|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный  исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №7  по дисциплине «Языки программирования»  Вариант 5 | | |
|  | | Работу выполнил  студент группы БАС-1,2023 2 курса  Иванов С.A  «13» Апреля 2025 г. |
| Работу проверил  Ракина В.Д  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2025 г. |
| Пермь 2025 | | |

СОДЕРЖАНИЕ

[Постановка задачи 4](#_Toc195863812)

[Алгоритм решения 4](#_Toc195863813)

[Тестирование 4](#_Toc195863814)

[Код программы 5](#_Toc195863815)

[Постановка задачи 6](#_Toc195863816)

[Алгоритм решения 6](#_Toc195863817)

[Тестирование 6](#_Toc195863818)

[Код программы 7](#_Toc195863819)

[Постановка задачи 8](#_Toc195863820)

[Алгоритм решения 8](#_Toc195863821)

[Тестирование 8](#_Toc195863822)

[Код программы 9](#_Toc195863823)

[Постановка задачи 10](#_Toc195863824)

[Алгоритм решения 10](#_Toc195863825)

[Тестирование 10](#_Toc195863826)

[Код программы 11](#_Toc195863827)

[Постановка задачи 12](#_Toc195863828)

[Алгоритм решения 12](#_Toc195863829)

[Тестирование 12](#_Toc195863830)

[Код программы 12](#_Toc195863831)

[Постановка задачи 13](#_Toc195863832)

[Алгоритм решения 13](#_Toc195863833)

[Тестирование 13](#_Toc195863834)

[Код программы 14](#_Toc195863835)

[Постановка задачи 15](#_Toc195863836)

[Алгоритм решения 15](#_Toc195863837)

[Тестирование 15](#_Toc195863838)

[Код программы 16](#_Toc195863839)

[Постановка задачи 17](#_Toc195863840)

[Алгоритм решения 17](#_Toc195863841)

[Тестирование 18](#_Toc195863842)

[Код программы 18](#_Toc195863843)

[Постановка задачи 19](#_Toc195863844)

[Алгоритм решения 19](#_Toc195863845)

[Тестирование 19](#_Toc195863846)

[Код программы 20](#_Toc195863847)

[Постановка задачи 21](#_Toc195863848)

[Алгоритм решения 21](#_Toc195863849)

[Тестирование 22](#_Toc195863850)

[Код программы 22](#_Toc195863851)

# Постановка задачи

Разработать консольное приложение с дружественным интерфейсом с возможностью выбора заданий для работы с «базой данных (БД)», хранящейся в бинарном файле. Перечень полей (минимум 5), достаточно полно характеризующих заданную в варианте предметную область, предложить самостоятельно (постараться отразить в перечне полей такие, которые требуют разных типов данных). Приложение должно выполнять следующие функции:

1. Чтение базы данных из бинарного файла.

2. Просмотр базы данных.

3. Удаление элементов (по ключу).

4. Добавление элементов.

5. Реализация 4 запросов (формулировки запросов придумать самостоятельно). 2 запроса должны возвращать перечень, 2 запроса одно значение.

В классе должны присутствовать свойства, конструкторы, перегруженный метод ToString(). Весь функционал приложения реализовать в виде методов вспомогательного класса с помощью LINQ-запросов. Предусмотреть обработку возможных ошибок при работе программы.

Каталог книг

# Алгоритм решения

Описана структура данных BookCatalog, которая будет записываться в бинарный файл. Содержит поля: Id (уникальный идентификатор), Title (название книги), Author (автор книги), Year (год издания), Price (цена). Реализованы свойства для доступа к полям и конструктор для инициализации всех полей. После создан класс BooksOperation:

1. Метод Generate – генерирует рандомными значениями поля BookCatalog, для последующей сериализации в бинарный файл. Создает объект Random для генерации случайных значений, инициализирует массивы с тестовыми названиями книг и именами авторов, в цикле: записывает рандомно структуры в список. Возвращает список.
2. Метод AddNewBookInteractive – добавляет новую книгу в список и после этот список сериализуется в бинарный файл. Принимает параметры новой книги: title, author, year, price; Создает новую книгу с автоматическим назначением Id (текущий размер списка + 1); Добавляет книгу в список books.
3. Метод removeFromID – удаляет книгу из списка и после сериализует новый список без заданной книги по ключу в бинарный файл. Удаляет книгу с указанным ID из списка; В цикле пересчитывает ID оставшихся книг, чтобы сохранить последовательность.
4. Метод Serialize() – метод, который сериализует список с книгами в бинарный файл; Создает XmlSerializer для типа List<BookCatalog>; Открывает файловый поток для записи; Выполняет сериализацию списка books в файл;.
5. Метод Deserialize() – метод, который десериализует файл в список, аналогично с сериализацией.
6. LINQ-запросы: GetBooksByAuthor(string author) - Использует Where для фильтрации книг по автору (без учета регистра), преобразует результат в List с помощью ToList(), возвращает отфильтрованный список; GetBooksPublishedAfterYear(int year): - Использует Where для фильтрации книг по году издания, сортирует результат по году с помощью OrderBy, возвращает отфильтрованный и отсортированный список; GetAverageBookPrice() - Использует Average для вычисления среднего значения цен всех книг, Возвращает полученное значение типа double; GetMostExpensiveBookByAuthor(string author) - Фильтрует книги по автору (Where), сортирует по убыванию цены (OrderByDescending), возвращает первую книгу или null (FirstOrDefault);

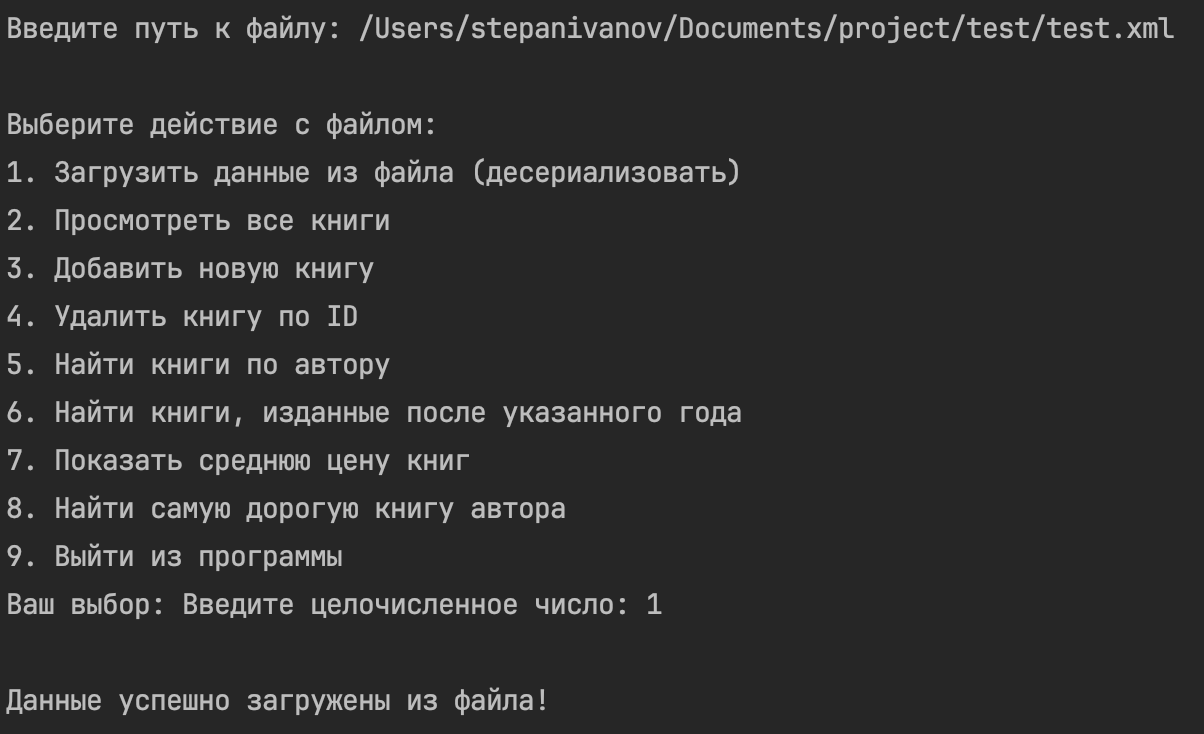
В main создается чекер для проверки ввода и запрашивается путь к файлу, инициализируются объект BooksOperation с параметрами ссылка на файл и список <BookCatalog>. В бесконечном цикле:

1. Выводит меню операций

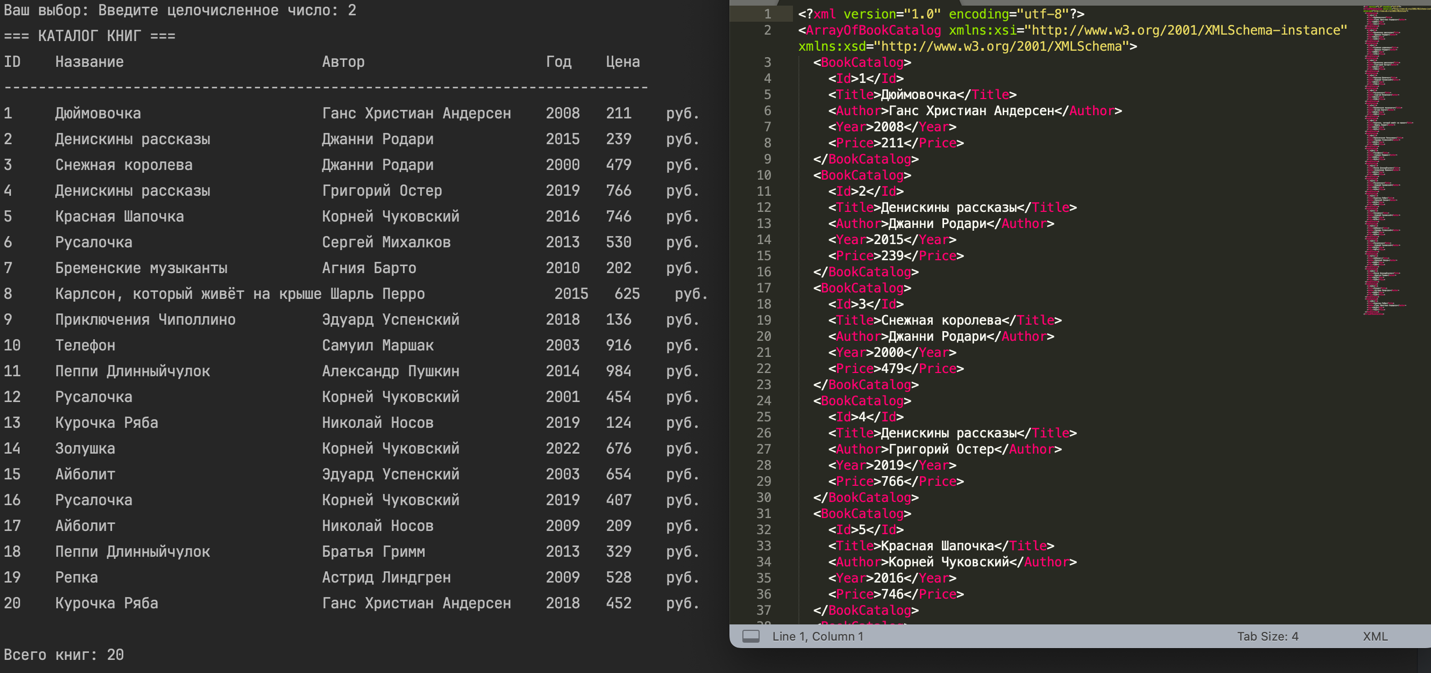
2. Обрабатывает выбор пользователя

