1 = Student("Иван", "Иванов", [5, 5, 5, 5, 5, 5, 5])
print(student1.__dict__)
# student1.sr_ball()
student1.sucess()
```

### Протокол работы программы:

```
{'name': 'Иван', 'surname': 'Иванов', 'grades': [5, 5, 5, 5, 5, 5, 5]}
```

Средний балл студента равен 5.0

Студент отличник

Process finished with exit code 0

### Текст программы:

```
# Создайте класс "Человек", который содержит информацию о имени, возрасте и поле.
# Создайте классы "Мужчина" и "Женщина", которые наследуются от класса
# "Человек". Каждый класс должен иметь метод, который выводит информацию о
# поле объекта.

class Human:
    def __init__(self, name, age, gender):
        self.name = name
        self.age = age
        self.gender = gender

    def show_gender(self):
        print("Пол: ", self.gender)

class Man(Human):
    def __init__(self, name, age):
        super().__init__(name, age, "Мужской")

class Woman(Human):
    def __init__(self, name, age):
        super().__init__(name, age, "Женский")

pers1 = Human("Иван", "18", "Мужской")
pers2 = Man("Алексей", "19")
pers3 = Woman("Евгения", "20")

pers1.show_gender()
pers2.show_gender()
pers3.show_gender()
```

### Протокол работы программы:

Пол: Мужской

Пол: Мужской

Пол: Женский

Process finished with exit code 0

## Текст программы:

```
# 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
import pickle

class Student:
    def __init__(self, name, surname, grades):
        self.name = name
        self.surname = surname
        self.grades = grades

    def sr_ball(self):
        sred = sum(self.grades) / len(self.grades)
        print("Средний балл студента равен ", round(sred, 2))
        return sred

    def sucess(self):
        if self.sr_ball() == 5:
            print("Студент отличник")
        else:
            print("Студент не отличник")

def save_def(*args):
    with open("students.pickle", "wb") as file:
        pickle.dump(args, file)

def load_def():
    with open("students.pickle", "rb") as file:
        data = pickle.load(file)
        return data

student1 = Student("Иван", "Иванов", [5, 5, 5, 5, 5, 5, 5])
student2 = Student("Сергей", "Сергеев", [5, 4, 4, 5, 5, 5, 3])
student3 = Student("Александр", "Александров", [3, 5, 4, 4, 3, 5, 4])

save_def(student1, student2, student3)

students = load_def()
for student in students:
    print("Имя: ", student.name, ", Фамилия: ", student.surname, ", Оценки: ", student.grades)
```

## Протокол работы программы:

Имя: Иван , Фамилия: Иванов , Оценки: [5, 5, 5, 5, 5, 5, 5]

Имя: Сергей , Фамилия: Сергеев , Оценки: [5, 4, 4, 5, 5, 5, 3]

Имя: Александр , Фамилия: Александров , Оценки: [3, 5, 4, 4, 3, 5, 4]

Process finished with exit code 0

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