Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

Допускаю к защите

|  |  |
| --- | --- |
| Руководитель |  |
|  | подпись |
|  | В. Л. Аршинский |
|  | И.О. Фамилия |

Разработка приложения для ведения базы коллекции стендовых моделей

|  |
| --- |
| наименование темы |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

|  |
| --- |
| Объектно-ориентированное программирование |

|  |
| --- |
| 1.032.00.00 - ПЗ |
| обозначение документа |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | ИСТб-21-2 |  |  |  | | Н. В. Зимин |
|  |  | шифр |  | подпись |  | | И.О. Фамилия |
| Нормоконтроль |  |  |  |  |  | | В. Л. Аршинский |
|  |  |  |  | подпись |  | | И.О. Фамилия |
| Курсовой проект защищен с оценкой | | | |  | |  | |

Иркутск 2023 г.

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ**

**НА КУРСОВОЙ ПРОЕКТ**

|  |  |  |
| --- | --- | --- |
| По курсу | Объектно-ориентированное программирование | |
| Студенту | Зимину Н. В. | |
|  | (фамилия, инициалы) | |
| Тема проекта: | Разработка приложения для ведения базы коллекций | | |
| стендовых моделей | | | |
| Исходные данные: | | Разработать компьютерное приложение, реализующее | |
| функции чтения и редактирования баз данных | | | |
|  | | | |
|  | | | |
| Рекомендуемая литература: | | | |
| 1. Фримен Э. Head First. Паттерны проектирования. Обновленное юбилейное издание. – СПб.: Питер, 2018. –656 с. 2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е изд.: Пер. с англ. — М.: ООО «И.Д. Вильямс», 2008 – 720 с. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Дата выдачи задания |  | 1 марта 2023 г. | |
| Задание получил студент | | |  | |  | Зимин Н. В. |
|  | | | подпись | |  | Фамилия И.О. |

|  |  |
| --- | --- |
| Дата представления проекта руководителю | 23 июня 2023 г. |
| Руководитель курсового проекта |  |  | Аршинский В. Л. |
|  | подпись | Фамилия И.О. |

**Содержание**

[Введение 4](#_Toc138323755)

[1 Анализ задания и описание предметной области 5](#_Toc138323756)

[1.2 Постановка задачи 5](#_Toc138323757)

[1.2 Функциональные требования 5](#_Toc138323758)

[2 Проектирование приложения 6](#_Toc138323759)

[2.1 Проектирование базы данных 6](#_Toc138323760)

[2.2 Создание модели данных на физическом уровне 7](#_Toc138323761)

[2.3 Принципиальна схема работы приложения 8](#_Toc138323762)

[2.4 Проектирование классов для работы приложения 9](#_Toc138323763)

[3 Реализация классов 11](#_Toc138323764)

[4 Тестирование 12](#_Toc138323765)

[4.1 Таблица тестов ожидаемых ошибок 12](#_Toc138323766)

[4.2 Результаты тестирования программы 13](#_Toc138323767)

[Заключение 15](#_Toc138323768)

[Список литературы 16](#_Toc138323769)

Введение

Коллекционер моделей занимается обменом моделей с другими коллекционерами. Для облегчения этой задачи ему нужно хранить информацию о имеющихся у него моделях и иметь возможность поделиться списком с другими коллекционерами.

**Цель курсового проекта:** закрепить навыки объектно-ориентированного программирования, для этого создать компьютерное приложение «Коллекции стендовых моделей» для просмотра и редактирования коллекций и хранимых ими стендовых моделей.

1 Анализ задания и описание предметной области

Для наилучшего описания предметной области была создана ER-диаграмма.

