

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Санкт-Петербургский государственный электротехнический университет

“ЛЭТИ” им.В.И.Ульянова (Ленина) »

Кафедра ВТ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ

по дисциплине «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

**«Создание программного комплекса средствами объектно-ориентированного программирования»**

Выполнил студент Бондаренко Р.В.

Факультет КТИ

Группа № 1307

Руководитель Павловский М.Г.

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2013 г

1.Техническое задание

## 1.1 Введение

Программный комплекс (ПК) администрирования библиотеки предназначен для использования в составе системы программно-информационного обеспечения учета и администрирования выдача и возврат книг.

## 1.2 Основание для разработки

Основанием для разработки ПК «Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» является курсовой проект по дисциплине «Объектно-ориентированное программирование».

## 1.3 Назначение разработки

ПК «Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» должен входить в состав автоматизированной системы учета и администрирования информации, и предназначен для автоматизации деятельности лица (ОЛ), ответственного за учет выдаче и возврата книг.

ПК «Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» предназначен для автоматизации следующих процессов:

* учет и администрирование информации о выдаче книг;
* учет и администрирование информации о возврате книг;
* получение справочной информации по выдаче и о возврате книг;

## 1.4 Требования к программе

### 1.4.1 Требования к функциональным характеристикам

*1.4.1.1 Перечень функций*

ПК «Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» должен обеспечивать выполнение следующих функций:

* просмотр, добавление, удаление и изменение в базы данных (БД);
* выдача справочной информации, хранимой в БД, по запросам ОЛ.

*1.4.1.2 Требования к составу выполняемых функций*

*1.4.1.2.1 Функция «*просмотр, добавление, удаление и изменение в базы данных (БД)*»*

Ввод, просмотр, добавление, удаление и изменение в БД должны обеспечивать ведение и хранение следующих данных:

* данных о выдаче книг;
* данных о возврате книг.

*1.4.1.2 Требования к организации и форме представления выходных данных*

Выходные данные должны быть представлены в виде таблицы содержащий описание необходимых информационных объектов, выполненного посредствам представления его характеристик.

*1.4.1.3 Требования к организации и форме представления входных данных*

Входная информация для задачи «Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» содержится в приходно-расходной документации. Ввод исходных данных должен осуществляется ОЛ в режиме диалога. Вводимые данные являются значениями характеристик (атрибутов) информационных объектов.

### 1.4.2 Требования к надежности

ПК «Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» должен устойчиво функционировать при соблюдении гарантии устойчивого функционирования операционной системы и системы управления базой данных. Под устойчивой работой ПК понимается непрерывное функционирование программы в отсутствии критических сбоев, приводящих к аварийному завершению. Кроме того, должен быть обеспечен контроль входных данных на предмет соответствия предполагаемому типу.

### 1.4.3 Условия эксплуатации

Выполнение ПК «Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» своих функций должно быть обеспечено для однопользовательского режима работы с монопольным доступом к базе данных.

### 1.4.4 Требование к составу и параметрам технических средств

Задача должна решаться на ПЭВМ типа IBM РС или совместимой с ней с процессором Pentium III 500 и выше, ОЗУ не менее 128Мб, HDD не менее 4 Гб, монитор SVGA (цветной)15", видеокарта 64 Мб, клавиатура 102 кл., манипулятор типа "мышь".

### 1.4.5 Требование к информационной и программной совместимости

Выходная и входная информация ПК «Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» должна быть удобна для визуального восприятия. ПК должен быть выполнен на языке программирования высокого уровня Java и должен быть совместим с операционной системой Windows.

Обязательными требованиями при разработке кода ПК являются использование следующих конструкций языка Java:

* закрытые и открытые члены классов;
* наследование;
* конструкторы с параметрами;
* абстрактные базовые классы;
* виртуальные функции;
* обработка исключительных ситуаций;
* динамическое создание объектов.

## 1.5 Требования к программной документации

Программная документация (ПД) должна удовлетворять требованиям стандартов ЕСПД.

Документация должна быть представлена в следующем составе:

1. описание процесса проектирования ПК;
2. руководство оператора;
3. исходные тексты ПК.

## 1.6 Стадии и этапы разработки

1. Разработка технического задания;
2. Описание вариантов использования ПК;
3. Создание прототипа интерфейса пользователя;
4. Разработка объектной модели ПК;
5. Построение диаграмм программных классов;
6. Описание поведения ПК;
7. Построение диаграмм действий;

## 1.7 Порядок контроля и приемки

В процессе приема работы устанавливается соответствие ПК и прилагаемой документации требованиям, обозначенным в техническом задании.

# 2 Проектирование ПК

## 2.1 Описание вариантов использования ПК

Развернутое описание функциональных требований осуществляется на этапе проектирования комплекса. Для того чтобы детализировать требования, необходимо выделить процессы, происходящие в заданной предметной области. Описание таких процессов на UML выполняется в виде прецедентов (use case). Прецеденты являются сценарием или вариантом использования ПК при взаимодействии с внешней средой. Они являются продолжением описаний требований и функциональных спецификаций, указанных в техническом задании. Прецедент изображается в виде эллипса, в котором содержится имя прецедента. Название прецедента обязательно включает в себя глагол, выражающий суть выполняемой функции. С помощью прецедентов описывается функционирование ПК с точки зрения внешнего пользователя, который называется в UML актором (actor). Актор представляет собой любую внешнюю по отношению к моделируемой системе сущность (человек, программная система, устройство), которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей или решения частных задач. Актор на диаграмме изображается пиктограммой в виде человечка, под которым указано его имя. Совокупность функций, реализуемых ПК, изображается в виде диаграммы (use case diagram). Для построения диаграммы необходимо определить акторы, прецеденты (функции) и взаимоотношение между акторами и прецедентами, и между прецедентами, если один прецедент расширяет или использует другой. В языке UML для вариантов использования и действующих лиц поддерживается несколько типов связей. Это связи коммуникации (communication), использования (uses) и расширения (extends).

Связь коммуникации — это связь между прецедентом и актором. На языке UML связь коммуникации изображают в виде стрелки. Направление стрелки показывает, кто инициирует коммуникацию. При задании коммуникации необходимо указать данные, которые вводит или получает пользователь. Кроме данных на концах стрелки можно указать кратности отношения, которые характеризуют количество взаимодействующих между собой акторов и прецедентов. На диаграммах прецедентов наиболее распространенными являются две формы записи кратности 1 и 1 .. \*. Первая форма записи означает, что один актор (прецедент) участвует во взаимодействии, а вторая форма записи, что один или несколько акторов (прецедентов) участвуют во взаимодействии.

Связь использования предполагает, что один прецедент всегда применяет функцио­нальные возможности другого. С помощью таких связей структурируют прецеденты, показывая тем самым, какой прецедент является составной частью другого прецедента. Такой включаемый прецедент является абстрактным прецедентом в том смысле, что он не может исполняться независимо от других прецедентов, а лишь в их составе. Связь использования изображается с помощью стрелок и слова «uses» (использование). Направление стрелки указывает, какой прецедент используется для реализации функциональности другого прецедента.

Связь расширения задается в том случае, если необходимо показать родственные отношения между двумя прецедентами. Один из них является базовым, а другой его расширением. Базовый прецедент не зависит от расширяющих прецедентов и способен функционировать без них. С другой стороны, расширяющие прецеденты без базового прецедента функционировать не могут. Связи расширения изображают в виде стрелки со словом «extends» (расширение), которая имеет направление от базового прецедента к расширяемому.

Прецеденты необходимо ранжировать, чтобы в начальных циклах разработки реализовать наиболее приоритетные из них. Разбиение функциональности системы на отдельные прецеденты служит примерно той же цели, что и разбиение сложного алгоритма на подпрограммы. Основная стратегия должна заключаться в том, чтобы сначала сконцентрировать внимание на тех прецедентах, которые в значительной мере определяют базовую архитектуру ПК.

Диаграмма прецедентов представлена на рис. 2.1.