3. Для каждой операции вызывает соответствующий метод BooksOperation

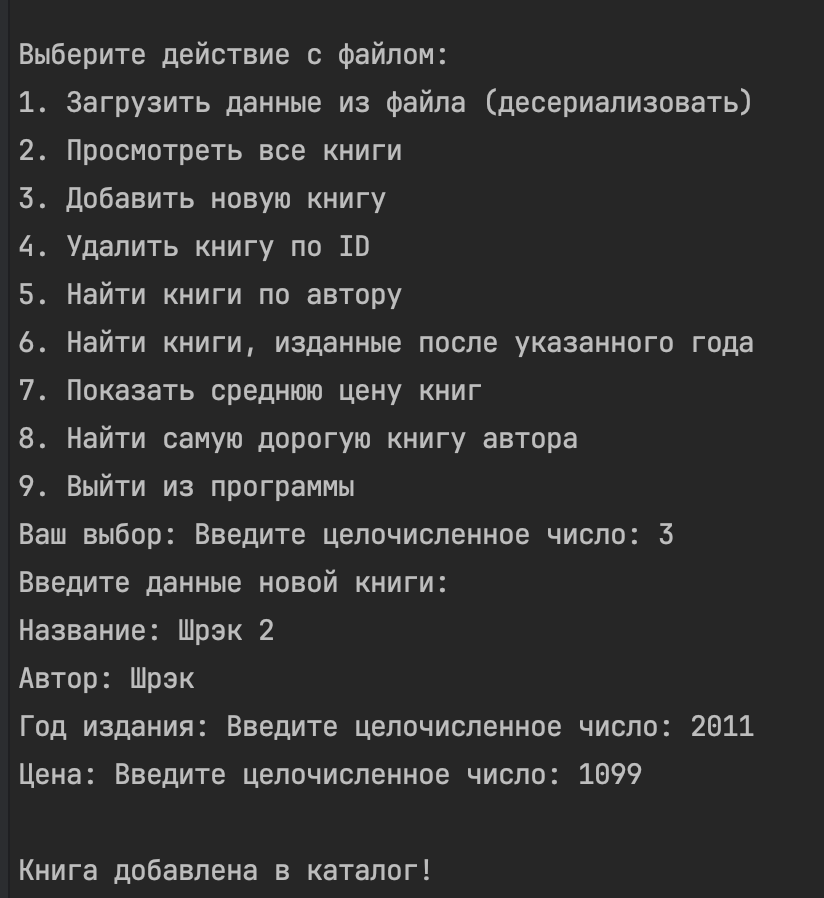