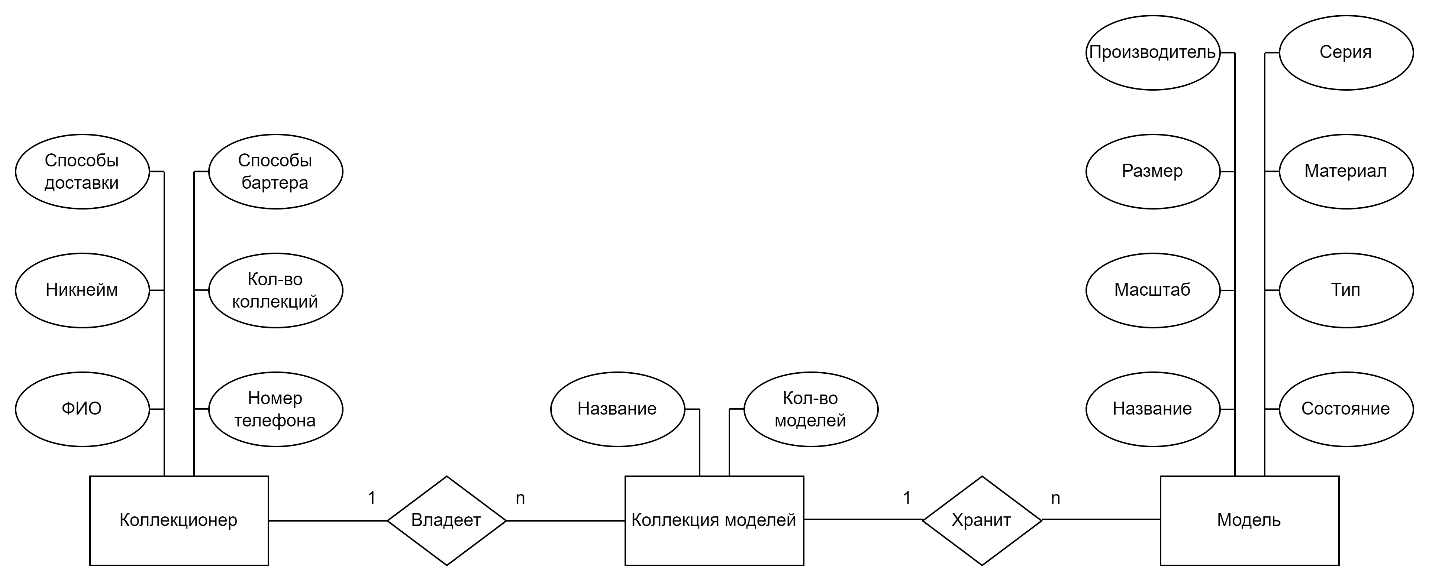


Рисунок 1.1 – ER-диаграмма предметной области

* 1. Постановка задачи

В ходе разработки необходимо создать настольное приложение, позволяющие просматривать коллекции стендовых моделей. Стендовые модели должны хранить информацию о номере модели, названии модели, типе модели, материале, размере, масштабе, производителе, описании, изображении и возможности для обмена. Коллекции должны хранить номер коллекции, название коллекции и стендовые модели .

1.2 Функциональные требования

Приложение должно выполнять следующие функции:

1. Хранение информации о владельце коллекций
2. Работа с коллекциями:
   1. Создание коллекций.
   2. Просмотр коллекций.
   3. Редактирования коллекций путём добавления в них стендовых моделей или их удаления.
   4. Возможность просмотра общей коллекции всех стендовых моделей коллекционера.
3. Работа с моделями
   1. Создание моделей.
   2. Удаление моделей.
   3. Редактирование моделей.

2 Проектирование приложения

* 1. Проектирование базы данных

База данных должна включать в себя 3 таблицы:

* Таблица Person(id записи, Имя, Фамилия, никнейм, номер телефона, id коллекции, название коллекции, возможности для бартера, возможности для доставки)

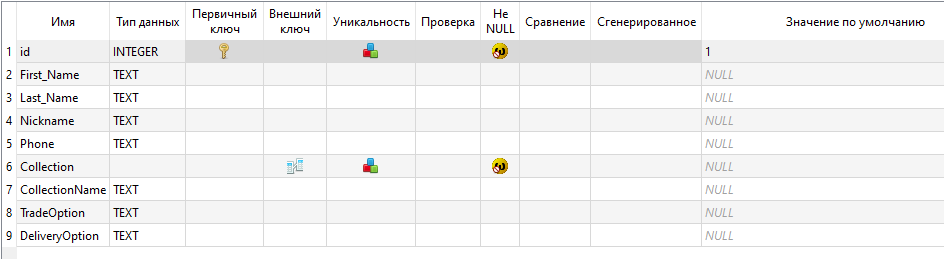


Рисунок 2.1 – Таблица Person

* Таблица Collection(id записи, id коллекции, id модели)