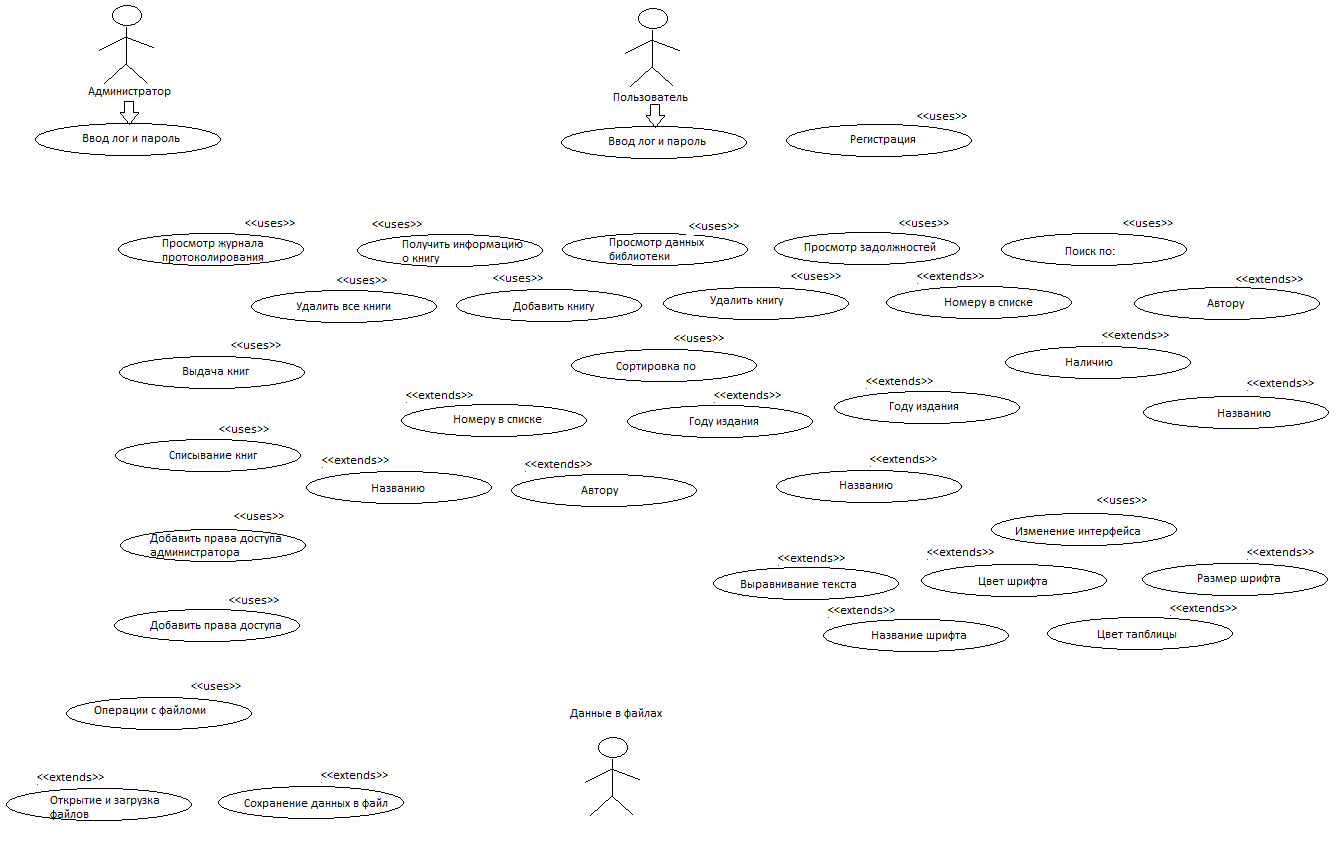
****

Рис.2.1 Диаграмма прецедентов

## 2.3 Создание прототипа интерфейса пользователя

Описание прецедента выражает общую сущность процесса без детализации его реализации. Проектные решения, связанные с интерфейсом пользователя, при этом опускаются. Для разработки пользовательского интерфейса необходимо описать процесс в терминах реальных проектных решений, на основе конкретных технологий ввода-вывода информации. Когда речь идет об интерфейсе пользователя, прецеденты разбиваются на экранные формы, которые определяют содержимое диалоговых окон и описывают способы взаимодействия с конкретными устройствами. Для каждой экранной формы указываются поля ввода и перечень элементов управления, действия пользователя (нажать кнопку, выбрать пункт меню, ввести данные, нажать правую/левую кнопку мыши) и отклики системы (отобразить данные, вывести подсказку, переместить курсор). Такое описание интерфейса представляется в виде таблицы экранных форм.

Рис. 2.2 –экранная форма входа и регистрации; рис.2.3 –главное экранная форма; рис. 2.4 – экранная форма выдача книг; рис. 2.5 – экранная форма возврата книг; рис. 2.6 – экранная форма выдача прав доступа администратора; рис. 2.7 – экранная форма выдача прав доступа; рис. 2.8 – экранная форма регистрации. В табл. 2.1 представлено описание экранных форм.