# Тестирование

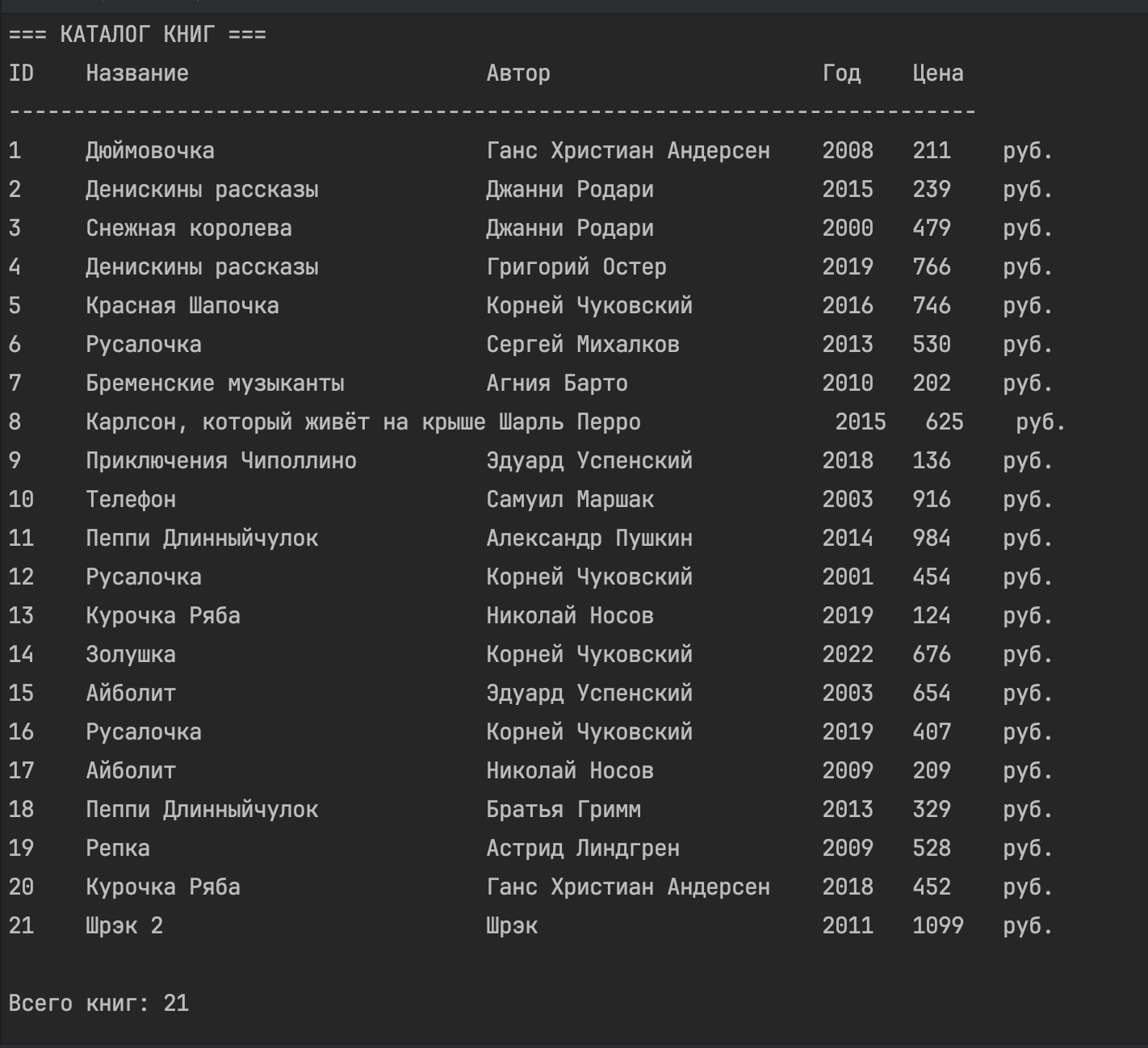


(Десериализация)