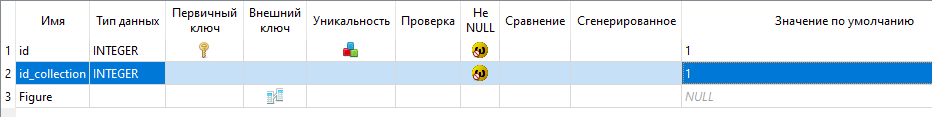


Рисунок 2.2 – Таблица Person

* Таблица figureInfo(id записи, название, тип модели, материал, состояние, размер, масштаб, описание, название изображения, доступность для обмена)

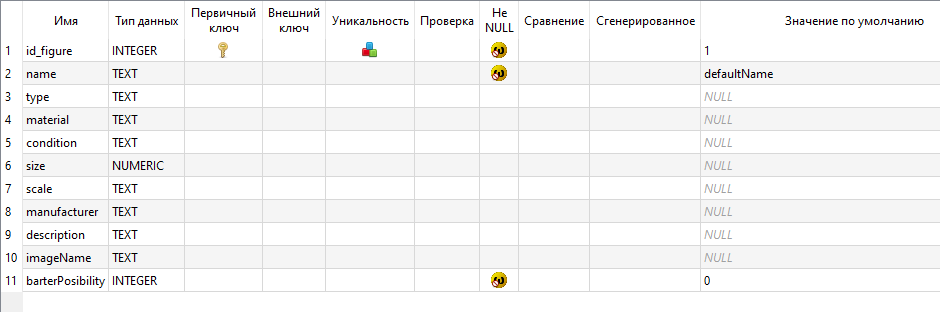


Рисунок 2.3 – Таблица Person

* 1. Проектирование графического интерфейса

На главном окне должны быть размещены верхняя панель(North) с помощью компоновщика GridLayout, левая панель(West) с помощью BoxLayout, центральная панель(Center) с помощью BoxLayout и нижняя панель(South) с помощью GridLayout. На верхней панели будет вводится путь к базе данных и выводится информация о владельце коллекций. На левой и центральной панелях выводится таблицы с приминением JScrollPane. На нижней панели будут размещены кнопки

Главное окно должно: на верхней панели – принимать путь к базе данных и выводить информацию о владельце коллекций; на левой панели – таблицу с списком коллекций; на центральной панели – таблицу с списком моделей в выбранной коллекции; на нижней панели – кнопки для взаимодействия с приложением.