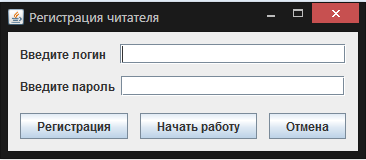


Рис. 2.2.Экранная форма входа и регистрации

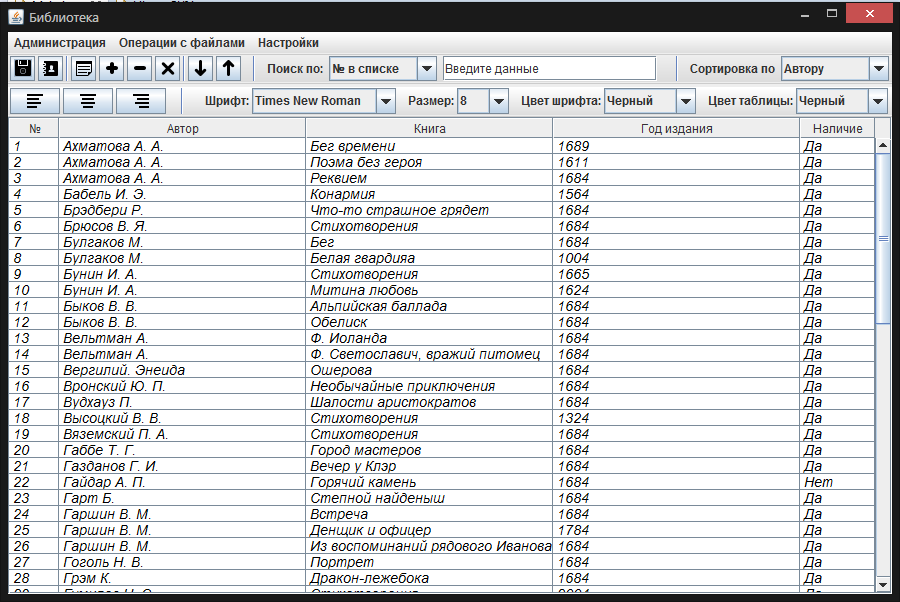


Рис.2.3. Главное экранная форма

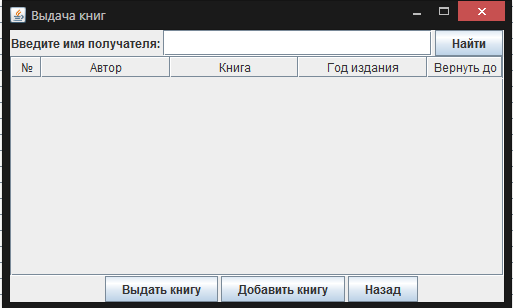


Рис. 2.4. Экранная форма выдача книг

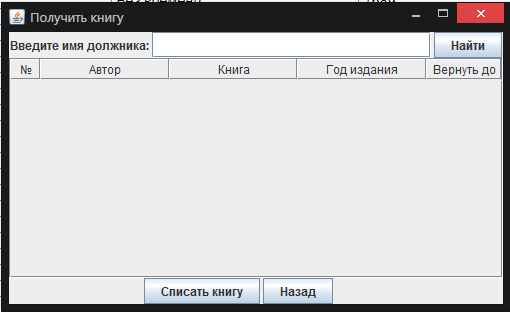


Рис. 2.5. Экранная форма возврата книг

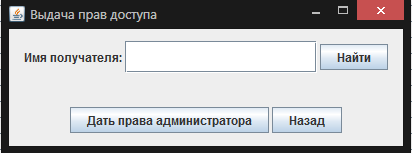


Рис. 2.6. Экранная форма выдача прав доступа администратора

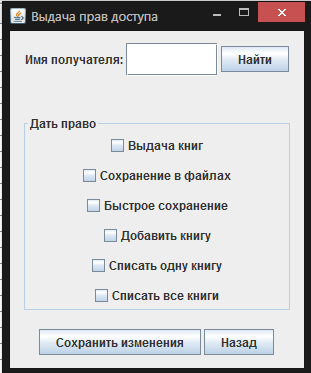


Рис. 2.7. Экранная форма выдача прав доступа

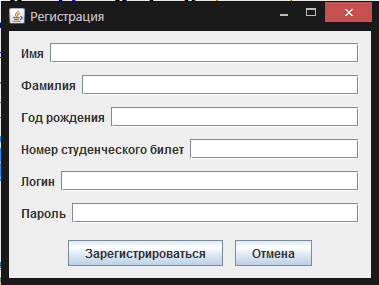


Рис. 2.8. Экранная форма регистрации

Таблица 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Экранная форма | Элементы управления | Действия пользователя | Отклик системы |
| Регистрация читателя | Кнопки:  «Регистрация»,  «Начать работу», «Отмена» | Открыть окно «Регистрация»  (нажать кнопку «Регистрация»). | Открыть диалог «Библиотека». |
| Открыть окно  «Начать работу»  (ввести логин и пароль и нажать кнопку «Начать работу»). | Открыть диалог «Регистрация». |
| Выйти из программы (нажать кнопку «Отмена») | Закрыть окно «Регистрация читателя». |
| Библиотека | Кнопки:  «Быстрое сохранение в \*.txt»  «Задолженности»  «Получить информацию»  «Добавить книгу»  «Удалить выделенную книгу»  «Удалить всё»  «Скроллинг вверх»  «Скроллинг вниз»  «Текст с лева»  «Текст по центру»  «Текст с права»  Список элементов с полями :  «Поиск по»: № в списке,  автору, книге, году издания.  «Сортировка по»: автору, книге, году издания, наличию.  «Шрифт»: Times New Roman, Calibri, Arial, Cambria, GungsuhChe.  «Размер»: 8, 9, 10, 11, 12, 14, 16, 18, 20.  «Цвет шрифта»: чёрный, красный, синий, жёлтый, зеленый, белый.  «Цвет таблицы»: чёрный, красный, синий, жёлтый, зеленый, белый.  Список элементов меню : «Администрация»: перезайти, журнал протоколирования, выдать книгу, получить книгу, выход.  «Операция с файлами»:  Открыть в \*.txt, сохранить в \*.txt, открыть в \*.xml,  сохранить в \*. xml,  сохранить в \*.pfd,  сохранить в \*.html.  «Настройки»: добавить администратора, добавить права доступа. | Нажатие кнопки «Быстрое сохранение в \*.txt». | Сохранение данных таблицы в БЗ. |
| Нажатие кнопки «Задолженности». | Вывод окна со списком задолженностей. |
| Нажатие кнопки  «Получить информацию», после нажать на нужную книгу. | Изменение главного окна и вывод с правой части информаций о книге (при условий, если книга есть). |
| Нажатие кнопки  «Добавить книгу». | Добавляется новая строка для заполнения данными. |
| Нажать на нужную книгу, после нажать кнопку  «Удалить выделенную книгу». | Удаление выделенной книги. |
| Нажатие кнопки  «Удалить всё». | Все данные из таблицы будут удалены. |
| Нажатие кнопки  «Скроллинг вверх». | Скроллинг вверх таблицы. |
| Нажатие кнопки  «Скроллинг вниз». | Скроллинг вниз таблицы. |
| Нажатие кнопки  «Текст с лева». | Текст становиться по левую сторону. |
| Нажатие кнопки  «Текст по центру». | Текст становиться по центру сторону. |
| Нажатие кнопки  Таблица 2.1. Продолжение  «Текст с права» | Текст становиться по правую сторону. |
| Поиск по элементу из списка по ключевому слову  (нажать на список «Поиск по», выбрать элемент поиска, ввести ключевое слово поиска, нажать на кнопку «Найти»). | Выделение введенного ключевого слова, если оно найдено, бездействие в ином случае. |
| Сортировка элементов из списка  (нажать на список «Сортировка по», выбрать элемент сортировки). | Сортировка элементов по выбранному элементу из списка. |
| Изменение шрифта текста в таблице  (нажать на список «Шрифт», выбрать название шрифта). | Изменение шрифта текста в таблице, по выбранному шрифту. |
| Изменение размера текста в таблице  (нажать на список «Размер», выбрать размер шрифта). | Изменение размера текста в таблице, по выбранному размеру. |
| Изменение цвета шрифта текста в таблице  (нажать на список «Цвет шрифта», выбрать цвет шрифта). | Изменение цвета шрифта текста в таблице, по выбранному цвету шрифта. |
| Изменение цвета таблицы  (нажать на список «Цвет таблицы», выбрать цвет таблицы). | Изменение цвета таблицы, по выбранному цвету таблицы. |
| Смена пользователя  «Перезайти»  (нажать меню «Администрация», после нажать подпункт меню «Перезайти»). | Открытие диалога «Регистрация читателя». |
| Открыть окно  «Журнал протоколирования»  (нажать меню «Администрация», после нажать подпункт меню «Журнал протоколирования»). | Открытие диалога «Журнал протоколирования». |
| Открыть окно  «Выдать книгу»  (нажать меню «Администрация», после нажать подпункт меню «Выдать книгу»). | Открытие диалога  «Выдача книг». |
| Открыть окно  «Получить книгу»  (нажать меню «Администрация», после нажать подпункт меню «Получить книгу»). | Открытие диалога  «Получить книгу». |
| Закрытие программы  «Выход»  (нажать меню «Администрация», после нажать подпункт меню «Выход»).  Таблица 2.1. Продолжение | Выход из программы. |
| Открыть окно для сохранения или загрузки (указанных) данных в определенном формате  (нажать меню «Операция с файлами», выбрать нужный список подменю). | Сохранение или загрузка (указанных) данных, в определенном формате. |
| Открыть окно  «Выдача прав доступа администратора»  (нажать меню «Настройки», после нажать подпункт меню «Добавить администратора»). | Открытие диалога  «Выдача прав доступа администратора» |
| Открыть окно  «Выдача прав доступа»  (нажать меню «Настройки», после нажать подпункт меню «Добавить права доступа»). | Открытие диалога  «Выдача прав доступа» |
| Выдача книг | Кнопки:  «Выдать книгу»  «Добавить книгу»  «Назад»  «Найти»  Список элемента для ввода данных:  «Введите имя получателя» | Найти пользователя в БД  (ввести логин пользователя в «Введите имя получателя», который берет книгу и нажать на кнопку «Найти») | Вывод сообщения, есть ли данный пользователь в БД, если есть, то откроется его картотека. |
| Добавить строку для заполнения информации о книге  (нажать кнопку «Добавить книгу»). | Добавление строки для заполнения информаций. |
| Добавить книгу пользователю  (нажать кнопку «Выдать книгу»). | Добавление книги пользователю в картотеку. |
| Вернуться в главное диалоговое окно  (нажать кнопку «Назад») | Закрытие данного окна. |
| Получить книгу | Кнопки:  «Списать книгу»  «Назад»  «Найти»  Список элемента для ввода данных:  «Введите имя должника» | Найти пользователя в БД  (ввести логин пользователя в «Введите имя должника», который возвращает книгу и нажать на кнопку «Найти») | Вывод сообщения, есть ли данный пользователь в БД, если есть, то откроется его картотека. |
| Списать заданную книгу с картотеки пользователя  (нажать кнопку «Списать книгу»). | Списывание заданной книги с картотеки пользователя. |
| Вернуться в главное диалоговое окно  (нажать кнопку «Назад») | Закрытие данного окна. |
| Выдача прав доступа администратора | Кнопки:  «Найти»  «Назад»  «Добавить права администратора»  Список элемента для ввода данных:  «Введите имя» | Найти пользователя в БД  (ввести логин пользователя в «Введите имя», который получит права администратора и нажать на кнопку «Найти»).  Таблица 2.1. Продолжение | Вывод сообщения, есть ли данный пользователь в БД. |
| Добавить заданному пользователю, права администратора  (нажать на кнопку «Добавить права администратора»). | Заданный пользователь станет администратором. |
| Вернуться в главное диалоговое окно  (нажать кнопку «Назад») | Закрытие данного окна. |
| Выдача прав доступа | Кнопки:  «Найти»  «Назад»  «Сохранить изменения»  Список элемента для ввода данных:  «Введите имя» | Найти пользователя в БД  (ввести логин пользователя в «Введите имя», который получит права администратора и нажать на кнопку «Найти»). | Вывод сообщения, есть ли данный пользователь в БД. |
| Добавить заданному пользователю, выбранные права  (нажать на кнопку «Сохранить изменения»). | У заданного пользователя появятся или уменьшатся права. |
| Вернуться в главное диалоговое окно  (нажать кнопку «Назад») | Закрытие данного окна. |
| Регистрация | Кнопки:  «Зарегистрироваться»  «Назад»  Список элемента для ввода данных:  «Имя»  «Фамилия»  «Год рождения»  «Номер студенческого»  «Логин»  «Пароль» | Зарегистрировать  (заполнение полей: «Имя»,  «Фамилия», «Год рождения»,  «Номер студенческого»,  «Логин», «Пароль» , после нажать на кнопку «Зарегистрироваться»). | Вывод сообщения, что пользователь успешно зарегистрировался (в противном случае, сообщение об определенной ошибки). |
| Вернуться в главное диалоговое окно  (нажать кнопку «Назад») | Закрытие данного окна. |

## 2.3. Разработка объектной модели ПК

Объектная модель не описывает структуру ПК, она отображает основные понятия предметной области в виде совокупности типов объектов (сущностей). Сущности строятся путем выделения их из предметной области и анализа прецедентов. На диаграмме сущность обозначается прямоугольником, внутри которого записывается имя сущности, ее атрибуты и операции.

Атрибуты описывают свойства сущности. В объектную модель включаются те атрибуты, для которых определены соответствующие требования или для которых предполагается хранить определенную информацию. Атрибут характеризуется именем и типом. Для атрибута рекомендуется использовать простые типы данных (число, строка, дата, время и другие).

Описание операций помогает определить поведение объектов сущности. На этом этапе, прежде всего, определяется внутреннее поведение каждого объекта сущности, без учета взаимодействия с другими объектами предметной области. На диаграмме обычно указывается только имя операции, а ее подробное описание приводится в отдельной таблице. В таблице должно содержаться краткое описание назначения операции, ее имя и список входных и выходных параметров.

Ассоциация между сущностями отражает некоторое бинарное отношение между ними. Ассоциация обозначается проведенной между сущностями линией, с которой связывается определенное имя. Имя записывается в глагольной форме, и оно должно отражать семантический смысл отношения. Стрелка на линии указывает, в каком направлении нужно читать имя. На концах линии могут содержаться выражения, определяющие количественную связь между экземплярами сущности (кратность). Кратность определяет, сколько экземпляров одной сущности может быть ассоциировано с одним экземпляром другой сущности. Примеры кратностей:

0 ..\* - нуль или больше,

1 .. \* - один или больше,

1 – ровно один.

Необходимо устанавливать отношения ассоциации между двумя сущностями в том случае, если объект одной сущности должен знать об объекте другой. Прежде всего, следует включать в модель те ассоциации, которые отражают структурные отношения («содержит», «включает», «хранит» и т.д.), или те, которые должны сохраняться в течение некоторого времени.

Диаграмма сущностей представлена на рис. 2.7. Детальное описание операций представлено в табл. 2.2.

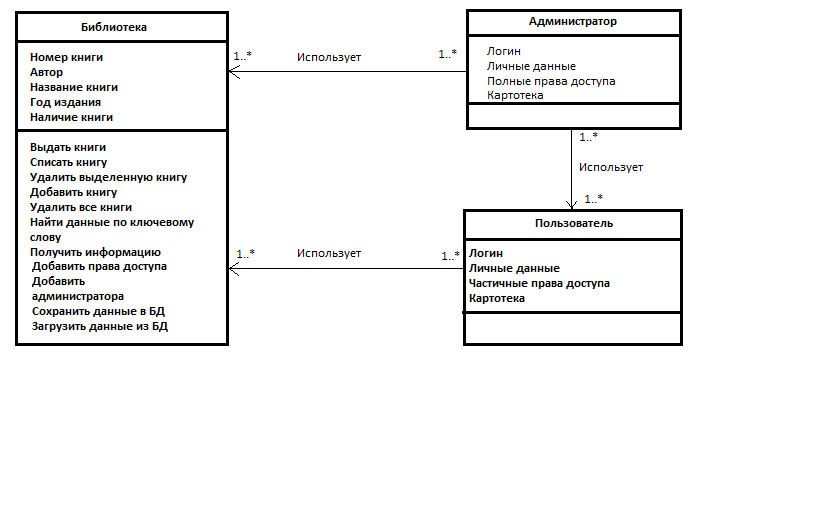


Рис. 2.7. Диаграмма сущностей

Таблица 2.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Сущность | Имя операции | Параметры операции | | | Тип возвращаемого значения | Назначение операции |
| Вид | Название | Тип |
| Библиотека | Выдать книгу | Вх. | Автор | Строка | БД | Добавляет в пользовательскую картотеку заданную книгу. |
| Вх. | Название книги | Строка |
| Вх. | Год издания | Строка |
| Вх. | Дата возврата | Строка |
| Списать книгу | Вх. | Автор | Строка | БД | Удаление с пользовательской картотеки заданную книгу. |
| Вх. | Название книги | Строка |
| Вх. | Год издания | Строка |
| Вх. | Дата возврата | Строка |
| Добавить книгу | Вх. | Автор | Строка | Пусто | Создает объекта с заданными значениями. |
| Вх. | Название книги | Строка |
| Вх. | Год издания | Строка |
| Вх. | Наличие книги | Строка |
| Удалить выделенную книгу | Вх. | Автор | Строка | Пусто | Удаление объекта. |
| Вх. | Название книги | Строка |
| Вх. | Год издания | Строка |
| Вх. | Наличие книги | Строка |
| Удалить все книги | Вх. | Автор | Строка | Пусто | Удаление всех объектов. |
| Название книги | Строка |
| Год издания | Строка |
| Наличие книги | Строка |
| Найти данные по ключевому слову | Вх. | Автор | Строка | Пусто | Поиск данных по ключевому слову. |
| Вх. | Название книги | Строка |
| Вх. | Год издания | Строка |
| Вх. | Наличие книги | Строка |
| Получить информацию | Вх. | Название книги | Строка | Список | Получает информацию о заданной книге. |
| Добавить права доступа | Вх. | Логин | Строка | Пусто | Добавляет права доступа, заданному пользователю. |
| Добавить администратора | Вх. | Логин | Строка | Пусто | Даёт право администратора, заданному пользователю. |
| Сохранить данные в БД | Вх. | Автор | Строка | Пусто | Сохраняет все данные в указанный файл. |
| Вх. | Название книги | Строка |
| Вх. | Год издания | Строка |
| Вх. | Наличие книги | Строка |
| Загрузить данные из БД | Вх. | Автор | Строка | Пусто | Загружает все данные из указанного файла. |
| Вх. | Название книги | Строка |
| Вх. | Год издания | Строка |
| Вх. | Наличие книги | Строка |

