

Министерство образования и науки Российской Федерации  
**Муромский институт (филиал)**  
федерального государственного бюджетного образовательного учреждения высшего образования  
**«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»**  
(МИ ВлГУ)

Факультет информационных технологий  
Кафедра информационных систем

# КУРСОВАЯ РАБОТА

по курсу Прикладная разработка на Java  
на тему: Программа учета книг в библиотеке

Руководитель

К. Т. Н., доц. каф. ИС  
(уч. степень, звание)

Метелкин А. С.  
(фамилия, инициалы)

(подпись) (дата)

Студент ИС-122  
(группа)

Десятникова Н.Д.  
(фамилия, инициалы)

(подпись) (дата)

Члены комиссии

(подпись) (Ф.И.О.)

(подпись) (Ф.И.О.)

В курсовой работе разработана программа учета книг в библиотеках, обеспечивающей управление данными о книгах, включая добавление новых книг, удаление списываемых книг, а также выдачу сведений о книгах с возможностью упорядочивания по фамилиям авторов или годам издания.

Табл. 1. Ил. 2. Библ. 4.

## Содержание

ВВЕДЕНИЕ .....	5
1. АНАЛИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ .....	7
1.1 Описание предметной области .....	7
1.2 Определение требований.....	7
1.3 Анализ программ-аналогов .....	8
1.4 Выбор технологии разработки приложения.....	8
2. ПРОЕКТИРОВАНИЕ ПРОГРАММЫ .....	10
2.1 Работа программы.....	10
2.2 Главные экраны.....	10
2.3 Разработка базы данны .....	10
2.4 Основные классы программы .....	11
3. РАЗРАБОТКА ПРОГРАММЫ .....	12
3.1 Архитектура программы .....	12
3.2 Этапы разработки.....	12
3.3 Используемые технологии .....	14
3.4 Алгоритм работы программы .....	14
4. ТЕСТИРОВАНИЕ .....	16
4.1 Автоматическое тестирование.....	16
4.2 Ручное тестирование.....	<b>Ошибка! Закладка не определена.</b>
4.3 Результаты тестирования .....	16
ЗАКЛЮЧЕНИЕ .....	18
СПИСОК ЛИТЕРАТУРЫ.....	19

					МИВУ 09.03.02 - 00.000 ПЗ			
Изм.	Лист	№ докум.	Подп.	Дата				
Разраб.	Десятникова Н.Д..				Программа учета книг в библиотеке		Лит.	Лист
Пров.	Метелкин А. С.						у	Листов
							4	18
Н. контр.							МИ ВлГУ	
Утв.							ИС-115	

## Введение

В современных библиотеках точный учет книг и управление библиотечным фондом являются важными задачами, требующими автоматизации. Ручной учет книг может привести к потере. Разработка программного обеспечения для автоматизации учета книг позволяет повысить эффективность работы библиотекарей и обеспечить удобство для пользователей.

Тема данной курсовой работы — разработка программы учета книг в библиотеке.

Актуальность этой темы обусловлена необходимостью упрощения процессов управления библиотечным фондом, сокращения ошибок при учете книг, таких как добавление, удаление или поиск книг.

Целью работы является разработка программы для эффективного и точного учета книг в библиотеке, а также совершенствование навыков программирования на языке Java с использованием технологий, таких как графический интерфейс Swing и база данных РедБазаДанных.

Проанализировав цель работы, можно выделить следующие задачи:

1. Изучение существующих систем учета книг в библиотеках.
2. Определение подходящей структуры базы данных для хранения информации о книгах, включая автора, название, год издания и количество экземпляров.
3. Разработка алгоритмов для добавления, удаления и сортировки книг, а также поиска по автору или названию.
4. Реализация управления количеством экземпляров книг, включая добавление и удаление экземпляров.
5. Создание удобного графического пользовательского интерфейса для взаимодействия с программой.
6. Тестирование программы для обеспечения ее надежности и корректности работы.

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		5

Результатом данной курсовой работы будет приложение, в котором будут реализованы все поставленные задачи.

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		6

## 1. Анализ технического задания

### 1.1 Описание предметной области

Программа учета книг в библиотеке для автоматизации процессов управления библиотечным фондом, включая хранение, добавление, изъятие, сортировку и поиск книг. Она обеспечивает учет данных о книгах (автор, название, год издания, количество экземпляров) и поддерживает работу библиотеки, например, регистрацию новых поступлений, составление книг, поиск по автору или названию и упорядочивание фонда.

