Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Основы конструирования программ

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

на тему

**РАЗРАБОТКА ПРОГРАММЫ УЧЕТА КНИГ В БИБЛИОТЕКЕ**

Выполнил: студент группы 894351

Галкин Илья Викторович.

Проверила:

Мельникова Елена Александровна

Минск 2020

Учреждение образования

«Белорусский государственный университет информатики   
и радиоэлектроники»

Факультет компьютерного проектирования

УТВЕРЖДАЮ

Заведующий кафедрой

*––––––––––––––––––––––––*

(подпись)

–––––––––––––––––––––2020 г.

ЗАДАНИЕ

по курсовой работе

Студенту    *Галкину Илье Викторовичу*

1. Тема работы    *Разработка программы учета книг в библиотеке*

2. Срок сдачи студентом законченной работы––*25.05.2020 г.* –––*–––––––––––––*

3. Исходные данные к работе

*Программа реализует авторизацию для входа в систему, функционал администратора (управление учетными записями пользователей, редактирование и обработка данных) и функционал пользователя (просмотр и обработка данных)*

*Язык программирования C#, среда разработки Microsoft Visual Studio*

*Вид приложения – консольное, парадигма программирования – объектно-ориентированная*

*Способ организации данных – классы, способ хранения данных – база данных и файлы*

*Каждая логически завершенная задача программы должна быть реализована в виде метода*

*Оформление кода должно соответствовать «С# Code Convention»» –––––––Текст пояснительной записки оформляется по стандарту СТП 01–2017–––––*

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)

*Титульный лист––––––––––––––––––––––––––––––––––––           ––––––––––––*

*Задание по курсовой работе––––––––––––––––––––––––––           ––––––––––––*

*Содержание ––––––––––––––––\_\_\_\_\_\_\_\_\_\_\_\_\_––––––––––           ––––––––––––*

*1 Требования к программе–––––––––––––––––––––––––––––––––––           –––––*

*2 Конструирование программы–––––––––––––––––––––––––––––––––––––––––*

*2.1 Разработка модульной структуры программы –––––––––––––––––––––––*

*2.2 Выбор способа организации данных–––––––––––––––––––––––––––––––––*

*2.3 Разработка перечня пользовательских функций программы \_\_   ––––––––*

*3 Разработка алгоритмов работы программы–\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_–––*

*3.2 Алгоритм функции … (по выбору)  (по выбору)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_––*

*3.2 Алгоритм функции … (по выбору)  (по выбору)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_––*

*3.3 Алгоритм функции … (по выбору)  (по выбору)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_––*

*4 Описание работы программы––––––           ––––––––––––––––\_\_\_\_\_\_\_\_\_\_\_\_–*

*4.1 Авторизация \_\_   ––––\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_––––*

*4.2 Модуль администратора \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   ––––––––*

*4.3 Модуль пользователя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   ––––––––*

*4.4 Исключительные ситуации \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   ––––––––*

*Приложение: Листинг кода с комментариями––––––––––––––––––––––––––––*

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

*––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––*

6. Консультант по работе *–––––––––Мельникова Елена Александровна–––––––––*

7. Дата выдачи задания –*\_\_.02.2020 г..*––––*––––––––––––––––––––––––––––––––*

8. Календарный график работы на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

*8.1 Разделы 1,2 – 15.03.2020 г. – 30% ––––––––––––––––––––––––––––––––––––8.2 Раздел 3 – 15.04.2020 г. – 20% –––––––––––––––––––––––– ––––––––––––––8.3 Раздел 4 – 15.05.2020 г. – 50% ––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––– ––––*

РУКОВОДИТЕЛЬ*–Мельникова Елена Александровна–*

(подпись)

Задание принял к исполнению *––\_\_.02.2020 г.\_\_\_9.09.20\_\_ \_\_ С.П. Иванов––––*

(дата и подпись студента)

**Содержание**

[1. Требования к программе 5](#_Toc39978607)

[1.1 Постановка задачи 5](#_Toc39978608)

[1.2 Исходные данные 5](#_Toc39978609)