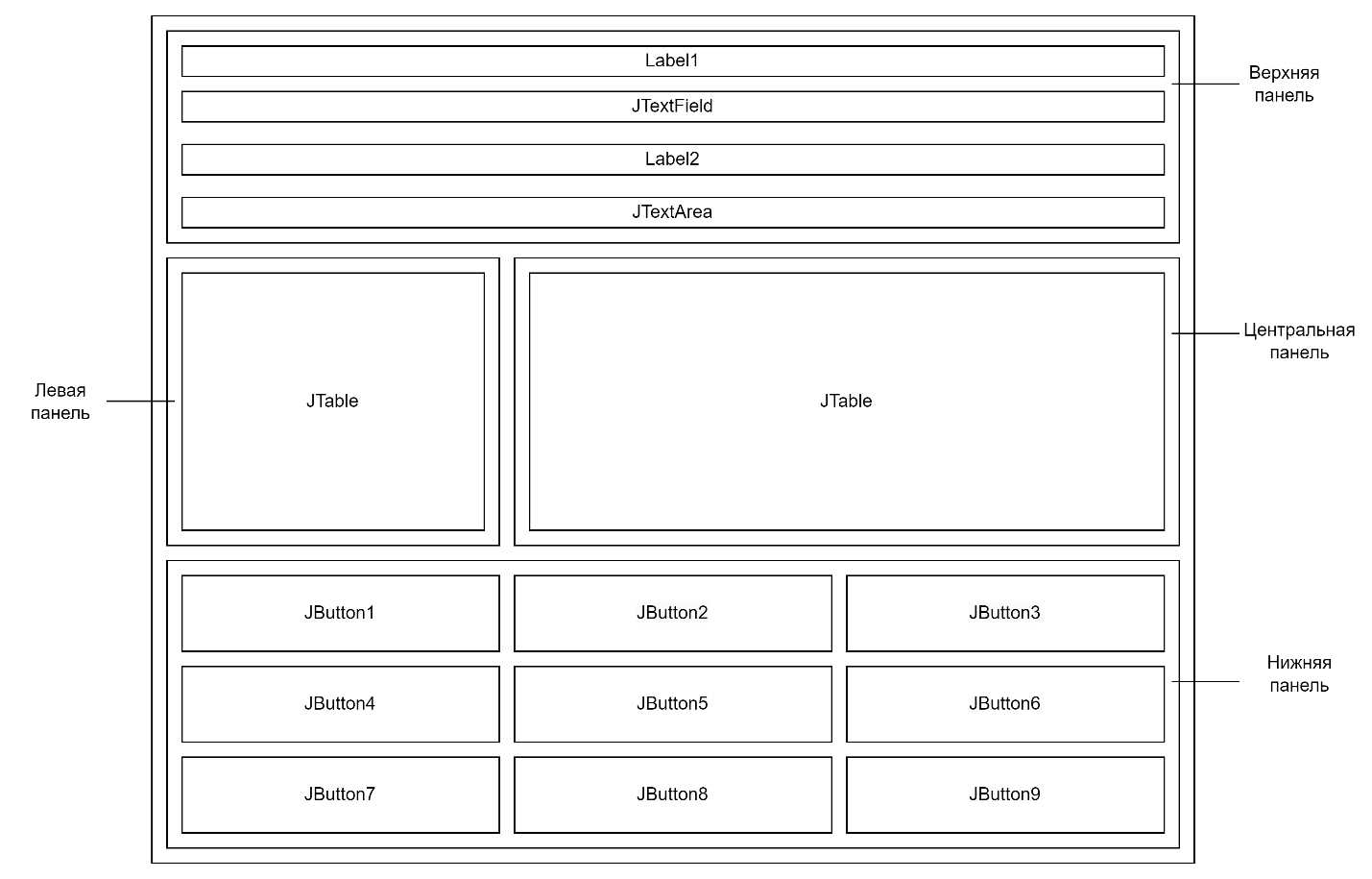


Рисунок 2.4 – Главное окно программы

На окнах создания и изменения моделей с помощью компоновщика GridLayout должна быть размещена центральная панель (Center). На центральной панели будут расположены текстовые поля и их подписи, хранящие параметры моделей, компонент-флажок доступности модели для обмена и кнопка подтверждающее создание или изменение модели.

В элементах Label должны быть подписаны названия вводимых параметров, в JCheckBox должны переключать возможность обмена моделью, в JTextField должны вводить параметры модели, JButton для принятия параметров.

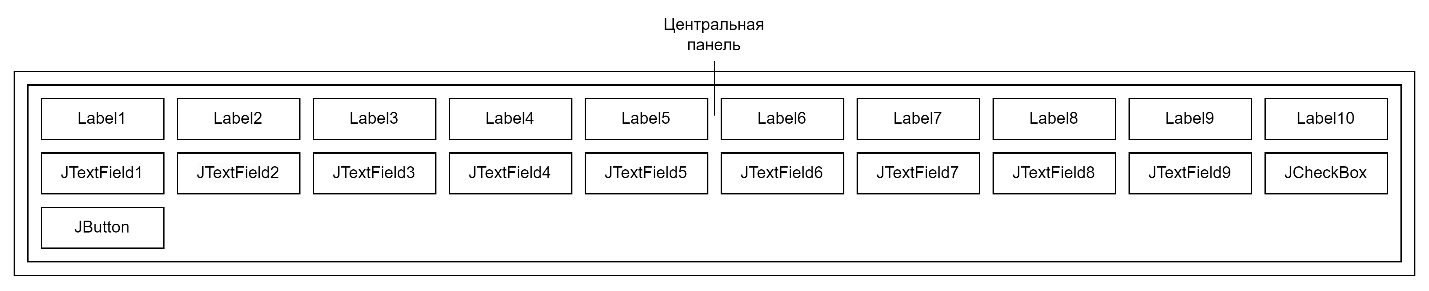


Рисунок 2.5 -Окна создания и изменения моделей

2.3 Принципиальна схема работы приложения

Приложение разбито на 3 сущности: графический интерфейс, контроллер, модель данных. Работа с ними происходит по одному основному сценарию:



Рисунок 2.6 – Концептуальная схема работы приложения

Пользователь даёт команду приложению, которую перехватывает контроллер. Контроллер вызывает определённый метод, обращающийся к модели данных.