## 2.4 Построение диаграммы программных классов

Диаграмма классов (class diagram) иллюстрирует спецификации будущих программных классов и интерфейсов. Она строится на основе объектной модели. В описание класса указываются три раздела: имя класса, состав компонентов класса и методы класса. Графически класс изображается в виде прямоугольника. Имя программного класса может совпадать с именем сущности или быть другим. Но поскольку для записи идентификаторов переменных в языках программирования используют латинские буквы, то и имена программных классов и имена их атрибутов, как правило, записываются латинскими буквами. Атрибуты и операции класса перечисляются в горизонтальных отделениях этого прямоугольника. Атрибутам и методам классов должны быть присвоены права доступа. Права доступа помечаются специальными знаками:

+ - означает открытый (public) доступ;

— - означает скрытый (private) доступ;

# - означает наследуемый (protected) доступ.

При описании атрибутов после двоеточия указывается их тип, а при описании методов класса возвращаемое значение (для конструкторов возвращаемое значение не указывается).

В диаграмме классов могут вводиться дополнительно новые атрибуты, операции и связи или осуществляться конкретизация ассоциаций, указанных в объектной модели. На диаграмме классов могут быть три вида отношений: ассоциация, агрегация и наследование.

На диаграмме классов ассоциация имеет такое же обозначение, как и в объектной модели. На линиях ассоциации может присутствовать стрелка. Это стрелка видимости, которая показывает направление посылки запросов в ассоциации. Стрелка видимости также показывает, какой из классов содержит компоненты для реализации отношения ассоциации, иными словами, кто является инициатором посылки запроса к другому объекту. Ассоциация без стрелки является двунаправленной.

Агрегирование - это отношение между классами типа целое/часть. Агрегируемый класс в той или иной форме является частью агрегата. На практике это может быть реализовано по-разному. Например, объект класса-агрегата может хранить объект агрегируемого класса, или хранить ссылку на него. Агрегирование изображается на диаграмме полым ромбом на конце линии со стороны агрегирующего класса (агрегата). Если агрегируемый объект может быть создан только тогда, когда создан агрегат, а с уничтожением агрегата уничтожаются и все агрегируемые объекты, то такое агрегирование называется сильным и отображается в виде закрашенного ромба.

Наследование - это отношение типа общее-частное между классами. Его следует вводить в том случае, когда поведение и состояние различных классов имеют общие черты. Наследование связывает конкретные классы с общими или в терминологии языков программирования производные классы (подклассы) с базовыми классами (суперклассами). На диаграммах наследование изображается в виде стрелки с полым треугольником, идущей от производного класса к базовому. Если один производный класс наследует несколько базовых, то такое наследование называется множественным.

Диаграмма классов представлена на рис. 2.8.

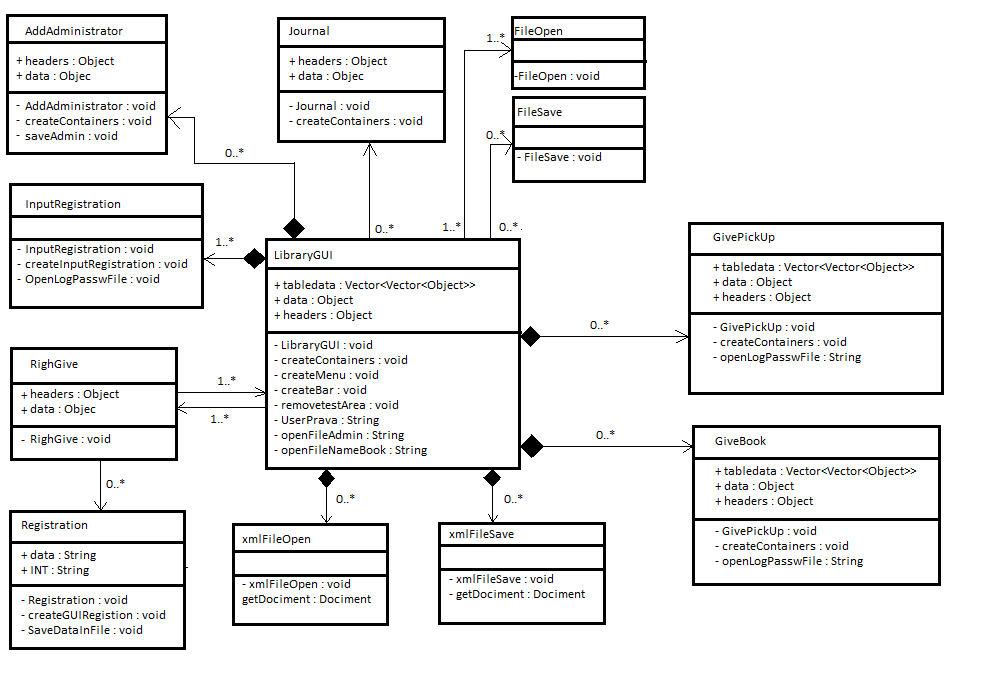


Рис.2.8. Диаграмма классов

## 2.5 Описание поведения ПК

Поведение ПК представляет собой описание того, какие действия выполняет ПК, без определения механизма их реализации. Одной из составляющей такого описания является диаграмма последовательностей (seguence diagram). Диаграмма последовательностей является схемой, которая для определенного сценария прецедента показывает генерируемые пользователями и объектами события (запросы) на выполнение некоторой операции и их порядок. Диаграммы последовательности имеют две размерности: вертикальная представляет время, горизонтальная - различные объекты. Чтобы построить диаграмму последовательностей необходимо выполнить следующие действия:

1. Идентифицировать пользователей и объекты программных классов, участвующие в начальной стадии реализации сценария прецедента, и их изображения в виде прямоугольников расположить наверху в одну линию. Для каждого пользователя и объекта нарисовать вертикальную пунктирную линию, которая является линией их жизни. Внутри прямоугольника указываются подчеркнутое имя объекта и имя класса, к которому принадлежит объект.
2. Из объектной модели выбрать те операции, которые участвуют в реализации сценария. Если такие операции не были определены при построении диаграммы программных классов, то необходимо их описать и внести в модель.
3. На диаграмме последовательностей каждому запросу на выполнение операции должна соответствовать горизонтальная линия со стрелкой, начинающаяся от вертикальной линии того пользователя или объекта, который вызывает операцию, и заканчивающаяся на линии жизни того пользователя или объекта, который будет ее выполнять. Над стрелкой указывается номер операции, число итераций, имя операции и в скобках ее параметры. После описания операции может следовать комментарий, поясняющий смысл операции и начинающийся со знака "//".

Операция, которая реализует запрос, на линии жизни объекта обозначается прямоугольником. Порядок выполнения операций определяется ее номером, который указывается перед именем, и положением горизонтальной линии на диаграмме. Чем ниже горизонтальная линия, тем позже выполняется операция. В диаграммах последовательности принято применять вложенную систему нумерации, так как это позволяет отобразить их вложенность. Нумерация операций каждого уровня вложенности должна начинаться с 1.

На диаграмме последовательностей можно описать вызов операции по условию (конструкция if-else) и показать моменты создания и уничтожения объектов. Если объект создается или уничтожается на отрезке времени, представленном на диаграмме, то его линия жизни начинается и заканчивается в соответствующих точках, в противном случае линия жизни объекта проводится от начала до конца диаграммы. Символ объекта рисуется в начале его линии жизни; если объект создается не в начале диаграммы, то сообщение о создании объекта рисуется со стрелкой, проведенной к символу объекта. Если объект уничтожается не в конце диаграммы, то момент его уничтожения помечается большим крестиком "Х".

Диаграмма последовательностей для операции копирования строк полученных в БД в список объектов представлена на рис. 2.9. Диаграмма последовательностей для операции получения данных из БД представлена на рис. 2.10.

