

Практическое занятие №16-(1, 2, 3)

Тема: Составление программ с использованием ООП в IDE PyCharm.

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

Постановка задачи:

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

16.2 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.

16.3 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:

```
# № 16.1
name = input("Введите имя: ")
surname = input("Введите фамилию: ")
a = []

for i in range(5):
    grades = int(input("Введите оценку: "))
    a.append(grades)

1 usage
class Student:
    def __init__(self, first_name, last_name, grades):
        self.first_name = first_name
        self.last_name = last_name
        self.grades = grades

    def is_honor(self):
        average_grade = sum(self.grades) / len(self.grades)
        return average_grade == 5

    def __repr__(self):
        average_grade = sum(self.grades) / len(self.grades)
        return (f"Студент: {self.first_name} {self.last_name}"
                f"Оценки: {self.grades}"
                f"Средний балл: {average_grade}"
                f"Отличник: {'Да' if average_grade == 5 else 'Нет'}")

print(Student(name, surname, a))
```

```
# № 16.2
2 usages
class Person:
    def __init__(self, name, age, gender):
        self.name = name
        self.age = age
        self.gender = gender

1 usage
class Man(Person):
    def __init__(self, name, age):
        super().__init__(name, age, gender: "Мужчина")

1 usage
class Woman(Person):
    def __init__(self, name, age):
        super().__init__(name, age, gender: "Женщина")

1 usage
def input_person_data():
    name = input("Введите имя: ")
    age = int(input("Введите возраст: "))
    gender = input('Введите пол ("м" или "ж"): ').lower()
    return name, age, gender

1 usage
def create_person(name, age, gender):
    if gender == "м":
        return Man(name, age)
    elif gender == "ж":
        return Woman(name, age)
    else:
        raise ValueError("Неправильно указан пол")

name, age, gender = input_person_data()
person = create_person(name, age, gender)

print(f"Данные: ({person.name}, {person.age}, {person.gender})")
```

```
# № 16.3
import pickle

3 usages
class Student:
    def __init__(self, first_name, last_name, grades):
        self.first_name = first_name
        self.last_name = last_name
        self.grades = grades

    def is_honor(self):
        average_grade = sum(self.grades) / len(self.grades)
        return average_grade == 5

    def __repr__(self):
        average_grade = sum(self.grades) / len(self.grades)
        return (f"Студент: {self.first_name} {self.last_name}"
                f"Оценки: {self.grades}"
                f"Средний балл: {average_grade}"
                f"Отличник: {'Да' if average_grade == 5 else 'Нет'}")

student_1 = Student(first_name="Ирина", last_name="Лебедева", grades=[5, 5, 5, 5, 5])
student_2 = Student(first_name="Иван", last_name="Петров", grades=[5, 4, 5, 4, 4])
student_3 = Student(first_name="Антон", last_name="Сидоров", grades=[3, 4, 3, 5, 4])

with open("student.pkl", "wb") as file:
    pickle.dump(student_1, file)
    pickle.dump(student_2, file)
    pickle.dump(student_3, file)

with open("student.pkl", "rb") as file:
    student_1 = pickle.load(file)
    student_2 = pickle.load(file)
    student_3 = pickle.load(file)

print(student_1.__dict__)
print(student_2.__dict__)
print(student_3.__dict__)
```

Протокол работы программы:

16.1

Введите имя: Саша

Введите фамилию: Петров

Введите оценку: 5

Введите оценку: 5

Введите оценку: 5

Введите оценку: 5

Введите оценку: 5

Студент: Саша Петров

Оценки: [5, 5, 5, 5, 5]

Средний балл: 5.0

Отличник: Да

16.2

Введите имя: Женя

Введите возраст: 19

Введите пол ("м" или "ж"): ж

Данные: (Женя, 19, Женщина)

16.3

```
{'first_name': 'Ирина', 'last_name': 'Лебедева', 'grades': [5, 5, 5, 5, 5]}
```

```
{'first_name': 'Иван', 'last_name': 'Петров', 'grades': [5, 4, 5, 4, 4]}
```

```
{'first_name': 'Антон', 'last_name': 'Сидоров', 'grades': [3, 4, 3, 5, 4]}
```

Вывод: В процессе выполнения практического задания выработал навыки составления программ с ООП в IDE PyCharm. Выполнены разработка кода, отладка, тестирование, оптимизация. Готовые программные коды выложены на GitHub.