

Студент группы ИС-36 Жирков В.

Практическое занятие №16

Тема: Составление программ с использованием ООП в IDE PyCharm Community.

Цель: Закрепить знания, понятия, алгоритмы и основные принципы составления программ с использованием объектно-ориентированного программирования.

Постановка задачи

****Выполнить три задачи (по одной из каждого блока):****

****Блок 1 (выбрана задача 2): Создать класс «Студент»****

Класс должен иметь атрибуты: имя, фамилия и оценки. Реализовать методы:

- Вычисление среднего балла
- Определение, является ли студент отличником (средний балл ≥ 4.5)

****Блок 2 (выбрана задача 1): Создать классы «Фигура», «Квадрат», «Прямоугольник»****

- Базовый класс `Фигура` содержит метод `площадь()`.
- Классы `Квадрат` и `Прямоугольник` наследуются от `Фигура` и реализуют собственный метод вычисления площади.

****Блок 3 (сериализация объектов):****

Для класса «Студент» из Блока 1 создать две функции:

- `save_def()` – сохраняет три экземпляра класса в файл (бинарный формат)
- `load_def()` – загружает объекты из файла

Использовать модуль `pickle`.

Текст программы

```
```python
import pickle
import os

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

```
self.last_name = last_name
self.grades = grades

def average_grade(self):
 if not self.grades:
 return 0.0
 return sum(self.grades) / len(self.grades)

def is_excellent(self):
 return self.average_grade() >= 4.5

def __str__(self):
 avg = self.average_grade()
 status = "отличник" if self.is_excellent() else "не отличник"
 return f"{self.last_name} {self.first_name}: средний балл {avg:.2f}"
 ({status})"

class Figure:
 def area(self):
 raise NotImplementedError("Метод area() должен быть реализован в подклассе")

class Square(Figure):
 def __init__(self, side):
 self.side = side

 def area(self):
 return self.side ** 2

 def __str__(self):
 return f"Квадрат со стороной {self.side}: площадь = {self.area():.2f}"

class Rectangle(Figure):
 def __init__(self, width, height):
 self.width = width
 self.height = height

 def area(self):
 return self.width * self.height
```

```
def __str__(self):
 return f"Прямоугольник {self.width}x{self.height}: площадь = {self.area():.2f}"

def save_def(students, filename="students.pkl"):
 try:
 with open(filename, 'wb') as file:
 pickle.dump(students, file)
 print(f"Объекты успешно сохранены в файл '{filename}'")
 except Exception as e:
 print(f"Ошибка при сохранении: {e}")

def load_def(filename="students.pkl"):
 if not os.path.exists(filename):
 print(f"Файл '{filename}' не найден")
 return []
 try:
 with open(filename, 'rb') as file:
 students = pickle.load(file)
 print(f"Объекты успешно загружены из файла '{filename}'")
 return students
 except Exception as e:
 print(f"Ошибка при загрузке: {e}")
 return []

if __name__ == "__main__":
 print("=" * 70)
 print("ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №16: ООП В PYTHON")
 print("=" * 70)

 print("\n1. ТЕСТ КЛАССА 'СТУДЕНТ':")
 student1 = Student("Иван", "Петров", [5, 4, 5, 5, 4])
 student2 = Student("Мария", "Сидорова", [3, 4, 3, 4, 4])
 student3 = Student("Алексей", "Иванов", [5, 5, 5, 5, 5])

 for student in [student1, student2, student3]:
 print(f" - {student}")
```

```
print("\n2. ТЕСТ КЛАССОВ ФИГУР:")
square = Square(5.0)
rectangle = Rectangle(4.0, 6.0)

print(f" - {square}")
print(f" - {rectangle}")

print("\n3. ТЕСТ СЕРИАЛИЗАЦИИ:")

students_to_save = [student1, student2, student3]
save_def(students_to_save)

loaded_students = load_def()

print("\n Загруженные студенты:")
if loaded_students:
 for i, student in enumerate(loaded_students, 1):
 print(f" {i}. {student}")

print("\n" + "=" * 70)
print("РАБОТА ЗАВЕРШЕНА")
print("=" * 70)
```

```

■ Протокол работы программы

```

=====

#### 1. ТЕСТ КЛАССА 'СТУДЕНТ':

- Петров Иван: средний балл 4.60 (отличник)
- Сидорова Мария: средний балл 3.60 (не отличник)
- Иванов Алексей: средний балл 5.00 (отличник)

#### 2. ТЕСТ КЛАССОВ ФИГУР:

- Квадрат со стороной 5.0: площадь = 25.00
- Прямоугольник 4.0x6.0: площадь = 24.00

#### 3. ТЕСТ СЕРИАЛИЗАЦИИ:

Объекты успешно сохранены в файл 'students.pkl'

Объекты успешно загружены из файл 'students.pkl'

**Загруженные студенты:**

1. Петров Иван: средний балл 4.60 (отличник)
2. Сидорова Мария: средний балл 3.60 (не отличник)
3. Иванов Алексей: средний балл 5.00 (отличник)

---

**РАБОТА ЗАВЕРШЕНА**

---

...

**■ Вывод**

**В ходе выполнения работы были успешно реализованы три задачи по объектно-ориентированному программированию:**

1. Блок 1: Создан класс `Student` с методами вычисления среднего балла и проверки статуса отличника.
2. Блок 2: Построена иерархия классов `Figure` → `Square`, `Rectangle` с применением наследования и полиморфизма.
3. Блок 3: Реализованы функции `save\_def()` и `load\_def()` для сериализации и десериализации объектов класса `Student` с использованием модуля `pickle`.

**Код соответствует стандарту PEP 8 и демонстрирует основные принципы ООП.**