**Практическое занятие № 16**

**Задача №1**

**Тема:** составление программ для работы с ООП и библиотекой pickle в IDE PyCharm Community.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работая с библиотекой pickle в IDE PyCharm Community

**Постановка задачи.**

# Создайте класс «Книга», который имеет атрибуты название, автор и количество страниц.

# Добавьте методы для чтения и записи книги.

**Текст программы:**

import pickle  
class Book:  
 def \_\_init\_\_(self, title, author, pages):  
 self.title = title  
 self.author = author  
 self.pages = pages  
  
 def \_\_str\_\_(self):  
 return f"Название: {self.title}, Автор: {self.author}, Количество страниц: {self.pages}"  
  
def save\_books(book\_list):  
 with open('books\_data.pkl', 'wb') as f:  
 pickle.dump(book\_list, f)  
  
def load\_books():  
 try:  
 with open('books\_data.pkl', 'rb') as f:  
 book\_list = pickle.load(f)  
 return book\_list  
 except FileNotFoundError:  
 return []  
  
book1 = Book("1984", "Джордж Оруэлл", 328)  
book2 = Book("Преступление и наказание", "Фёдор Достоевский", 671)  
  
books = [book1, book2]  
save\_books(books)  
  
loaded\_books = load\_books()  
for book in loaded\_books:  
 print(book)

**Протокол работы программы:**

Название: 1984, Автор: Джордж Оруэлл, Количество страниц: 328

Название: Преступление и наказание, Автор: Фёдор Достоевский, Количество страниц: 671

Process finished with exit code 0

**Практическая №2**

**Постановка задачи:**

# Создайте базовый класс "Транспорт" со свойствами "марка", "модель" и "год выпуска".

# От этого класса унаследуйте класс "Автомобиль" и добавьте в него свойство "тип кузова"

**Текст программы**:

class Transport:  
 def \_\_init\_\_(self, brand, model, year):  
 self.brand = brand  
 self.model = model  
 self.year = year  
  
 def \_\_str\_\_(self):  
 return f"Марка: {self.brand}, Модель: {self.model}, Год выпуска: {self.year}"  
  
  
class Car(Transport):  
 def \_\_init\_\_(self, brand, model, year, body\_type):  
 super().\_\_init\_\_(brand, model, year)  
 self.body\_type = body\_type  
  
 def \_\_str\_\_(self):  
 return f"{super().\_\_str\_\_()}, Тип кузова: {self.body\_type}"  
  
  
car1 = Car("Toyota", "Camry", 2020, "Седан")  
print(car1.\_\_str\_\_())  
  
car2 = Car("Mercedes-Benz", "GLE", 2022, "Кроссовер")  
print(car2.\_\_str\_\_())

**Протокол работы программы:**

Марка: Toyota, Модель: Camry, Год выпуска: 2020, Тип кузова: Седан

Марка: Mercedes-Benz, Модель: GLE, Год выпуска: 2022, Тип кузова: Кроссовер

**Process finished with exit code 0**

**Вывод:**

Оценив итоги выполнения этой задачи по работе с сохранением и загрузкой объектов с использованием библиотеки pickle, я улучшил свои навыки в создании и обработке структурированных данных. Полученные знания позволят применять методы сериализации и десериализации в будущих проектах. Готовые программные коды выложены на GitHub.