ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель		В. А. Миклуш
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 7

СТРУКТУРЫ ДАННЫХ – СЛОВАРИ, МНОЖЕСТВА. JSON

по курсу:

ИНФОРМАТИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 4326		Г. С. Томчук
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2023

Оглавление

Цель работы

Целью лабораторной работы является изучение словарей, множеств и JSON. Задача: реализовать приложение, используя JSON для хранения актуальных данных.

Вопросы, изучаемые в работе:

- Знакомство со словарями и множествами в Python
- Методы и операции со словарями и множествами
- Использование словарей и множеств для хранения и поиска данных
- Использование JSON для хранения данных.

Задание

Вариант 20. Разработать систему, реализующую CRUD операции с базой данных книжного магазина. При этом:

- Запрещено использовать внешние библиотеки.

Листинг программы

Были написаны следующие модули, листинги которых представлены ниже:

```
# json_service.py

import json

def get_database():
  with open("db.json", encoding="utf-8") as db_file:
    return json.load(db_file)

def set_database(db):
  with open("db.json", "w", encoding="utf-8") as db_file:
    json.dump(db, db_file, indent=2, ensure_ascii=False)
```

```
# collections.py

from db import json_service

db = json_service.get_database()

def get_name():
  return db["name"]

def get_address():
  return db["address"]

def get_one_by_id(collection, id):
  for item in db[collection]:
    if item["id"] == id:
      return item

  return {"message": f"Элемент {id=} не найден в {collection}!"}

def get_all(collection):
  return db[collection]

def create_one(collection, new_item):
  last_item_id = get_all(collection)[-1]["id"] if
get_all(collection) else 0
  item = {"id": last_item_id + 1, **new_item}
  db[collection].append(item)
  json_service.set_database(db)
  return item

def update_one_by_id(collection, id, new_item):
  for item in db[collection]:
    if item["id"] == id:
      for key in item:
        item[key] = new_item[key] if key in new_item else item[key]
      json_service.set_database(db)
      return item

  return {"message": f"Элемент {id=} не найден в {collection}!"}

def delete_one_by_id(collection, id):
  for i, item in enumerate(db[collection]):
    if item["id"] == id:
      deleted_item = db[collection].pop(i)
      json_service.set_database(db)
      return deleted_item

  return {"message": f"Элемент {id=} не найден в {collection}!"}
```

Результат выполнения программы

На рисунке 1 представлен результат работы программы.

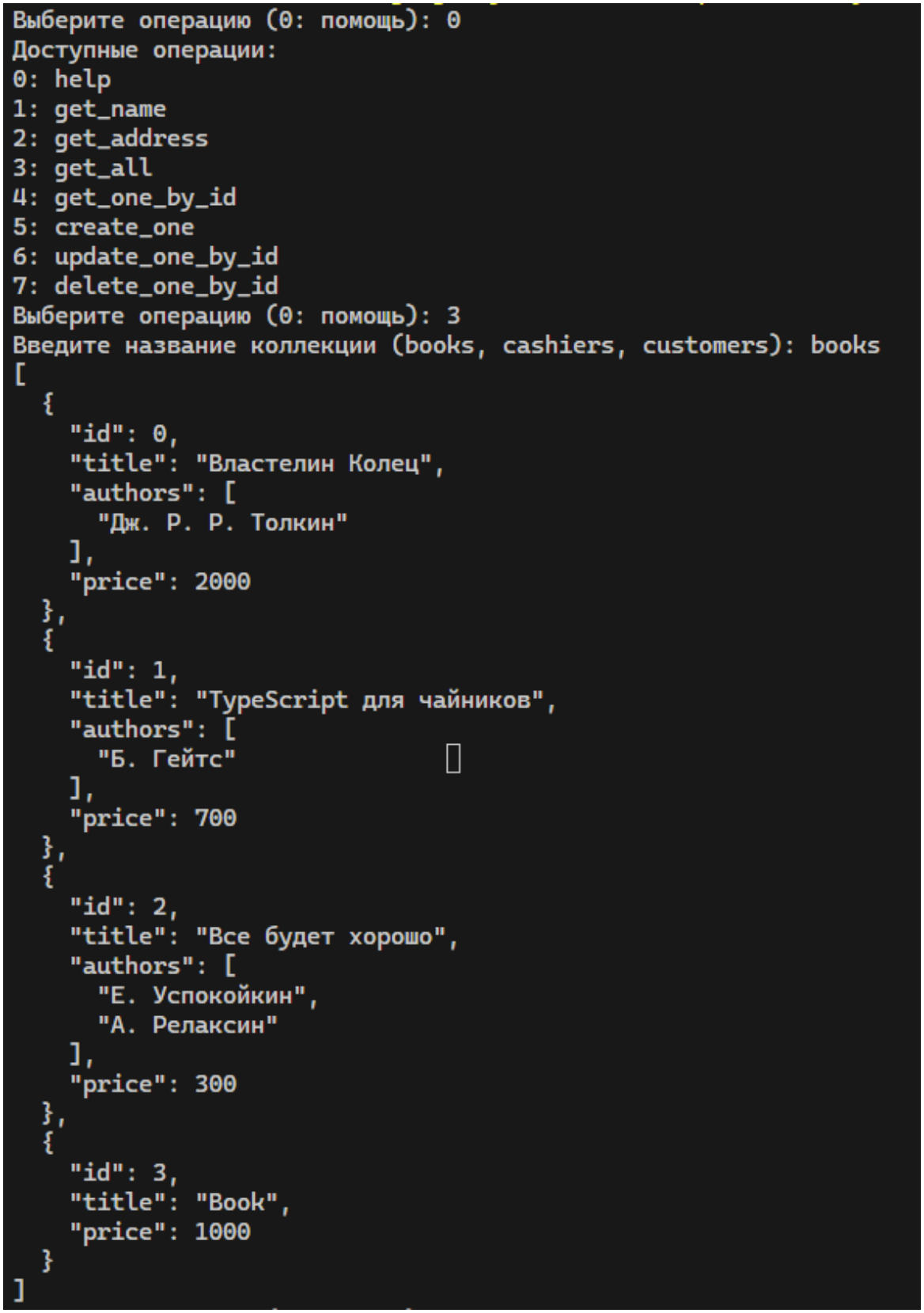

Рисунок 1 – Результат работы программы

Выводы

В ходе выполнения лабораторной работы мной были изучены словари, множества и JSON. При написании программы был освоен навык реализации приложения, используя JSON для хранения актуальных данных. Написанная программа была протестирована, полученный результат соответствует ожидаемому.