Основные задачи системы:

1. Автоматизация учета библиотечного фонда:
  - Централизованное хранение информации о книгах
  - Точный контроль количества экземпляров каждой книги
2. Оптимизация рабочих процессов:
  - Устранение дублирования данных
  - Минимизация рутинных ручных операций
3. Обеспечение удобства работы:
  - Быстрый доступ к актуальной информации о книгах
  - Удобные инструменты поиска и сортировки

### 1.2 Определение требований

Программа реализует следующие алгоритмы для управления библиотечным фондом:

1. Добавление книги: Проверяет наличие книги по автору и названию. Если книга существует, увеличивает количество экземпляров; если нет — создает новую запись в базе данных.

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подп.	Дата		

2. Удаление книги: Позволяет удалить все экземпляры книги (обнуление) или уменьшить их количество. Использует SQL-запросы для обновления базы данных.

3. Сортировка книг: Выполняет сортировку по фамилиям авторов или годам издания с использованием SQL-запросов.

4. Поиск книг: Реализует поиск по частичному совпадению автора или названия с использованием SQL-запроса.

Эти алгоритмы обеспечивают выполнение требований технического задания и добавляют дополнительную функциональность поиска. Использование SQL-запросов повышает производительность при работе с большими объемами данных.

### 1.3 Анализ программ-аналогов

Для анализа были рассмотрены две популярные системы учета книг:

Koha: Открытая библиотечная система с веб-интерфейсом и базой данных MySQL.

- Преимущества: Поддержка больших библиотек, управление выдачей книг, веб-доступ.
- Недостатки: Сложность настройки, избыточная функциональность для малых библиотек.

OpenBiblio: Открытая система для малых библиотек с веб-интерфейсом и MySQL.

- Преимущества: Простота, подходит для небольших библиотек.
- Недостатки: Ограниченные возможности поиска и устаревший интерфейс.

### 1.4 Выбор технологии разработки приложения

1. Язык программирования для разработки приложения выбран Java 17. Он обеспечивает кроссплатформенность, запуская программы в разных операционных

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		8

средах без изменений. Кроме того, Java предоставляет среду Swing для создания графического интерфейса

2. Графический интерфейс Swing для обеспечения кроссплатформенности.

3. База данных РедБазаДанных для хранения информации о книгах. Выбрана из-за поддержки SQL, РедБазаДанных поддерживает стандарт SQL, что упрощает проектирование и взаимодействие с базой данных через JDBC-драйвер для Java. Также из-за кроссплатформенности, ведь РедБазаДанных поддерживает работу на Windows, Linux, что соответствует требованиям кроссплатформенности приложения.

4. В качестве средств разработки использовалась IntelliJ IDEA. В качестве интерфейса выбрана IDE.

5. Среда сборки для управления зависимостями и сборки проекта использовалась Maven. Он позволяет легко подключать библиотеки, такие как JDBC-драйвер для РедБазаДанных, через файл pom.xml. Она автоматизирует процесс сборки, компиляции и запуска приложений, что снижает риски ошибок и экономит время при разработке.

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		9



## 2. Проектирование программы

### 2.1 Работа программы

Программа будет состоять из 4 основных частей:

1. Окно с таблицей книг - показывает все книги в библиотеке
2. Кнопки управления - добавление, удаление, поиск книг
3. Система хранения данных - база данных Firebird
4. Логика программы - обработка всех действий

### 2.2 Главные экраны

#### 1. Основное окно:

- Таблица со списком книг (автор, название, год, количество)
- Кнопки: "Добавить", "Удалить", "Найти", "Сортировать"
- Поле для поиска

#### 2. Окно добавления книги:

- Поля для ввода: Автор, Название, Год, Количество
- Кнопки: "Сохранить", "Отмена"

#### 3. Окно поиска:

- Поле для ввода текста поиска

### 2.3 Разработка базы данны

Основным способом при создании базы данных была разработка простой структуры для хранения и обработки данных в библиотечном фонде. Для этого использовался инструмент RedExpert, с помощью которого создана ER-диаграмма. Она показывает сущности, их атрибуты и связи между ними, помогая понять, как организованы данные.

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		10

ER-диаграмма включает одну таблицу — BOOKS, которая хранит информацию о книгах (автор, название, год издания, количество экземпляров) и обеспечении операций: добавление, удаление, поиск и сортировка.