[1.3 Функциональные требования 6](#_Toc39978610)

[1.3.1 Авторизация 6](#_Toc39978611)

[1.3.2 Работа с данными 7](#_Toc39978612)

[1.3.2.1 Модуль администратора 7](#_Toc39978613)

[1.3.2.1 Модуль пользователя 8](#_Toc39978614)

[1.3.3 Дополнительные функциональные возможности 8](#_Toc39978615)

[1.4 Требования к программной реализации 9](#_Toc39978616)

[2. Конструирование программы 10](#_Toc39978617)

[2.1 Разработка модульной структуры программы 10](#_Toc39978618)

[2.2 Выбор способа организации данных 10](#_Toc39978619)

[2.3 Разработка перечня пользовательских функций программы 12](#_Toc39978620)

[3. Разработка алгоритмов работы программы 14](#_Toc39978621)

[3.1 Алгоритм функции авторизации 14](#_Toc39978622)

[3.2 Алгоритм добавления новой книги в базу данных 15](#_Toc39978623)

[3.3 Алгоритм вывода отсортированных книг по определенному полю 16](#_Toc39978624)

[4. Описание работы программы 18](#_Toc39978625)

[4.1 Авторизация 18](#_Toc39978626)

[4.2 Модуль администратора 18](#_Toc39978627)

[4.3 Модуль клиента 21](#_Toc39978628)

# Требования к программе

# 1.1 Постановка задачи

Разработать программу учета книг в библиотеке. Для книг, хранящихся в библиотеке, задаются: регистрационный номер книги; автор; название; год издания; издательство; количество страниц; номер читательского билета (шесть цифр) последнего читателя; отметка о нахождении книги у читателя или в библиотеке в текущий момент.

Индивидуальное задание: вывести список книг с фамилиями авторов в алфавитном порядке, изданных после заданного года (год вводится с клавиатуры). Вывести список книг, находящихся в текущий момент у читателей.

Реализовать авторизацию для входа в систему, функционал администратора и функционал пользователя.

# 1.2 Исходные данные

1. Тема курсовой работы: Разработка программы учета книг в библиотеке
2. Язык программирования C#
3. Среда разработки Microsoft Visual Studio
4. Вид приложения – консольное
5. Парадигма программирования – объектно-ориентированная
6. Способ организации данных – классы
7. Способ хранения данных – файлы и база данных
8. Каждая логически завершенная подзадача программы должна быть реализована в виде отдельного метода
9. Построение программного кода должно соответствовать соглашению о коде «C# Code Convention»
10. К защите курсовой работы представляются: консольное приложение и пояснительная записка
11. Текст пояснительной записки оформляется в соответствии со стандартом предприятия СТП 01-2017

# 1.3 Функциональные требования

Ниже приведены функциональные требования к программе, которые должны быть соблюдены при реализации.

### 1.3.1 Авторизация

В программе должен быть реализован алгоритм авторизации – предоставления прав доступа пользователю. У пользователя должна быть возможность ввести свои персональные данные для получения доступа к функциональности приложения.

Приложение должно поддерживать 2 возможности входа:

1. В качестве администратора
2. В качестве пользователя

Для добавления новых пользователей в приложение должна быть реализована возможность импортирования пользователей через файл. Такая возможность должна быть доступна только через интерфейс администратора. Файл должен содержать основные данные о пользователе:

1. Логин
2. Пароль
3. Роль

Также у администратора должна быть возможность регистрировать новых пользователей, причем при создании должно соблюдаться условие уникальности логина.

### 1.3.2 Работа с данными

Приложение должно быть разделено на 2 функциональных модуля, доступ к которым зависит от роли авторизованного пользователя: модуль администратора и модуль пользователя.