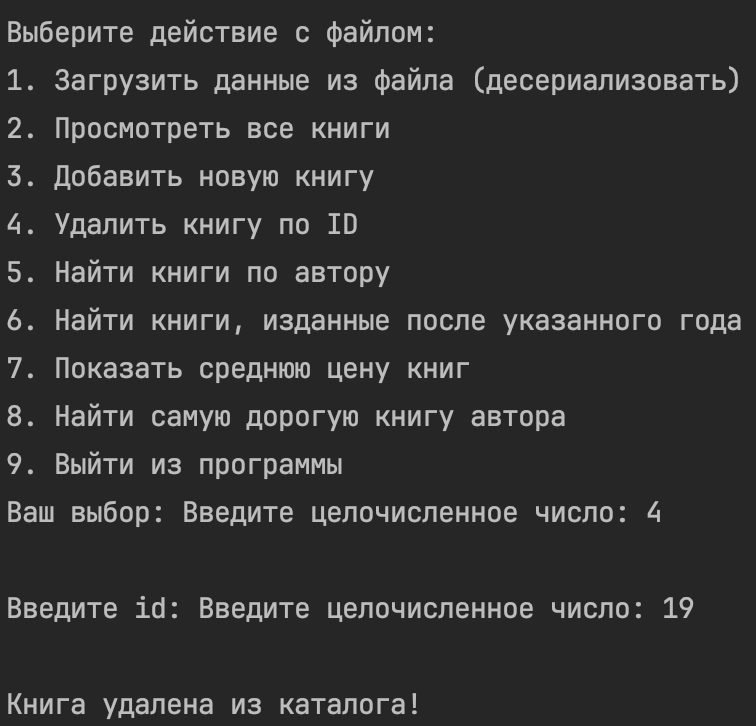


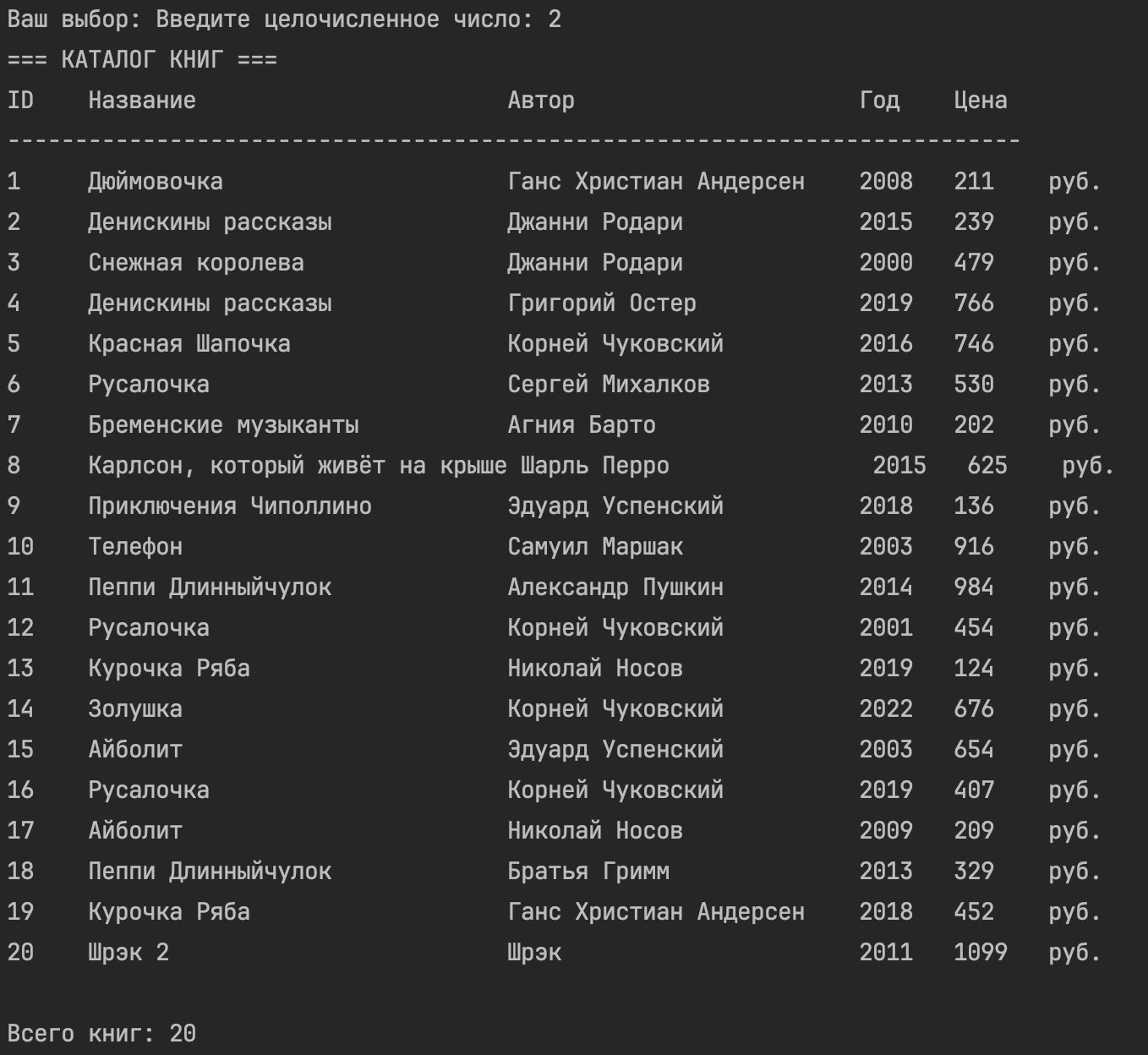
(Просмотр книг)