Модель данных возвращает данные для отображения. Контроллер передаёт полученные данные графическому интерфейс в виде требуемой для отображения сущности.

* 1. Проектирование классов для работы приложения

SQLworker – класс для обработки обращений к базе данных. Он позволяет считывать нужные данные владельца коллекций, коллекций и стендовых моделей.

Поля:

* Connection connection – поле для установки соединения с базой данных
* Path filePath – поле для хранения пути к базе данных

Методы:

* createConnection – создаёт соединение с базой данных и считывает данные
* saveDB – сохраняет изменённые данные в базу данных
* getFilePath – возвращает путь к базе данных
* setFilePath – задаёт путь к базе данных
* setDefaultFilePath – задаёт путь к базовой базе данных, хранимой в папке с приложением

Person – класс, реализующий структуру владельца коллекций. Позволяет хранить информацию о коллекционере и его коллекциях.

Поля:

* String firstName – поле, хранящее имя владельца коллекций
* String lastName – поле, хранящее фамилию владельца коллекций
* String nickname – поле, хранящее никнейм владельца коллекций
* String phone – поле, хранящее номер телефона владельца коллекций
* ArrayList<FigureCollection> collection – поле, хранящее список коллекций
* FigureCollection baseCollection – поле, хранящее общий список стендовых моделей
* String tradeOption – поле, хранящее возможные способы обмена
* String deliveryOption – поле, хранящее возможные способы доставки моделей

Методы:

* addCollection – добавляет новую коллекцию в список коллекций
* addFigureInBaseCollection – добавляет стендовую модель в общий список всех моделей коллекционера
* setFigure – заменяет стендовую модель под указанным номером на полученную методом

FigureCollection – класс, реализующий структуру коллекции стендовых моделей. Он позволяет хранить список стендовых моделей и изменять их.

Поля:

* int ID – поле, хранящее номер коллекции
* String collectionName – поле, хранящее название коллекции
* List<Figure> collection – поле, хранящее список моделей в коллекции

Методы:

* getFigure – возвращает стендовую модель по указанному номеру
* setFigure - заменяет стендовую модель под указанным номером на полученную методом
* getLastID – получает номер последней стендовой модели в списке
* addFigure – добавляет стендовую модель в список
* deleteFigure – удаляет стендовую модель под указанным номером

Figure – класс, реализующий структуру стендовой модели. Позволяет хранить информацию о модели редактировать её.

Поля:

* int ID – поле, хранящее номер стендовой модели
* String name – поле, хранящее название стендовой модели
* String figureType – поле, хранящее тип стендовой модели
* String material – поле, хранящее материал стендовой модели
* String condition – поле, хранящее текущее состояние стендовой модели
* float size – поле, хранящее размер стендовой модели в см
* String scale – поле, хранящее масштаб стендовой модели
* String manufacturer – поле, хранящее производителя стендовой модели
* String description – поле, хранящее описание стендовой модели
* String imageName – поле, хранящее название изображения стендовой модели
* boolean barterPosibility – поле, хранящее возможность обмена данной стендовой моделью

PersoneModel - класс, реализующий модель данных владельца коллекций стендовых моделей для вывода и редактирования данных.

Методы:

* addCollection – добавляет новую коллекцию в список коллекций
* setFigure - заменяет стендовую модель под указанным номером на полученную методом
* setFilePath – указывает путь на диске к базе данных
* getFilePath – возвращает путь на диске к базе данных
* setDefaultFilePath – задаёт путь к базовой базе данных, хранимой в папке с приложением
* addFigure – добавляет стендовую модель в список

FigureModel – класс, реализующий модель данных стендовой модели для вывода и редактирования данных.