### 1.3.2.1 Модуль администратора

В полномочия администратора входит:

1. Управление учетными записями пользователей
   1. Просмотр всех учетных записей
   2. Добавление новых учетных записей, включая администраторов
   3. Редактирование существующих учетных записей
   4. Удаление других учетных записей
2. Управление базой данных книг:
   1. Фиксация выдачи книги пользователю
   2. Фиксация возвращения книги обратно в библиотеку
   3. Добавление новых книг в базу данных
   4. Редактирование уже существующих в базе книг
   5. Удаление книг из базы данных
3. Просмотр базы данных книг:
   1. Просмотр всех добавленных книг и отслеживание их нахождения в библиотеке
   2. Просмотр книг с фильтрацией по определенному критерию
   3. Просмотр книг с сортировкой по определенному критерию
   4. Просмотр книг в алфавитном порядке изданных после заданного года
   5. Просмотр книг находящихся в текущий момент у читателей
4. Управление файлами данных:
   1. Создание файла данных
   2. Удаление файлов данных
   3. Загрузка данных из файла

### 1.3.2.1 Модуль пользователя

В полномочия пользователя входит:

1. Просмотр базы данных книг:
   1. Просмотр всех добавленных книг и отслеживание их нахождения в библиотеке
   2. Просмотр книг с фильтрацией по определенному критерию
   3. Просмотр книг с сортировкой по определенному критерию
   4. Просмотр книг в алфавитном порядке изданных после заданного года
   5. Просмотр книг находящихся в текущий момент у читателей

### Дополнительные функциональные возможности

В программе необходимо предусмотреть обработку исключительных ситуаций:

1. Ввод не соответствующих формату данных
2. Некорректный формат файла
3. Ввод номера записи выходящий за допустимый диапазон

Также, необходимо реализовать навигацию по программе с возможностью выхода из учетной записи.

Программа должна предполагать обратную связь с пользователем.

# 1.4 Требования к программной реализации

1. Названия переменных, методов и классов должны соответствовать своему логическому смыслу.
2. Все неизменяемые переменные должны быть обозначены ключевым словом const. В коде не должно быть «магических» чисел и строк.
3. Запрет на использование goto.
4. Код должен содержать комментарии при определении классов и при нетривиальной логике.
5. Каждый модуль и каждая сущность должны быть определены в своем классе.
6. Загрузка данных из файла должна сохранять данные в базу данных
7. Методы не должны быть слишком большими.
8. Код не должен содержать повторяющуюся логику. Весь повторяющийся код должен быть вынесен в отдельные методы.