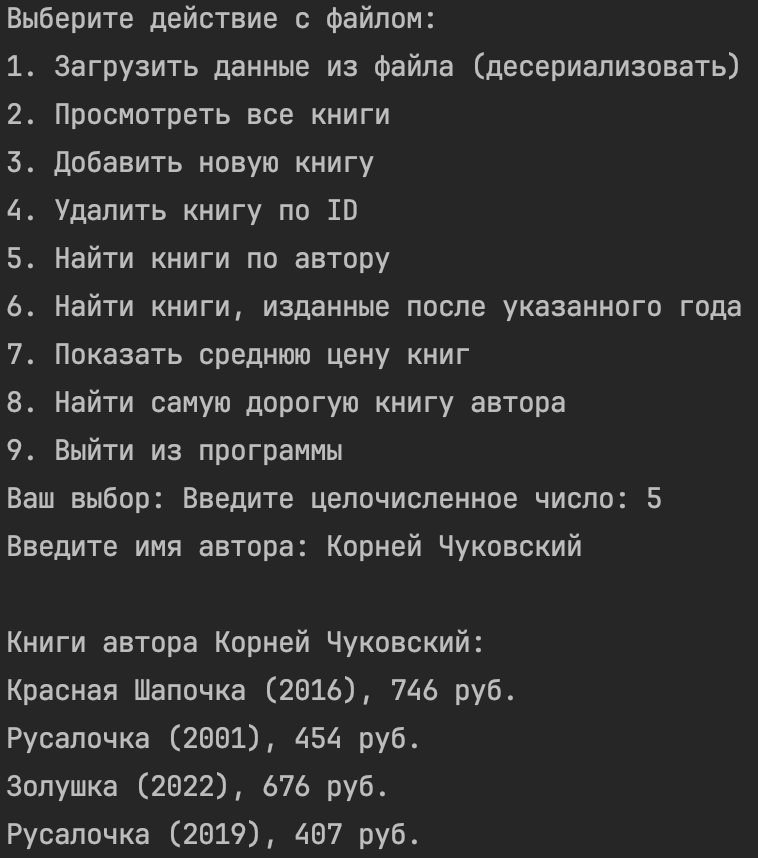


(Добавление кастомной книги, добавил шрэка 2)