Методы:

* getFigure – возвращает стендовую модель по указанному номеру
* setFigure - заменяет стендовую модель под указанным номером на полученную методом
* addFigure – добавляет стендовую модель в список
* deleteFigure – удаляет стендовую модель под указанным номером

GUI – класс, реализующий графический интерфейс для взаимодействия с пользователем.

Методы:

* MainWindow –создаёт и выводит основное окно приложения
* MyTable – создаёт таблицы для вывода данных и выводит их на графический интерфейс
* ButtonPanel – создаёт панель с кнопками для взаимодействия с приложением и выводит их на графический интерфейс
* isNumeric – проверяет введённую строку является ли она числом

3 Реализация классов

Программа была реализована с использованием Swing библиотеки и пакет JDBC.

Swing – библиотека для языка Java. Она нужна для создания графического интерфейса приложения. Относится к классу JFC, который представляет набор библиотек, содержащие средства для создания графических оболочек.

Swing можно использовать по конкретным частям, для реализации приложения было использовано:

* Swing.Table.AbstractTableModel – для работы с моделью данных для вывода в таблицы и их редактирования

JDBC – пакет для языка Java. Нужен для получения соединения с базой данных по описанному URL. JDBC основан на концепции драйверов, способных загружаться во время работы программы и вызваться автоматически, когда программа требует URL, содержащий протокол, за который драйвер отвечает.

В качестве СУБД была выбрана SQLite, поскольку она проста в освоении и является открытым ПО.

1. Тестирование

Тестирование приложения производится ручным тестированием для проверки работы приложения.

4.1 Таблица тестов ожидаемых ошибок

Таблица 4.1 – Тестирование ожидаемых ошибок

|  |  |  |
| --- | --- | --- |
| № | Назначение теста | Ожидаемый результат |
| 1 | Чтение базы данных по введённому пути | Чтение базы данных и вывод информации о владельце коллекций и коллекциях |
| 2 | Попытка чтения базы данных по неправильному пути | Вывод сообщения о неправильном пути |
| 3 | Нажатие кнопки «Выбрать коллекцию» с выделением строки в таблице коллекций | Вывод информации о моделях в выбранной коллекции в таблице |
| 4 | Нажатие кнопки «Вывести все фигуры» | Вывод информации о всех моделях коллекционера |
| 5 | Нажатие кнопки «Добавить фигурку» | Вывод окна для ввода данных новой модели |
| 6 | Нажатие кнопки «Сохранить базу данных» | Сохранение данных моделей и владельца моделей в файл базы данных |

4.2 Результаты тестирования программы

**Тест 1.**

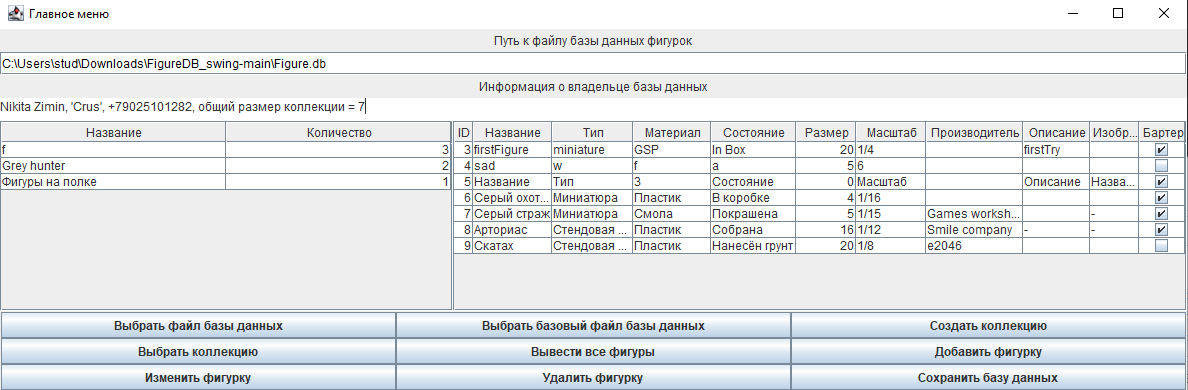


Рисунок 4.1 – Результаты теста №1 (успешно)