# Конструирование программы

# 2.1 Разработка модульной структуры программы

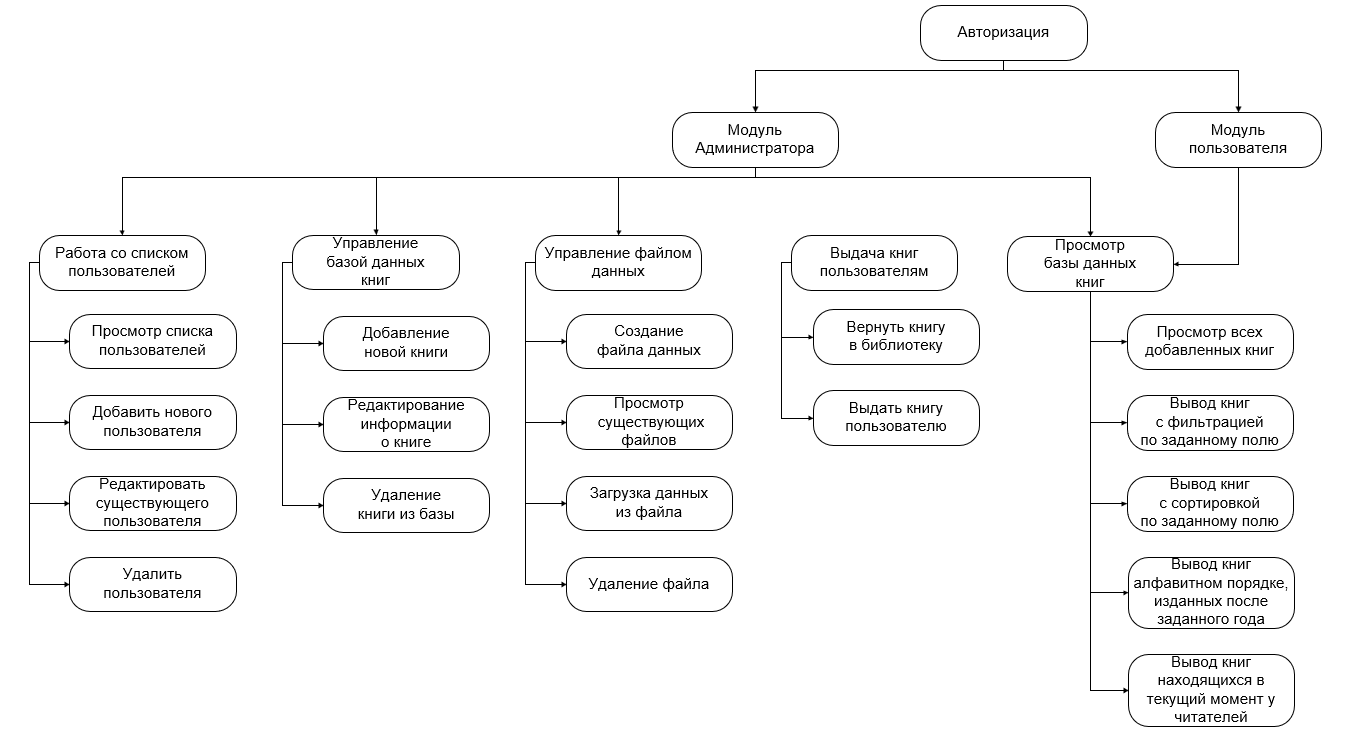


Рисунок 2.1 Модульная структура программы

# 2.2 Выбор способа организации данных

Все данные хранятся в четырех таблицах базы данных MS SQL Server. Для учетных записей, книг, авторов и издательств.

Для учетных записей предусмотрено 4 поля:

1. Идентификатор записи
2. Логин
3. Пароль
4. Роль (1 – для администратора, 0 – для пользователя)

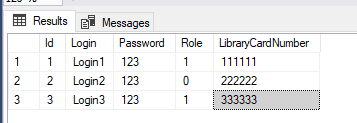


Рисунок 2.2.1 Структура таблицы данных пользователей

Для книг предусмотрено 9 полей:

1. Идентификатор записи
2. Регистрационный номер книги
3. Название книги
4. Количество страниц
5. Год публикации
6. Флаг нахождения книги в библиотеке
7. Внешний ключ на запись издателя
8. Внешний ключ на запись автора
9. Внешний ключ на последнего пользователя, взявшего книгу

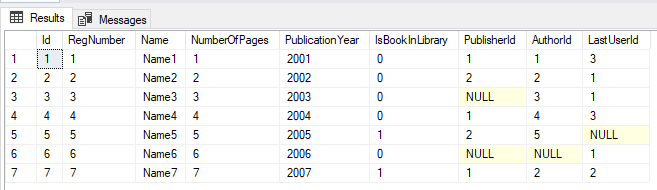


Рисунок 2.2.2 Структура таблицы данных книг

Для авторов предусмотрено 4 поля:

1. Идентификатор записи
2. Имя автора
3. Фамилия автора
4. Отчество автора

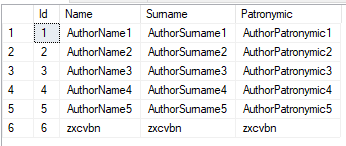


Рисунок 2.2.3 Структура таблицы данных авторов

Для издателей предусмотрено 2 поля:

