

Чтение и запись в бинарном режиме доступа

Модуль Pickle

Это собственный формат сериализации объекта Python. Интерфейс pickle обеспечивает четыре метода: `dump`, `dumps`, `load`, и `loads`. Метод `dump()` сериализует в открытый файл (файл-подобный объект). Метод `dumps()` сериализует в строку. Метод `load()` десериализует из открытого файлового объекта. Метод `loads()` десериализует из строки.

Pickle поддерживает по умолчанию текстовый протокол, но имеет также двоичный протокол, который более эффективен, но не читается человеком (полезно при отладке).

Бинарный режим доступа - режим доступа, когда данные из файла считываются один в один без какой-либо обработки. Это можно использовать, в первую очередь, для сохранения и считывания объектов. Предположим, что нужно сохранить в файл данный список (рис. 1):

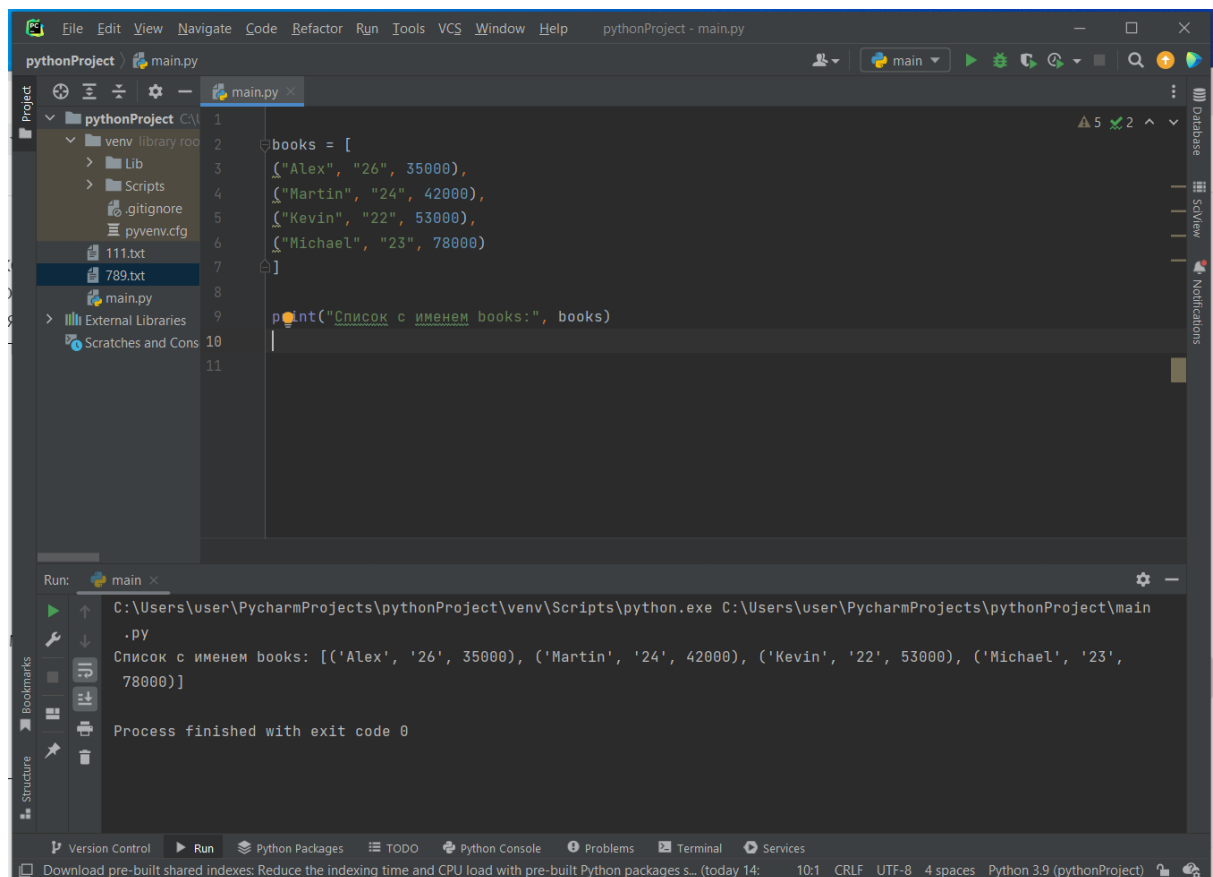


Рисунок 1 - Синтаксис содержимого списка с именем *books*

Затем нам нужно открыть файл на запись в бинарном режиме и подключить специальный встроенный модуль **pickle**. Для этого модуля также вызовем метод `dump()`. После этого произошло сохранение объекта в файл.