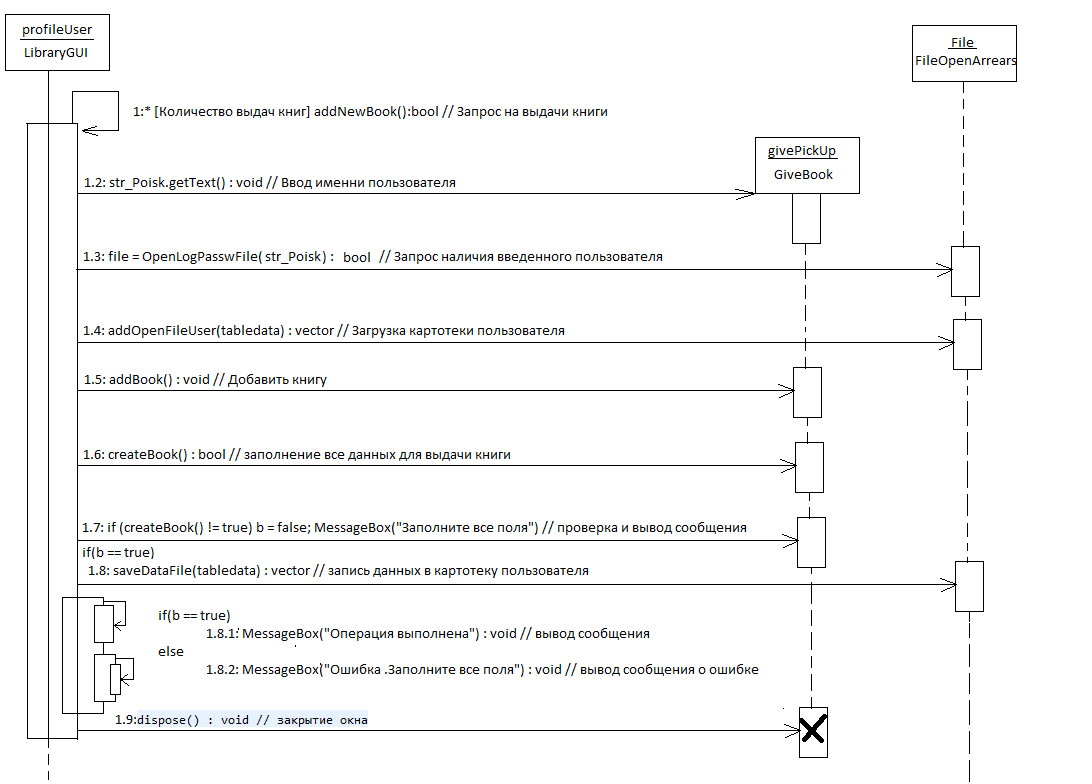
****

Рис.2.9 Диаграмма последовательностей для операции копирования строк из БД в список

2.6 Построение диаграммы действий

Диаграмма действий (activity diagram) строится для сложных операций. Основным направлением использования диаграмм деятельности является визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения. Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются действия, а дугами — переходы от одного действия к другому. Она очень похожа на блок-схемы алгоритмов. Каждая диаграмма деятельности должна иметь единственное начальное и единственное конечное состояние. Диаграмму деятельности принято строить таким образом, чтобы действия следовали сверху вниз. Отличительной чертой диаграммы действий является то, что в ней можно отобразить параллельные процессы. Для этой цели используется специальный символ (линия синхронизации), который позволяет задать разделение и слияние потоков управления. При этом разделение имеет один входящий переход и несколько выходящих, а слияние, наоборот, имеет несколько входящих переходов и один выходящий.

В общем случае действия на диаграмме деятельности выполняются над теми или иными объектами. Эти объекты либо инициируют выполнение действий, либо определяют некоторый результат этих действий. При этом действия специфицируют вызовы, которые передаются от одного объекта графа деятельности к другому. Чтобы связать объекты с действиями, необходимо явно указать их на диаграмме деятельности. Для графического представления объектов, используются прямоугольник, в котором указывается подчеркнутое имя класса. Подчеркнутое имя означает, что на диаграмме задается объект, а не его класс. Далее после имени можно указать в прямых скобках значения атрибутов объекта после выполнения предшествующего действия. Такие прямоугольники объектов присоединяются к переходам отношением зависимости с помощью пунктирной линией со стрелкой.

Диаграмма действий представлена на рис. 2.11.

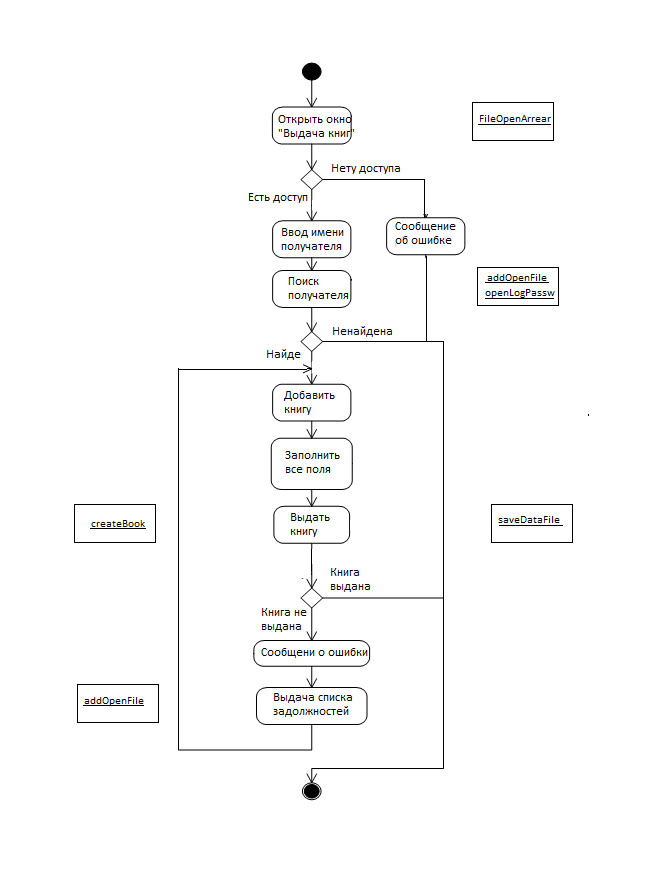


Рис. 2.11 Диаграмма действий

# 3. Руководство оператора

## 3.1 Назначение программы

ПК «Учет Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» должен входить в состав автоматизированной системы учета и администрирования информации, и предназначен для автоматизации деятельности ОЛ, ответственного за учет прихода и расхода лекарств.

В рамках ПК «Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» ОЛ может:

* добавлять, править и удалять информацию о добавление книг;
* добавлять, править и удалять информацию о списывание книг;
* добавлять и удалять информацию в картотеке читателя;
* получать справочную информацию по приходу и выдачи книг;

## 3.2 Условия выполнения программы

ПК предназначен для функционирования под операционной средой Windows (XP, 2x) при поддержке MS Access.

Персональная электронно-вычислительная машина (ПЭВМ) должна обладать следующими характеристиками:

1. тип процессора Pentium III 500 и выше;
2. объем ОЗУ – не менее 128Мб;
3. объем жесткого диска – не менее 4Гб;
4. видеокарта – 64Мб;
5. стандартная клавиатура;
6. манипулятор типа "мышь".

## 3.3 Описание задачи

В ПК должны храниться сведения о картотеках пользователях. Администратор библиотеки может добавлять, изменять и удалять эти сведения. Ему может потребоваться следующая информация:

* права доступа пользователя;
* имеется ли книга в библиотеки;
* до какого числа можно выдать книгу;
* предыдущие задолженности пользователя.

Обязательными требованиями при разработке кода ПК являются использование следующих конструкций языка Java:

* закрытые и открытые члены классов;
* наследование;
* конструкторы с параметрами;
* абстрактные базовые классы;
* виртуальные функции;
* обработка исключительных ситуаций;
* динамическое создание объектов.

С целью выполнения поставленной задачи в процессе проектирования разработана общая модель ПК с выявлением основных объектов и связей между ними. На основании полученной модели разработаны программные классы. Также в процессе проектирование принято решение о создании двух взаимосвязанный информационных объектов, первый из которых предназначен для хранения информации о книгах, второй – для хранения информации о выданных книгах. Данные об информационных объектах хранятся в базе данных.

Требования к коду ПК учтены при создании программных классов и непосредственном написании программы.

## 3.4 Входные и выходные данные

Выходные данные должны быть представлены в виде таблице содержащий описание необходимых информационных объектов, выполненного посредствам представления его характеристик.

Входная информация для задачи «Учет Регистрация, учет, администрирование, редактирования и выдача сведений о книгах» содержится в приходно-расходной документации. Ввод исходных данных должен осуществляется ОЛ в режиме диалога. Вводимые данные являются значениями характеристик (атрибутов) информационных объектов. Вводимая информация может выбираться или набираться из списка предлагаемых значений.

## 3.5 Выполнение программы

### 3.5.1 Подготовка к запуску (осуществляется один раз после установки ПК на ЭВМ)

В источники данных (ODBC) добавить драйвер баз данных MS Access с именем MicrosoftBases. В качестве базы данных указать полный путь к БД ПК dbase.mdb.

### 3.5.2. Запуск программы

При запуски программы на экране появится диалоговое окно представленное на рис. 3.1.

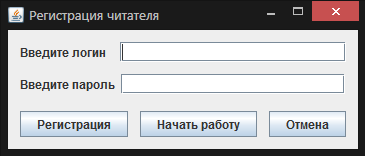


Рис. 3.1 Главное диалоговое окно

### 3.5.3. Выполнение основных функций

*3.5.3.1 Регистрация и вход в бибилотеку*

На главном диалоговом окне, рис. 3.1, нажать кнопку «Регистрация». При этом появится диалоговое окно регистрации – рис. 3.2

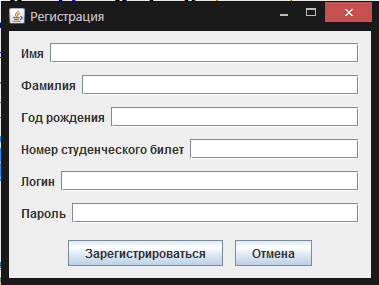


Рис. 3.2 Диалоговое окно регистрации

Заполнив все поля и нажав на кнопку «Зарегистрироваться», пользователю будет выведено сообщение, представленное на рис. 3.3, о удачной регистраций.

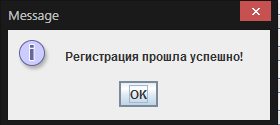


Рис. 3.4 Сообщение об удачной регистрации

После диалоговое окно регистрации закроется, далее выведется диалоговое окно регистрации читателя, в котором нужно будет ввести свои логин и пароль и нажать на кнопку «Начать работу», после чего откроется диалоговое окно библиотека представленное на рис. 3.4.