1. Идентификатор записи
2. Название издателя

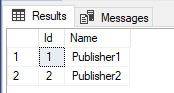


Рисунок 2.2.4 Структура таблицы данных издателей

# Разработка перечня пользовательских функций программы

1. Авторизация
   1. Ввод логина и пароля
   2. Поиск данных в базе данных
   3. В случае корректного ввода данных предоставлять доступ к функциональности приложения в зависимости от типа учетной записи
   4. В случае некорректного ввода данных выводить ошибку
2. Управление базой книг
   1. Добавление книг в базу данных
   2. Изменение информации о существующих книгах
   3. Удаление записи о книге
3. Управление файлов данных
   1. Создание файла данных
   2. Просмотр существующих файлов
   3. Загрузка данных из файла
   4. Удаление файла
4. Выдача книг пользователям
   1. Вернуть книгу в библиотеку
   2. Выдать книгу пользователю
5. Просмотр базы данных книг
   1. Просмотр всех добавленных книг
   2. Вывод книг с фильтрацией по заданному полю
   3. Вывод книг с сортировкой по заданному полю
   4. Вывод книг в алфавитном порядке изданных после заданного года
   5. Вывод книг находящихся в текущий момент у читателей

# Разработка алгоритмов работы программы

# 3.1 Алгоритм функции авторизации

Алгоритм начинается с ввода логина и пароля пользователем. Если пользователь оставит ввод пустым (сразу нажмет Enter), то пользователю выведется сообщение с ошибкой. Если ввод корректный, то произведется поиск в базе данных пользователя с соответствующим логином и паролем. В случае, если пользователя не существует в базе, выводится сообщение с ошибкой и пользователю предлагается ввести логин и пароль снова. Если пользователь есть в базе, то в зависимости от его типа, ему выводится соответствующее меню приложения.

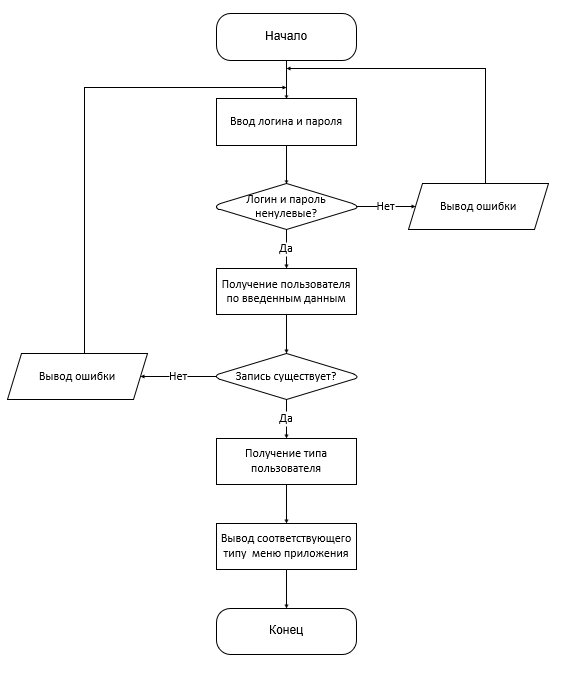


Рисунок 3.1 Блок-схема алгоритма авторизации

# 3.2 Алгоритм добавления новой книги в базу данных

Алгоритм начинается с ввода названия книги. Если была введена пустая строка, то выводится сообщение с ошибкой и предлагается еще раз ввести поле. После корректного ввода предлагается ввод количества страниц, если было введено не число, то выводится сообщения об ошибке и предлагается еще раз ввести количество страниц. Далее, предлагается ввод года публикации. Диапазон ввода определяется через константное значение в коде, если оно выходит за его рамки, то выводится сообщения об ошибке. После этого предлагается ввод регистрационного номера книги с валидацией на ввод пустой строки.

Затем приложение предлагает на выбор 2 опции: добавить нового автора, либо выбрать из уже существующих в базе. В случае первой опции, пользователю необходимо ввести имя, фамилию и отчество нового автора. В случае выбора автора из существующих выполняется запрос в базу данных, и пользователь может выбрать уже существующего автора.

Далее идет идентичный алгоритм выбора, либо создания нового издателя книги. После чего идет формирование модели данных из всех введенных данных, которая отправляется на бизнес-слой приложения.

На бизнес-слое приложения осуществляется проверка на существование регистрационного номера в базе. Если такой номер существует, то возвращается соответствующая ошибка, которая выводится пользователю. Если же такого номера нет, то происходит сохранение записи в базе.