**Тест 2.**

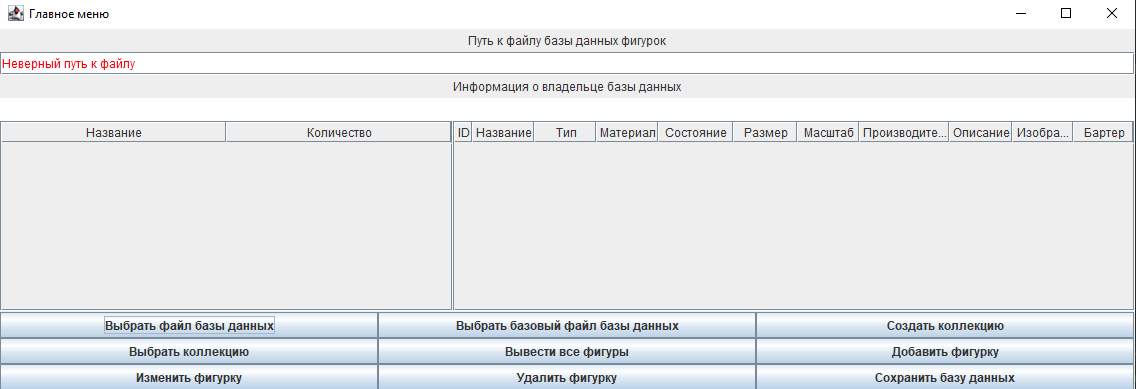


Рисунок 4.2 – Результаты теста №2 (успешно)

**Тест 3.**

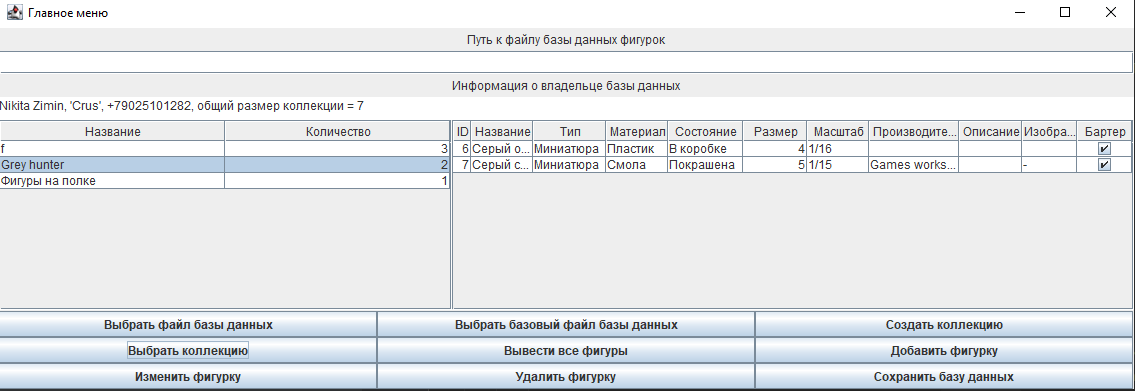


Рисунок 4.3 – Результаты теста №3 (успешно)

**Тест 4.**

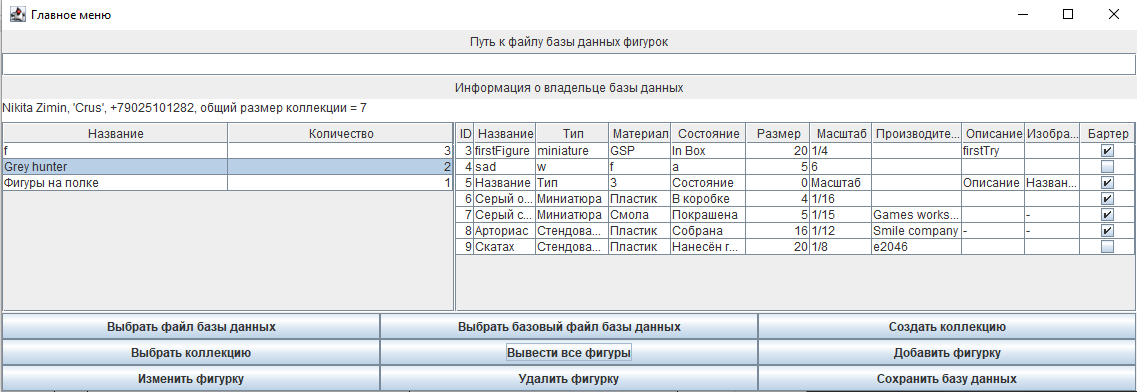


Рисунок 4.4 – Результаты теста №4 (успешно)