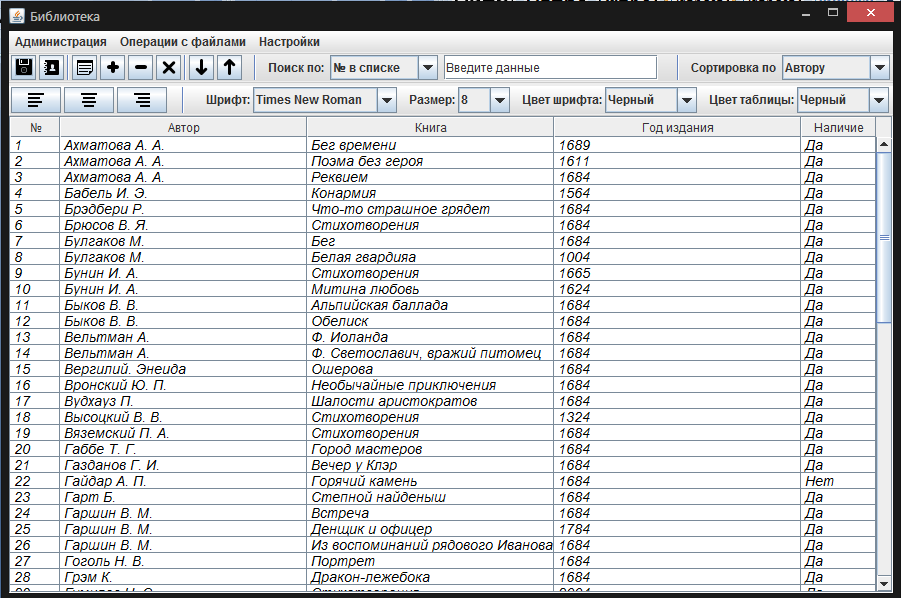


Рис. 3.4 Диалоговое окно библиотека

*3.5.3.2 Ввод информации*

При открытие диалогового окна библиотека, сразу автоматически загружаются все данные в таблицу из БД, что можно видеть на рис. 3.4.

*3.5.3.3 Использование библиотеки от лица администратора*

Администратор может выполнять любые действия в диалоговом окне библиотека, а так же в других диалоговых окнах.

*3.5.3.3.1 Использования меню Администрация*

При нажатий на пункт меню «Администрация», администратор может выбрать один из пунктов подменю. Список подменю «Администрация», представлен на рис. 3.5.

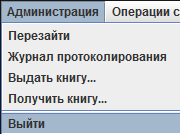


Рис. 3.5 Пункт меню Администрация

Если нажать перезайти, то пользователю будет выведено диалоговое окно, представленное на рис. 3.1. При нажатий на журнал протоколирования, будет выведено диалоговое окно с протоколом всех действий в программе, данное диалоговое окно представлено на рис. 3.6.

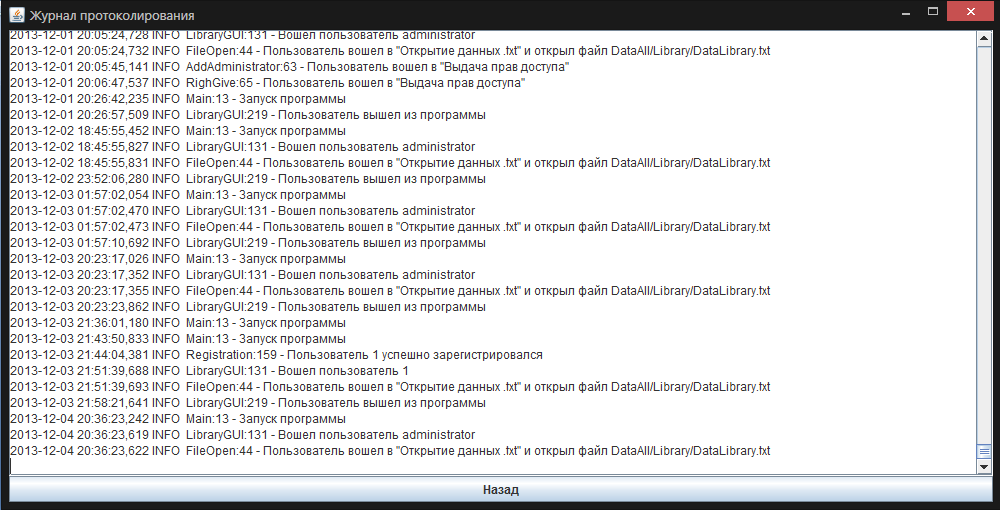


Рис. 3.6 Диалоговое окно «Журнал протоколирования»

При нажатий на подпункт меню выдать книги, выведется диалоговое окно, представленное на рис.3.7, в котором нужно будет ввести имя пользователя, если пользователь будет найден, то будет выведено сообщение, представленное на рис.3.8.

*3.5.3.3.1.1 Выдача книг пользователям*

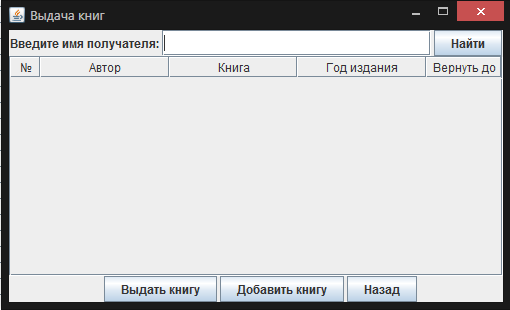


Рис. 3.7 Диалоговое окно «Выдача книг»

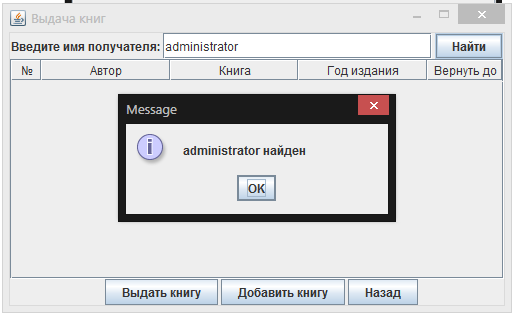


Рис. 3.8 Сообщение о найденном пользователе

Если у пользователя нет долгов, то будет выведено сообщение, представленное на рис. 3.9.

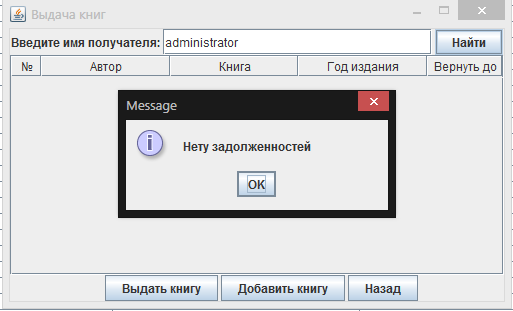


Рис. 3.9 Сообщение о ненайденных задолженностях

В противном случае, будет выведены все книги, который пользователь должен вернуть библиотеке, данные представлены на рис. 3.10.

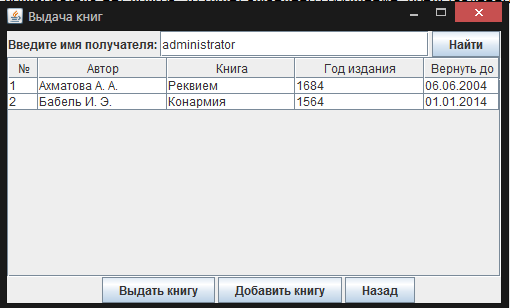


Рис. 3.10 Диалоговое окно «Выдача книг», с выданными книгами пользователю

Чтобы выдать книгу пользователю, нужно проделать следующие действия: нажать на кнопку «Добавить книгу» (будет добавлена новая строка в таблице), после заполнить все поля, нажать на кнопку добавить, в результате вкартотеку пользователя будет добавлена книга и будет выведено сообщение. Все операция представлены на рис. 3.11а, рис. 3.11б.

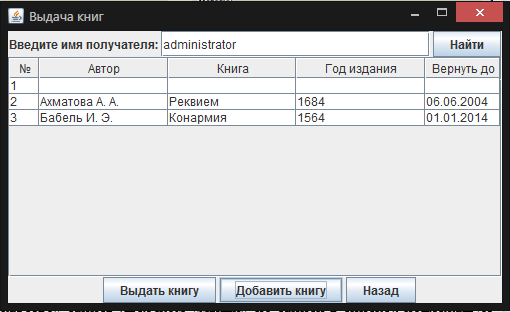


Рис. 3.11а Диалоговое окно «Выдача книг», после нажатия на кнопку «Добавить книгу»

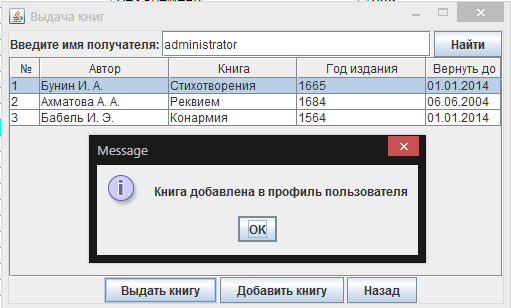


Рис. 3.11б Диалоговое окно «Выдача книг», после заполнения всех полей и нажатия на кнопку «Выдать книгу»

Если же не заполнить все поля или ввести не существующие данные и нажать на кнопу «Выдать книгу», то выведутся сообщение об ошибке представленные на рис. 3.12.

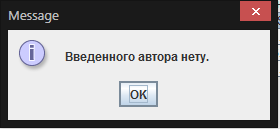
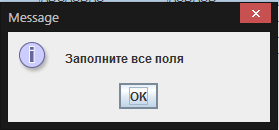


Рис.3.12 Сообщения об ошибках.

*3.5.3.3.1.2 Списывание книги у пользователя*

*При нажатие на подпункт меню «Получить книгу», выведется диалоговое окно приведенное на рис. 3.13.*

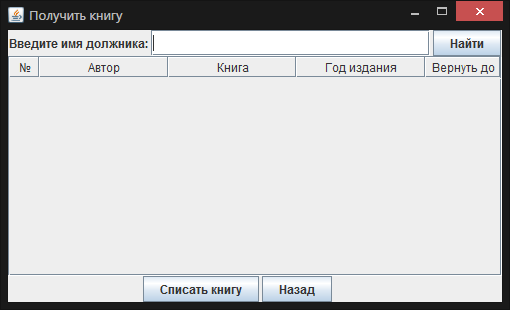


Рис. 3.13 Диалоговое окно «Получить книг»

Поиск пользователя и вывод сообщений аналогичны, что и в подпункте меню «Выдача книг». Для того, что бы списать книгу у пользователя с картотеки, нужно выбрать нужную книгу, и нажать на кнопку «Списать книгу», после чего книгу будет удаленна с картотеки пользователя.