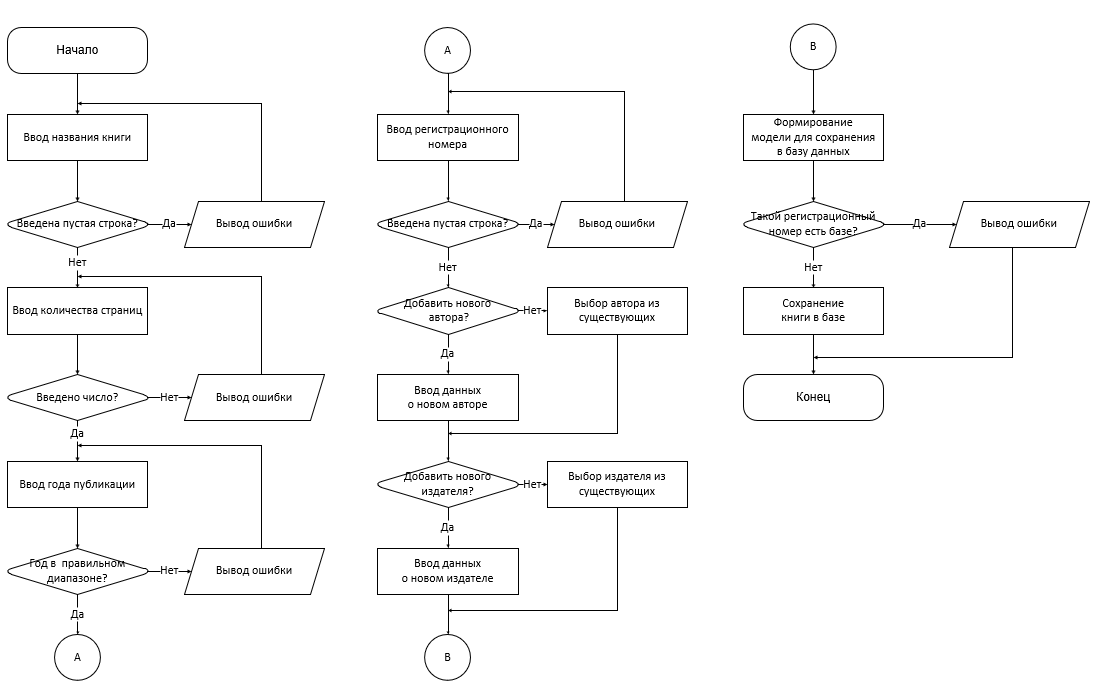


Рисунок 3.2 Блок-схема алгоритма добавления новой

книги в базу данных

# 3.3 Алгоритм вывода отсортированных книг по определенному полю

Алгоритм начинается с вывода всех доступных критериев сортировки из перечисления BookFilteringType. В случае, если пользователь выходит за диапазон выбора критерия, то выводится сообщение с ошибкой и предлагается ввод еще раз. После корректного ввода пользователю предлагается выбор направления сортировки: по возрастанию или по по убыванию. После валидации ввода отправляется запрос на базу данных с указанными критериями. После получения данных, в цикле формируются модели для отображения в консоли, которые после отображаются посредством функции PrintBookInfo.

