

Практическое занятие №16

Тема: Составление программ для работы с ООП в IDE PyCharm Community

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

Постановка задачи №1

```
#Блок 1
# Создайте класс "Студент", который имеет атрибуты имя, фамилия, и оценки.
# Добавьте методы для вычисления среднего балла и определения, является ли студент
отличником.
#Блок 3
#Для задачи из блока 1 создать две функции, save_def и load_def, которые позволяет 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 Student:
    def __init__(self, name, surname, grade):
        self.name = name
        self.surname = surname
        self.grade = grade

    def grades(self):
        print(self.grade)
        ocenki = sum(self.grade) / len(self.grade)
        print(ocenki)
        if ocenki == 5.0:
            print("Отличник")
        else:
            print("Не отличник")

    def save_def(self):
        with open('student.txt', 'wb') as f:
            pickle.dump(self, f)
            print("Успех")

    def load_def(self):
        with open('student.txt', 'rb') as f:
            st = pickle.load(f)
            print(st, st.name, st.surname, st.grade)
```

```
student1 = Student('John', 'Smith', [5, 5, 5, 4, 5, 5])
student1.grades()
student1.save_def()
student1.load_def()
```

Протокол работы программы:

```
[5, 5, 5, 4, 5, 5]
4.833333333333333
Не отличник
Успех
<__main__.Student object at 0x7fab237d7e80> John Smith [5, 5, 5, 4, 5, 5]
```

Process finished with exit code 0

Постановка задачи №2

#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.

Текст программы:

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

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

    def mans(self):
        print("Мужчина " + self.name + " " + self.age + " " + self.gender)

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

    def women(self):
        print("Женщина " + self.name + " " + self.age + " " + self.gender)

man1 = Man("Jon", "20", "male")
man1.mans()
woman1 = Woman("Alla", "21", "female")
woman1.women()
```

Протокол работы программы:

Мужчина Jon 20 male

Женщина Alla 21 female

Process finished with exit code 0

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

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

Готовые программные коды выложены на GitHub.