*3.5.3.3.2 Использования меню Операции с файлами*

При нажатий на пункт меню «*Операции с файлами*», администратор может выбрать один из пунктов подменю. Список подменю «*Операции с файлами*», представлен на рис. 3.14.

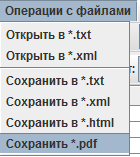


Рис. 3.14 Пункт меню *Операции с файлами*

*3.5.3.3.2.1 Открытие файла одного из формата*

При нажатий на подпункт меню «Открыть в \*.txt», будет выведено диалоговое окно, представленное на рис.15, в котором нужно будет выбрать файл .txt формата.

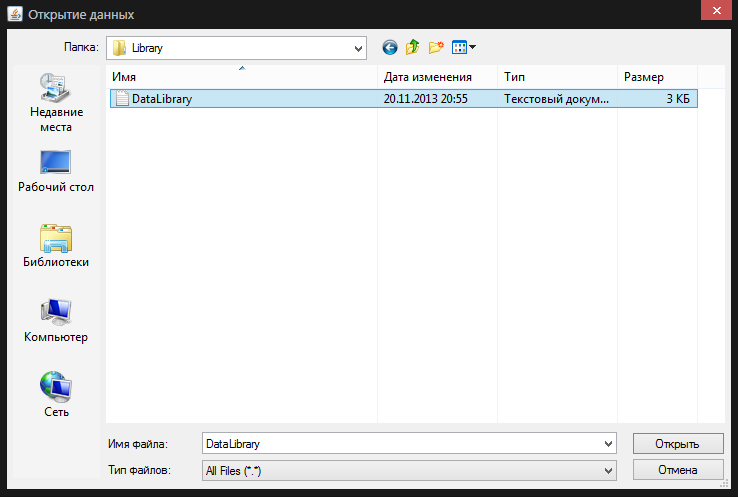


Рис. 3.15 Диалоговое окно «Открытие данных»

При открытие, например DataLibrary документа, все данные загрузятся в таблицу главного окна «Библиотека», которая представлена на рис. 3.16.

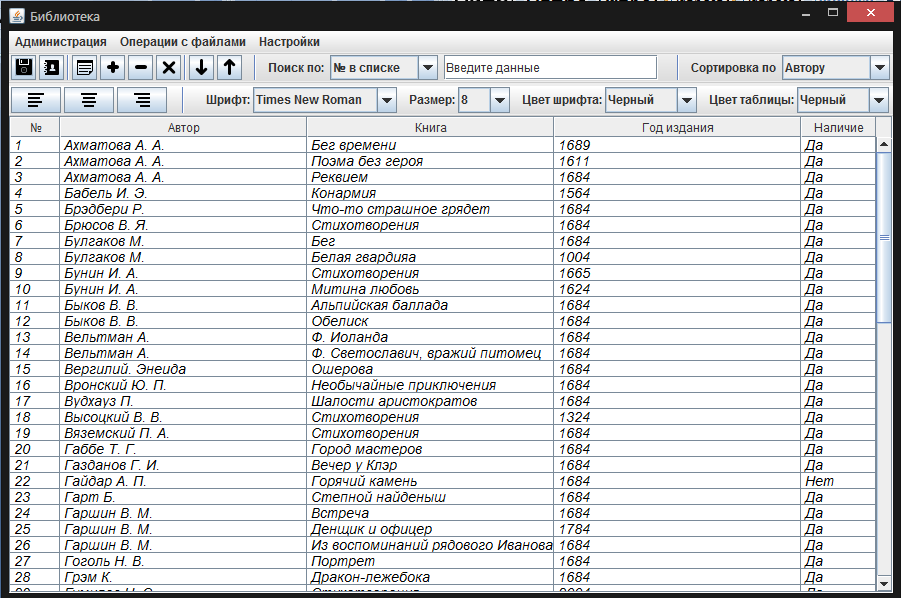


Рис. 3.16 Диалоговое окно библиотека

Если при открытие файла, было введено не существующие имя файла, то будет выведена следующие сообщение, представленная на рис. 3.17.

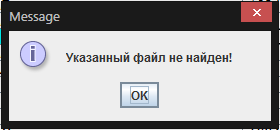


Рис. 3.17 Сообщение об ненайденном файле.

Примечание! Для открытия файлов других форматов, действие будут аналогичны.

*3.5.3.3.2.2 Сохранение файла одного из формата*

При нажатий на подпункт меню «Сохранить в \*.txt», будет выведено диалоговое окно, представленное на рис.3.18, в котором нужно будет ввести имя файл .txt формата.

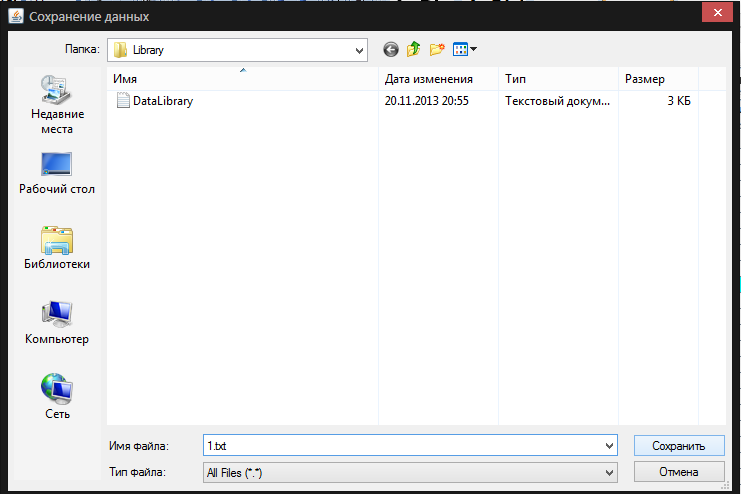


Рис. 3.18 Диалоговое окно «Сохранение данных»

Сохраним все данные в файле с название 1.txt и откроем данный файл, чтобы убедиться в правильности работы, файл 1.txt представлен, на рис. 3.19.

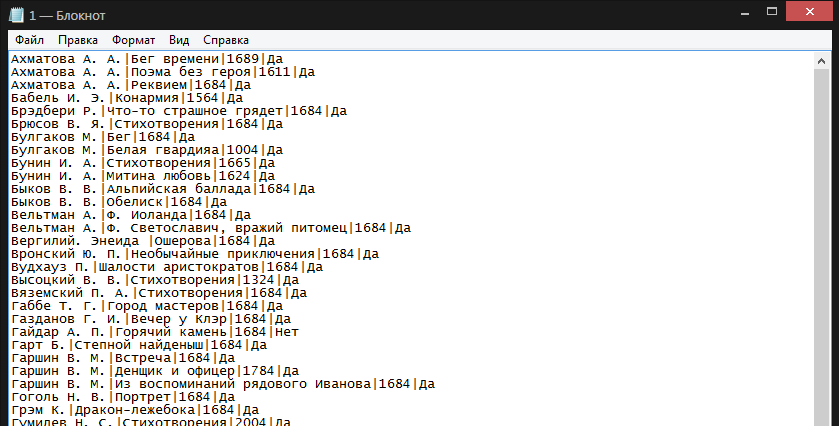


Рис. 3.19 Файл 1.txt

Примечание! Для сохранения файлов других форматов, действие будут аналогичны.

*3.5.3.3.3 Использования меню Настройки*

При нажатий на пункт меню «*Настройки*», администратор может выбрать один из пунктов подменю. Список подменю «*Настройки*», представлен на рис. 3.20.

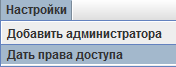


Рис. 3.20 Пункт меню *Настройки*

Если нажать на подпункт меню «Добавить администратора», то будет выведено окно представленное на рис. 3.21.

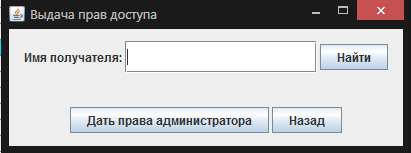


Рис. 3.21 Диалоговое окно «Выдача прав доступа»

В данном диалоговом окне можно выдать права доступа любому пользователю, ввод и поиск пользователя аналогична, что в диалоговом окне «Выдача книг». После того как пользователь будет найден, нужно будет нажать на кнопку «Дать права администратора», и данный пользователь будет добавлен в список администраторов.

Если нажать на подпункт меню «Добавить администратора», то будет выведено окно представленное на рис. 3.22.

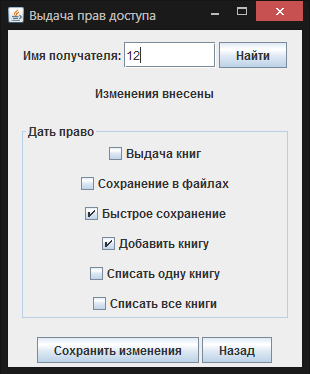


Рис. 3.22 Диалоговое окно «Выдача прав доступа»

В данном диалоговом окне можно выдать права доступа любому пользователю, ввод и поиск пользователя аналогична, что в диалоговом окне «Выдача книг». Если у данного пользователя, есть уже права, то они будут показаны галочками, чтобы снять или добавить права, нужно снять или добавить галочки, после нажать на кнопку «Сохранить изменения».

*3.5.3.3.4 Добавление, удаление книг из библиотеки*

Чтобы добавить книгу в библиотеку, нужно нажать в диалоговом окне библиотека на кнопку «Добавить книгу», после чего будет выведена новая строку(первая), после чего администратор вписывает все нужные поля, аналогично как и в диалоговом окне «Выдать книгу»(для того, чтобы добавить новую книгу, нужно чтобы все поля были заполнены , иначе программа проигнорирует нажатие на кнопку «Добавить книгу»).

Что бы удалить книгу, администратор должен нажать на нужную ему книгу и нажать на кнопку «Удалить книгу», действие аналогично действие в диалоговом окне «Забрать книгу».

Для того, чтобы удалить все книги с библиотеки, нужно нажать кнопку «Удалить все».

*3.5.3.3.5 Получить дополнительную информацию о книге*

Чтобы получить дополнительную информацию о книгах, администратор должен сначала нажать в диалоговом окне «Библиотека» на кнопку «Получить информацию», после чего нажать на нужную книгу(при нажатий на кнопку она будет переведена в состояние true(в этот период кнопка будет выделена)).Если информация о книге будет, то будет выведено переделанное диалоговое окно «Библиотека» представленное на рис. 3.23, если же информация будет отсутствовать, то программа проигнорирует данное действие.