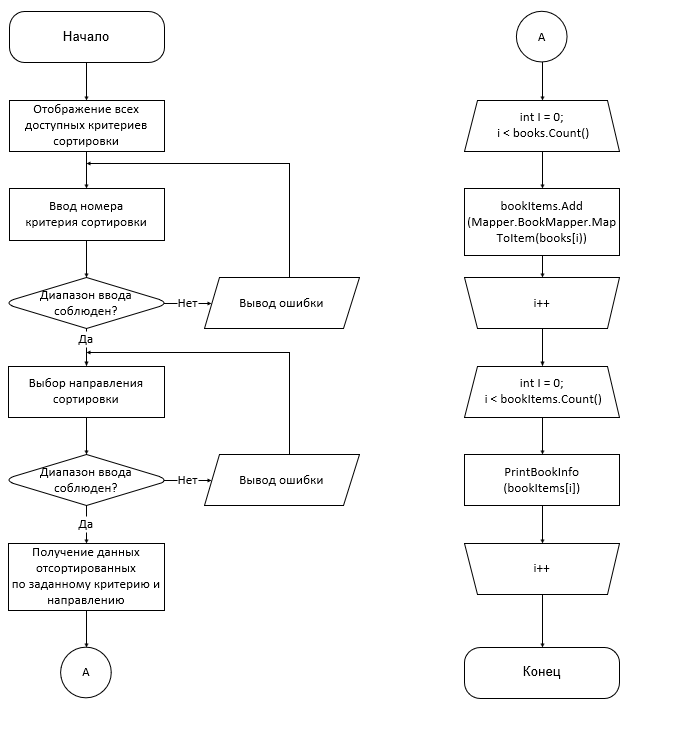


Рисунок 3.3 Блок-схема алгоритма получения отсортированных книг

# Описание работы программы

# 4.1 Авторизация

При входе в приложение открывается начальная страница приложения, предлагающая пользователю авторизоваться. На странице авторизации вводится логин и пароль. После чего, в зависимости от типа пользователя, открывается меню пользователя или меню администратора.

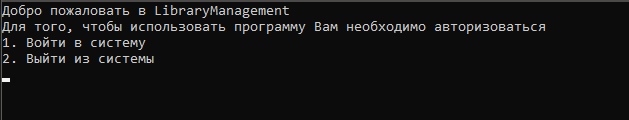


Рисунок 4.1 Начальная страница приложения

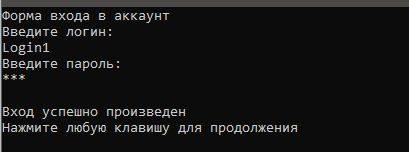


Рисунок 4.2 Форма авторизации

# Модуль администратора

После авторизации, если тип указанного пользователя администратор, открывается меню администратора. На выбор администратору предлагается несколько модулей:

* Управление учетными записями пользователей
* Управление книгами
* Просмотр книг
* Выдача книг пользователям
* Работа с файлами данных
* Вернутся на окно входа в учетную запись
* Выход из программы

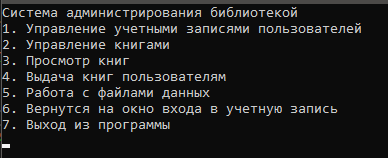


Рисунок 4.3 Главное меню администратора

Если администратор выберет модуль управления учетными записями, то приложения предлагает на выбор: просмотр всех пользователей, просмотр всех пользователей с книгами, добавление нового пользователя, редактирование данных и удаление пользователя.

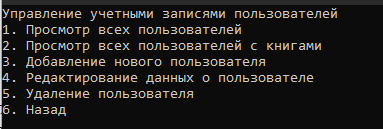


Рисунок 4.4 Меню управления пользователями

Администратор может выбрать добавление нового пользователя, в которой он сможет создать новую учетную запись с определенным типом, в которую можно будет потом авторизоваться.