Аналогичным образом можно записывать и считывать сразу несколько объектов, например, следующим образом (рис. 2):

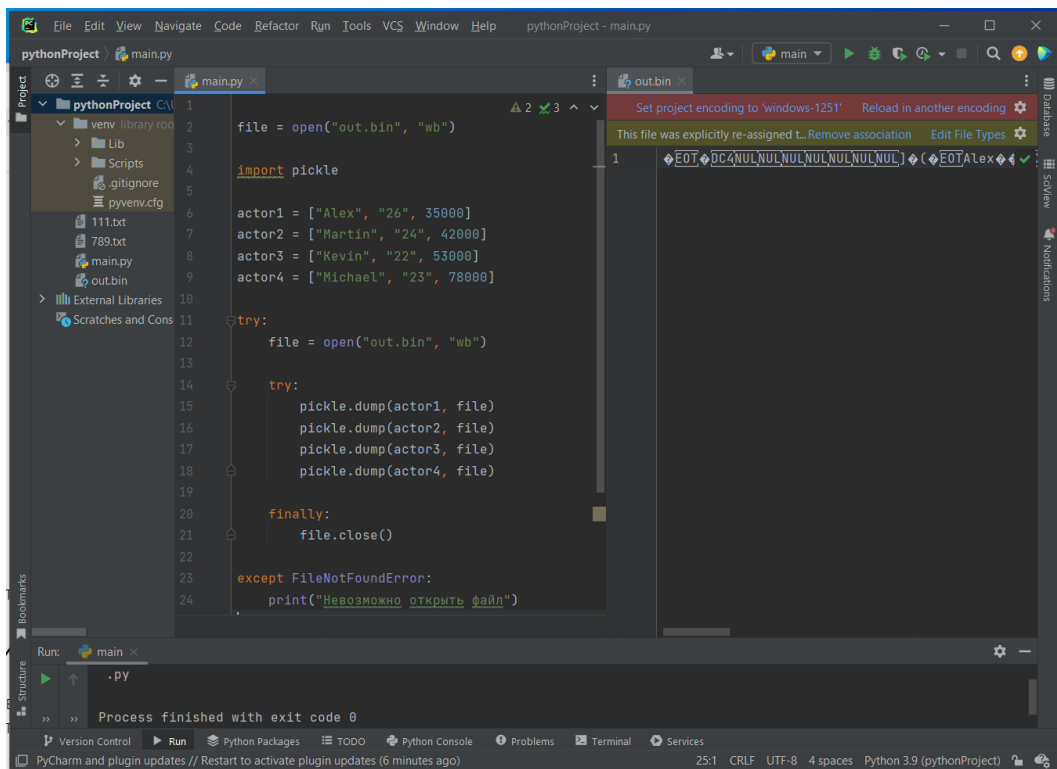


Рисунок 2 - Синтаксис операций записи и считывания нескольких объектов

Затем нужно просто считать данные в том же порядке (рис. 3):