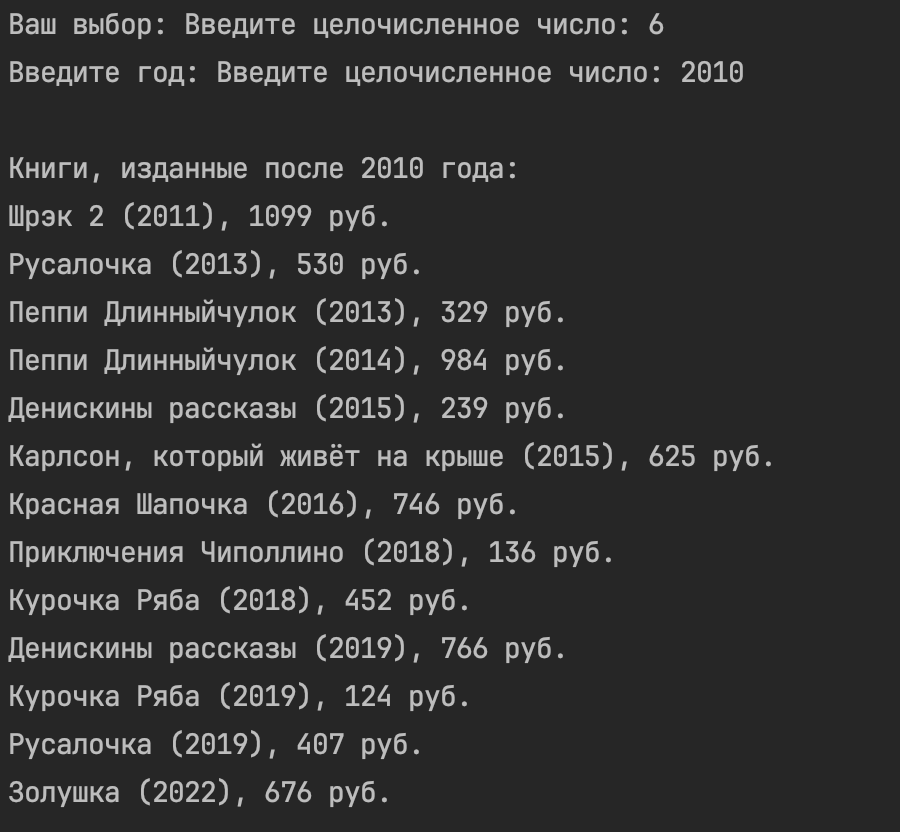




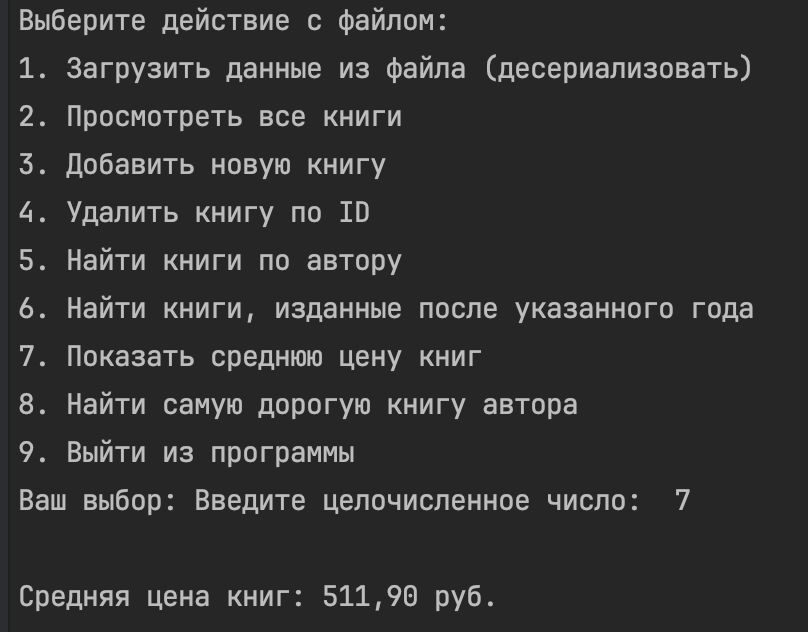
(Удалил книгу с id 19 репку)

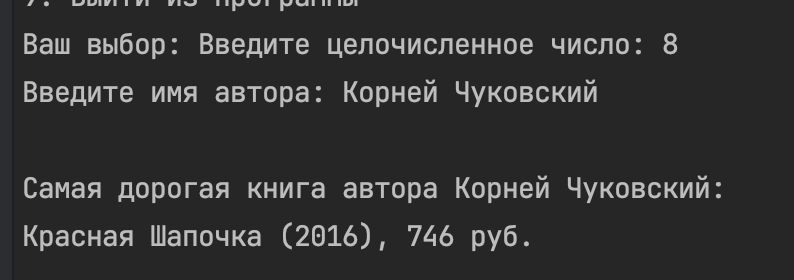


(Запрос на книги по автору)



(Запрос на книги, который вышли изданный после N года)

  
(Запрос на среднюю цену книг)



(Запрос на самую дорогую книгу автора)

# Код программы

https://github.com/6ermutka/yplab/tree/master/C%232