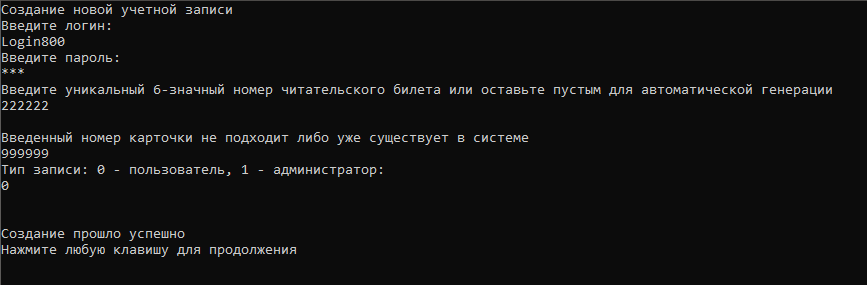


Рисунок 4.5 Форма создания пользователя

Также администратору доступно добавление новых книг в базу данных

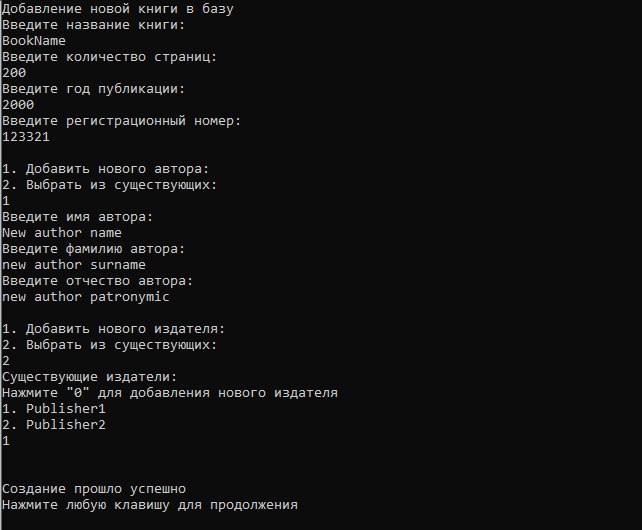


Рисунок 4.6 Форма добавления книги

# Модуль клиента

При входе в клиентский модуль приложения, пользователю предлагается 3 операции:

* Просмотр базы данных книг
* Вернутся на окно авторизации
* Выйти из программы

При выборе модуля просмотра книг, клиенту предлагается на выбор несколько опций выбора книг:

* Просмотр всех добавленных книг
* Вывод книг с фильтрацией по заданному полю
* Вывод книг в отсортированном виде
* Вывод книг в алфавитном порядке, изданных после заданного года
* Вывод книг находящихся в текущий момент у читателей
* Вернуться назад

В зависимости от выбранной опции пользователь может настроить вывод книг, подходящий ему наиболее всего. При выборе просмотра всех добавленных книг, пользователю выводится список:

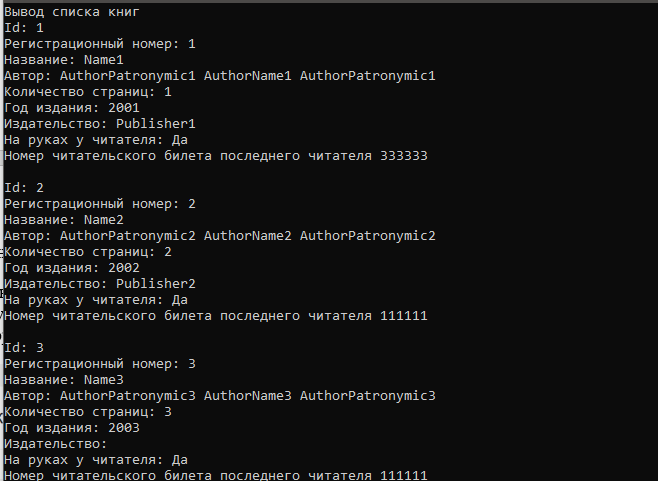


Рисунок 4.4 Вывод списка книг

# Исключительные ситуации

В приложении реализована обработка ввода некорректных данных:

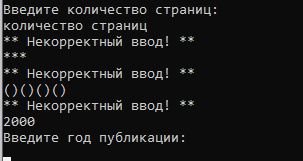


Рисунок 4.5 Обработка ввода чисел

Реализованы несколько проверок на уникальность данных

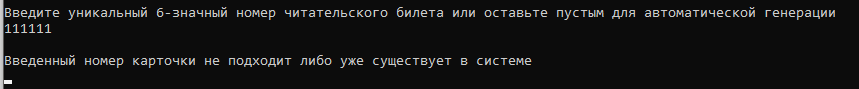


Рисунок 4.6 Обработка ввода уникальных данных

Добавлены проверки на выход за допустимый диапазон:

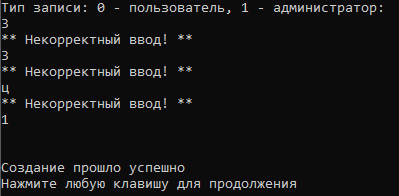


Рисунок 4.7 Проверка на выход за диапазон