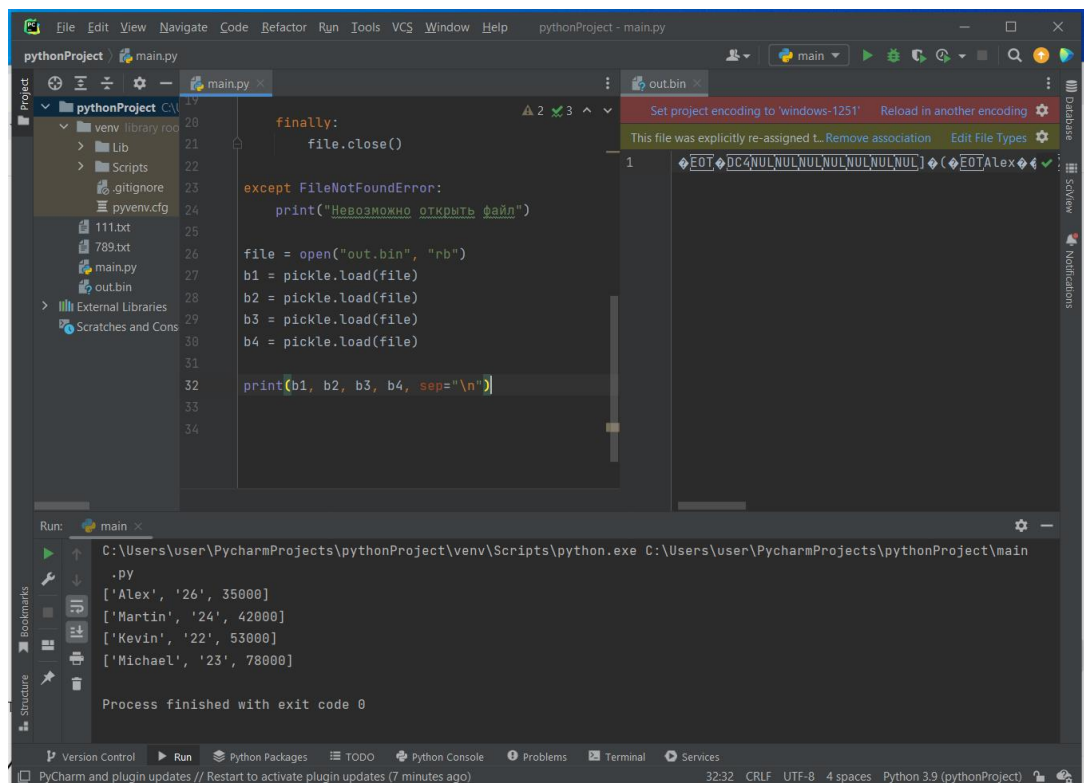


Рисунок 3 - Листинг кода для считывания информации в заданном порядке

Весь код выглядит следующим образом (рис. 4):

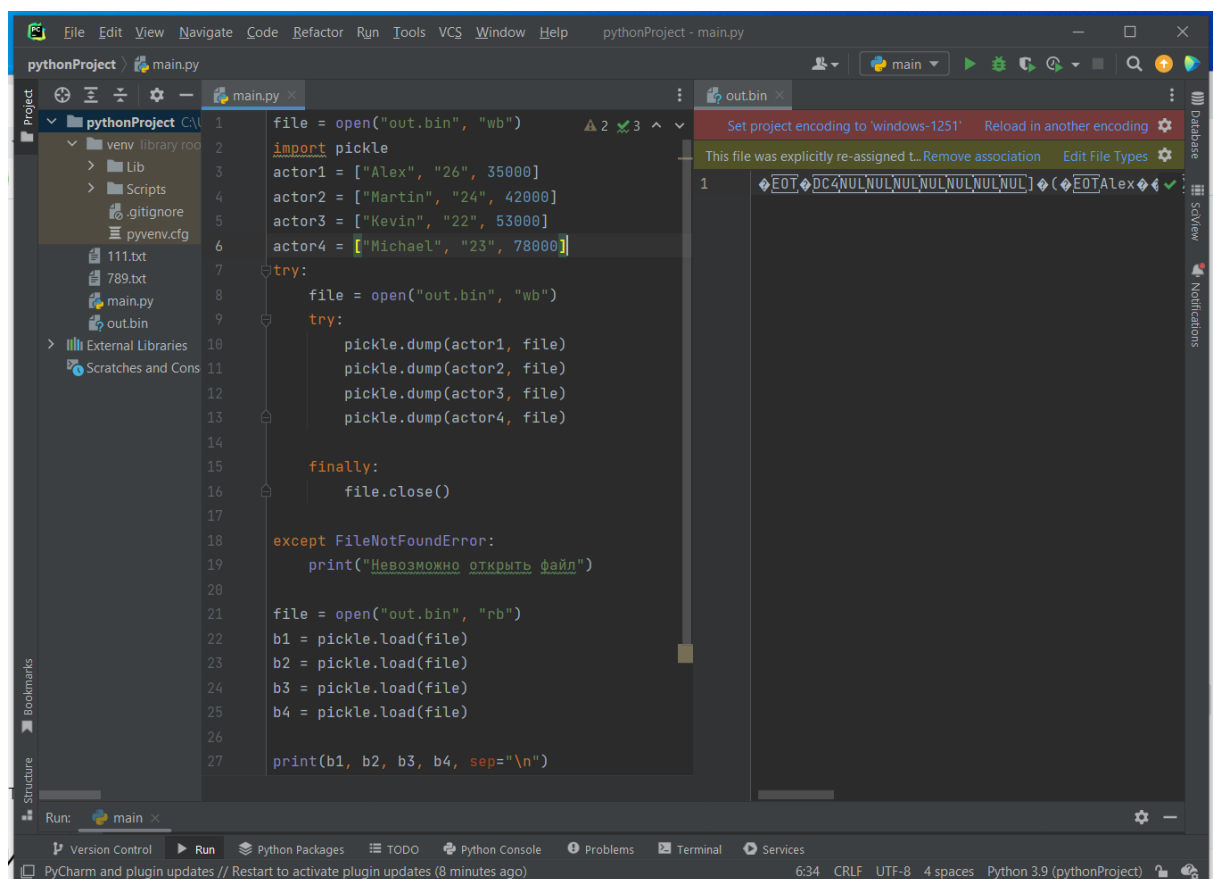


Рисунок 4 - Листинг кода для операций записи и считывания информации

Пример 2

```

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

    def get_values(self):
        print(f"{self.age = }")
        print(f"{self.name = }")

import pickle

with open("User.bin", "wb") as file:
    pickle.dump(User, file)

with open("User.bin", "rb") as file:
    user = pickle.load(file)

# print(file)

a = User(10, "Max")
a.get_values()

```