Первоначально была создана таблица BOOKS, которая хранит данные книг в библиотеке:

BOOKS (Книги):

- Хранить информацию о книгах в библиотечном фонде. Благодаря этой табличной системе можно управлять библиотечным фондом, поддерживая операции добавления, удаления, определения и сортировки книг.

BOOKS		
ID	BIGINT	(PK)
AUTHOR	VARCHAR(255)	
TITLE	VARCHAR(255)	
PUBLICATION_YEAR	INTEGER	
COPIES	INTEGER	

Рисунок 1 – ER-диаграмма

## 2.4 Основные классы программы

1. Класс Book - хранит информацию об одной книге;
2. Класс Database - работает с базой данных;
3. Класс MainWindow - главное окно программы:

### 3. Разработка программы

#### 3.1 Архитектура программы

Приложение построено по многослойной архитектуре, которая разделяет функции уровня: представление, бизнес-логику, доступ к данным и модели данных. Такая структура структурной адаптации, поддержки и расширения системы. К основным компонентам системы относятся:

Слой представления (UI):

Класс LibraryApp (графический интерфейс): Отвечает за взаимодействие с пользователем. Реализовано выполнение с использованием Java Swing, включает элементы управления (кнопки, поля ввода, область вывода) для таких операций, как добавление, удаление, поиск и сортировка книг.

Слой бизнес-логики (Услуги):

Класс BookService (обработка операций): Управление обеспечивает основную логику приложения. Выполняет операции добавления, удаления, поиска и сортировки книг, взаимодействующих с другими слоями, чтобы обеспечить корректную обработку данных.

Слой доступа к данным (Хранилище):

Класс BookStorage (взаимодействие с базой данных): Отвечает за работу с базой данных РедБазаДанных через JDBC. Выполняет SQL-запросы для хранения, обновления и извлечения данных книги из таблицы BOOKS.

Слой модели данных (Model):

Класс Book (хранение информации о книге): представляет модель данных для книг. Содержит поля для хранения информации: идентификатор, автор, название, год издания и количество экземпляров, обеспечивая единый формат данных для всех слоев.

#### 3.2 Этапы разработки

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подп.	Дата		

Разработка программы шла в несколько этапов, что привело к созданию функционального приложения:

1. Анализ требований и проектирование:

1.1. На основе технического задания старая ключевая функция: добавление книг, удаление книг, поиск по автору или названию и сортировка по автору или году издания.

1.2. Спроектирована структура базы данных с таблицей BOOKS для хранения данных о книгах.

2. Настройка окружения:

2.1. Настроен проект в среде разработки IntelliJ IDEA. Добавлены зависимости, такие как JDBC-драйвер для РедБазаДанных, для работы с базой данных.

2.2. Подготовлены настройки подключения к базе данных в коде.

2.3. Установлена и настроена база данных РедБазаДанных с созданием файла базы данных kursach\_jv.fdb

3. Реализация модулей:

3.1. Разработан класс LibraryApp, создающий графический интерфейс с использованием Java Swing. Реализованы элементы управления: кнопки для добавления, удаления, поиска и сортировки, а также поле ввода и область отображения списка книг.

3.2. Реализован слой бизнес-логики в классе BookService, который обрабатывает операции добавления, удаления, поиска и сортировки книг, включая проверку данных и взаимодействие с базой.

3.3. Настроено взаимодействие с РедБазаДанных через JDBC в классе BookStorage. Реализованы SQL-запросы для работы с таблицей BOOKS, включая добавление, обновление, поиск и сортировку записей.

3.4. Добавлен класс Book для представления модели данных, содержащий поля для хранения информации о книгах (ID, автор, название, год издания, количество экземпляров).

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подп.	Дата		

### 3.3 Используемые технологии

Java 17: Обеспечивает стабильность, производительность и поддержку современных возможностей языка. Использован для разработки всех слоев приложения, включая Swing для графического интерфейса.

IntelliJ IDEA: Упрощает разработку, отладку и тестирование кода благодаря удобным инструментам автодополнения и интеграции с Java.

РедБазаДанных: Хранит данные о книгах с высокой надежностью. Поддерживает сложные SQL-запросы, такие как поиск и сортировка, и легко интегрируется с Java через JDBC.

Разработка велась с учетом требований кроссплатформенности и удобства использования, что позволило создать функциональное приложение, готовое к эксплуатации в библиотеке.

### 3.4 Алгоритм работы программы

Программа работает по следующему алгоритму:

#### 1. Инициализация:

- Запускается класс Main, создающий объект LibraryApp для инициализации интерфейса.
- Устанавливается соединение с базой данных РедБазаДанных через JDBC.