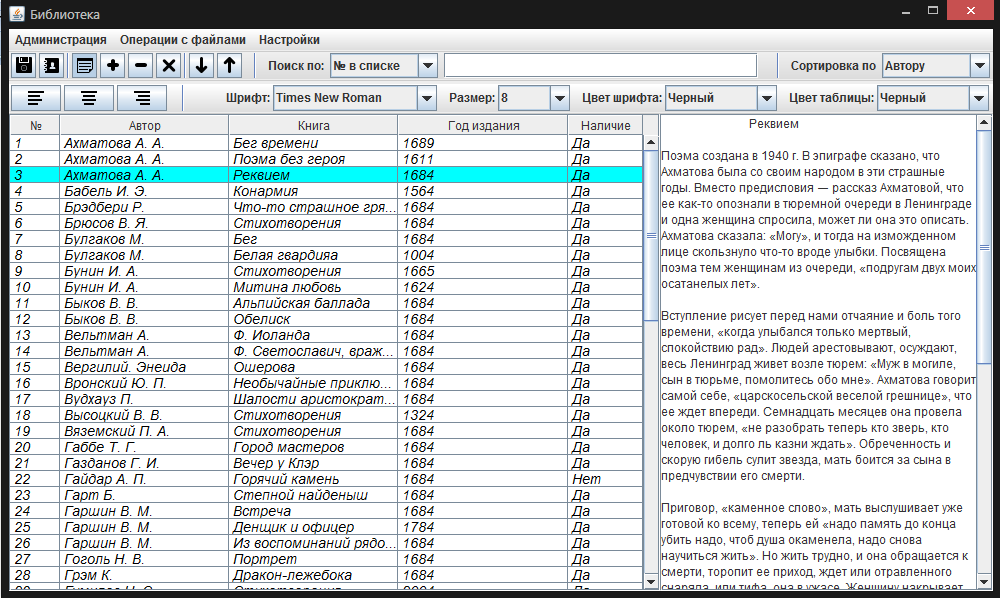


Рис. 3.23 Диалоговое окно «Библиотека», с информацией о книге

Для того, чтобы отключить режим выдачи информаций о книгах, нужно ещё раз нажать на кнопку «Получить информацию».

*3.5.3.3.6 Изменение данные в библиотеке*

Для того, чтобы изменить данные в библиотеке, нужно двойным нажатием левой кнопки мыши ну нужную ячейку, после чего внести изменения, чтобы данные были сохранены в БД нужна нажать на кнопку «Быстрое сохранение».

*3.5.3.3.6 Поиск и сортировка данных в библиотеке*

Чтобы отсортировать в прямом и обратном порядке любые из данных, нужно выбрать в поле «Сортировка по» и выбрать из списка нужный вам параметр. На примере выберем пункт «Книге», в результате все книги будут все книги расположены в алфавитном порядке, что показано на рис 3.24.

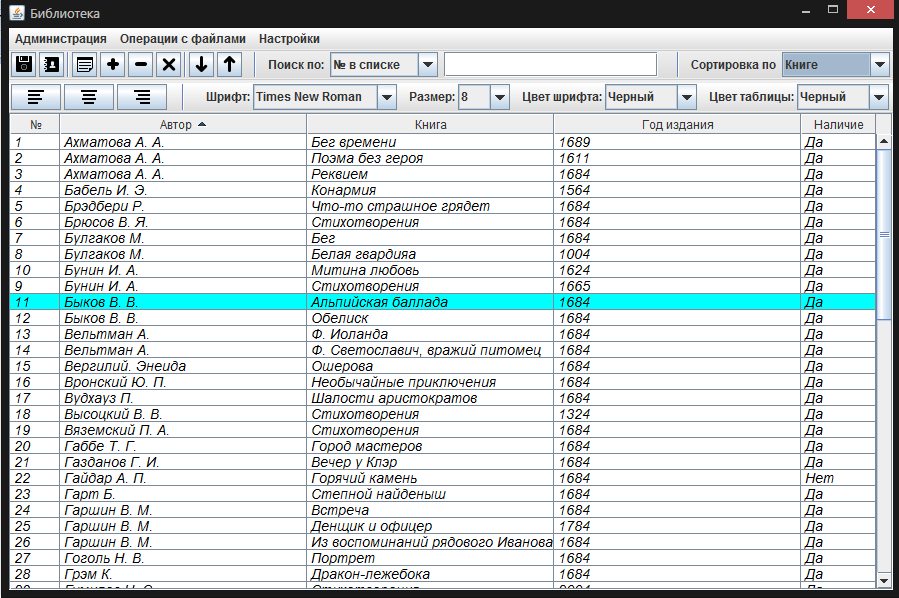


Рис. 3.24 Диалоговое окно «Библиотека» с сортированными книгами по алфавиту

Для поиска данных, нужно выбрать элементы, по которому будет вестись поиск, после вводить данные в строке «Поиск по:». На примере выберем пункт «Год издание» и введём 1324, в результате будет выделена строка, пример показан на рис.3.25. Если данные не будет найдены, то программа проигнорирует ваши действия.

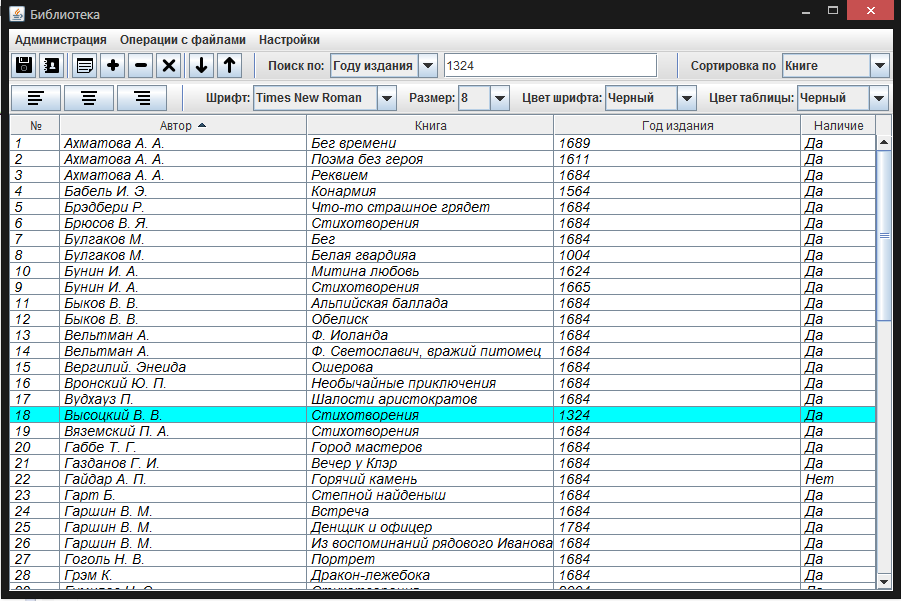


Рис. 3.25 Диалоговое окно «Библиотека» с найденной датой

*3.5.3.3.7 Изменение интерфейса программы*

Для того, чтобы сделать скроллинг верх или вниз таблицы все данных библиотеки, нужно нажать на кнопки «Скроллинг верх» и «Скроллинг вниз».

Чтобы выровнять данные по левому, правому или центу, нужно нажать на кнопки «Текст с лева», «Текст по центру» и «Текст с права».

Чтобы изменить шрифт текста, нужно в списке «Шрифт» выбрать нужный шрифт. Для изменения размера шрифта, нужно в списке «Размер» выбрать нужный размер шрифта. Для выбора цвета текста и цвета таблицы, действия аналогичны действия выбора размера шрифта.

Пример с измененным размером, шрифтом и местом расположения текста и изменение его цвета, приведено на рис 3.26.

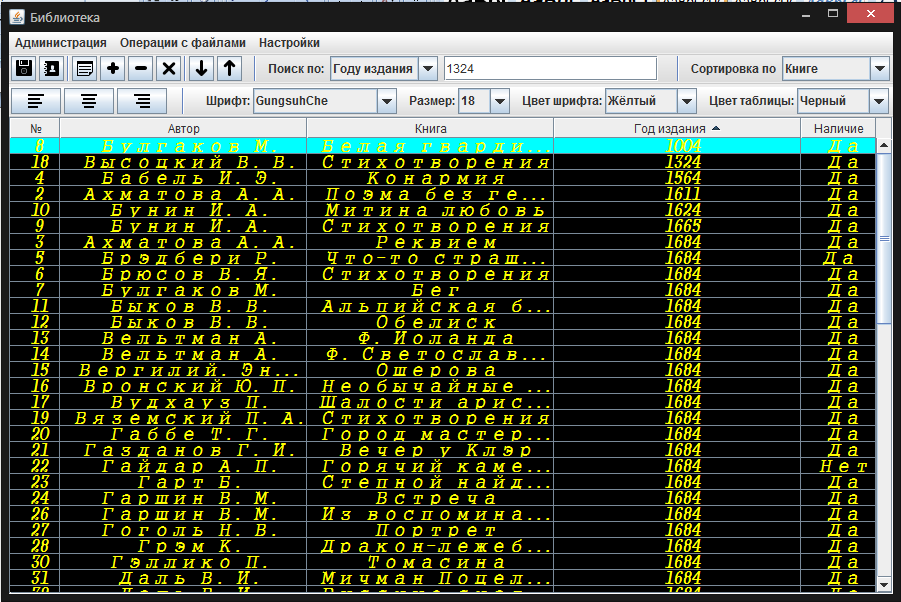


Рис. 3.26 Диалоговое окно «Библиотека» и измененным интерфейсом

# Исходные тексты ПК

# Заключение

В результате проделанной работы разработан ПК «Регистрация, учет, администрирование, редактирования и выдача сведений о книгах», предназначенный для администрирования и учета информации по списание и выдачи книг, разработано руководство оператора.

В процессе проектирования созданы описание вариантов использования ПК, прототип интерфейса пользователя, объектная модель ПК, диаграмма классов, описание поведения ПК, диаграмма действия ПК.

Курсовой проект удовлетворяет поставленным требованиям.