**Тест 5.**

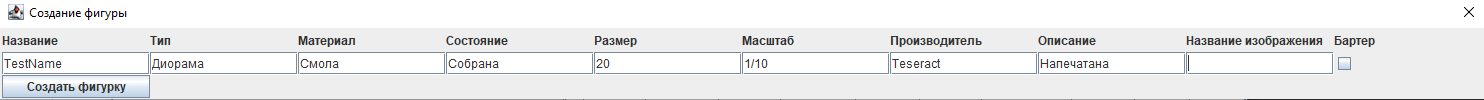


Рисунок 4.5 – Результаты теста №5 (успешно)

**Тест 6.**

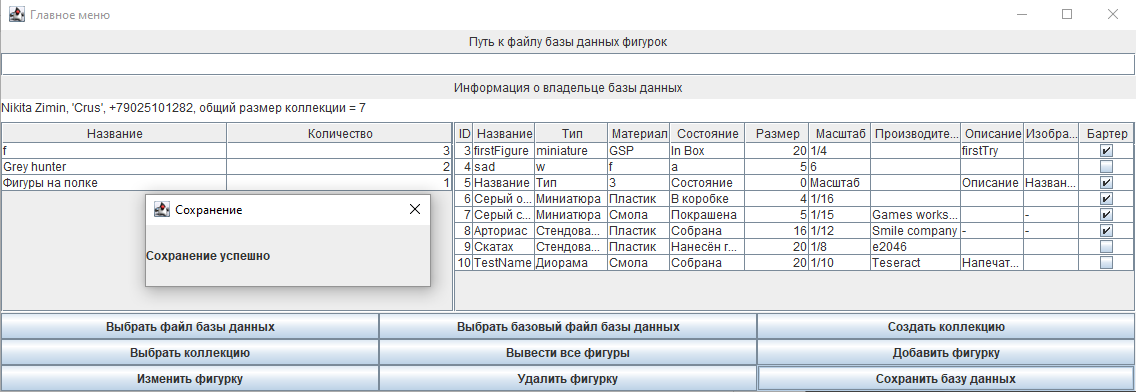


Рисунок 4.6 – Результаты теста №6 (успешно)

Заключение

В процессе разработки приложения были получены и закреплены практические навыки проектирования и применения объектного-ориентированного подхода к реализации приложений. Получены навыки проектирования пользовательского интерфейса и работы с базой данных путем использования библиотеки Swing и драйвера JDBC. Закреплены навыки составления тестов и UML-диаграмм классов.

В ходе выполнения проекта было разработано 6 тестов. Приложение успешно прошло все 6 тестов.

Исходный код был размещен в открытом репозитории GitHub[4].

Проект может быть продолжен, если добавить новые функции, а именно:

* Возможность перемещать модели между коллекциями
* Добавление опции выбора базы данных в специальном окне
* Добавление опции просмотра описания и изображения модели в специальных окнах
* Добавление возможности поиска моделей с заданными параметрами, например, модели из определённого материала

Поставленные задачи выполнены, цель проекта достигнута.

Список литературы

1. Миллионы на играх в войну: как устроен бизнес британской Games Workshop // BBC News[сайт], URL: <https://www.bbc.com/russian/features-45024308> (дата обращения: 22.06.2023)
2. Отношения классов – от UML к коду [электронный ресурс] // Хабр [сайт], URL: https://habr.com/ru/post/150041/ (дата обращения: 05.06.2023)
3. UML для самых маленьких: диаграмма классов [электронный ресурс] // Хабр [сайт], URL: https://habr.com/ru/post/511798/ (дата обращения: 06.06.2023)
4. Репозиторий с исходным кодом проекта [Электронный ресурс] // GitHub: [сайт], URL: <https://github.com/CrusaderKleptoman/FigureDB_swing>