#### 2. Отрисовка интерфейса:

- Отображаются компоненты Swing: поле поиска, кнопки, текстовая область.

#### 3. Взаимодействие с пользователем:

- Пользователь выполняет действия (поиск, добавление, удаление, сортировка) через интерфейс.
- События обрабатываются в LibraryApp, который вызывает методы BookService.

#### 4. Обработка данных:

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подп.	Дата		

- BookService проверяет данные и передает запросы в BookStorage.
- BookStorage выполняет SQL-запросы для обновления базы данных.

5. Отображение результатов:

- Результаты операций отображаются в текстовой области или диалоговых окнах.

6. Завершение:

- При закрытии программы соединение с базой данных закрывается.

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		15

## 4. Тестирование

Тестирование системы учета книг в библиотеке проводилось для проверки корректности работы всех функций, надежности приложения и удобства использования. Процесс включал автоматическое и ручное тестирование, а также анализ результатов для устранения ошибок.

### 4.1 Автоматическое тестирование

Автоматическое тестирование проводилось с использованием фреймворка JUnit для проверки ключевых компонентов системы. Были созданы тестовые случаи для слоев бизнес-логики (BookService) и доступа к данным (BookStorage).

– Тестирование добавления книги: Проверялось, что метод BookService.addBook корректно добавляет новую книгу в базу данных. Тест создавал книгу с заданными параметрами (автор, название, год, количество экземпляров) и проверял, что запись появляется в таблице BOOKS с правильными данными.

– Тестирование удаления книги: Проверялось, что метод BookService.deleteBook уменьшает количество экземпляров или полностью удаляет книгу (устанавливает количество в 0). Тест подтверждал изменение записи в базе данных.

– Тестирование поиска: Проверялось, что метод BookService.findBooks возвращает книги, соответствующие запросу (например, по автору или названию), и правильно обрабатывает случаи, когда книг нет.

### 4.3 Результаты тестирования

Результаты тестирования показали, что система работает стабильно и выполняет все заявленные функции:

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подп.	Дата		

Таблица 1 – Тестирование программы

№	Сценарий тестирования	Ожидаемый результат	Фактический результат	Статус
1	Добавление книги (автор: "Пушкин А.С.", название: "Евгений Онегин", год: 1833, базу данных экземпляры: 10)	Книга добавлена в базу данных	Книга отображается в списке	Успех
2	Увеличение экземпляров (добавить 5 экземпляров)	Количество экземпляров увеличено до 15	Количество обновлено	Успех
3	Удаление книги (обнуление экземпляров)	Количество экземпляров равно 0	Книга удалена из списка	Успех
4	Поиск по автору ("Пушкин")	Отображаются книги с автором "Пушкин"	Найдена книга "Евгений Онегин"	Успех
5	Сортировка по году	Книги упорядочены по году издания	Список упорядочен корректно	Успех
6	Ввод некорректных данных (пустой автор)	Ошибка сообщением	Сообщение об ошибке	Успех



## Заключение

В ходе выполнения курсового проекта была разработана программа учета книг в библиотеке, соответствующая требованиям технического задания. Программа поддерживает добавление, удаление, сортировку и поиск книг, используя РЕД Базы Данных и Swing. Использование базы данных вместо контейнерного класса повысило надежность и масштабируемость решения.

Цели проекта достигнуты: создана функциональная программа, а навыки программирования на Java улучшены. В дальнейшем возможно добавление функций экспорта данных или поддержки нескольких библиотек.

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		18

## Список литературы

1. Oracle. Java SE 17 Documentation [Электронный ресурс] // Oracle. — URL: <https://docs.oracle.com/en/java/javase/17/> (дата обращения: 05.05.2025).  
*Официальная документация Java 17, содержащая описание языка, стандартных библиотек и инструментов для разработки.*
2. РедБазаДанных Foundation. РедБазаДанных 3.0 Documentation [Электронный ресурс] // РедБазаДанных. — URL: <https://www.РедБазаДанныхsql.org/en/documentation/> (дата обращения: 05.05.2025).  
*Официальная документация базы данных РедБазаДанных, включающая описание SQL, JDBC-драйвера и администрирования.*
3. Ред База Данных. Версия 5.0. Руководство по SQL (Руководство пользователя).
4. Документация по Java Swing:  
<https://docs.oracle.com/javase/tutorial/uiswing/>.

					МИВУ 09.03.02 – 00.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		19