

Обработка метаинформации баз данных

Работа выполнена студентами группы 4ПР-1:
Барановым К.С. И Фадиным К.Д.
Руководитель: Ишкова Лариса Георгиевна
Рецензент: кто-то там

Актуальность выбранной темы

Тема актуальна, в первую очередь, тем, что в настоящее время широко распространены технологии работы с базами данных и в погоне за наиболее оптимальной конфигурацией среди уже существующих вручную перебрать множество вариантов и выяснить их различия не представляется возможным. Поэтому для автоматизации поиска различий в базах данных было создано данное программное обеспечение.



Цель исследования

Главной целью данного дипломного проекта является разработка утилиты сравнения баз данных.



Задачи исследования

Задачи исследования:

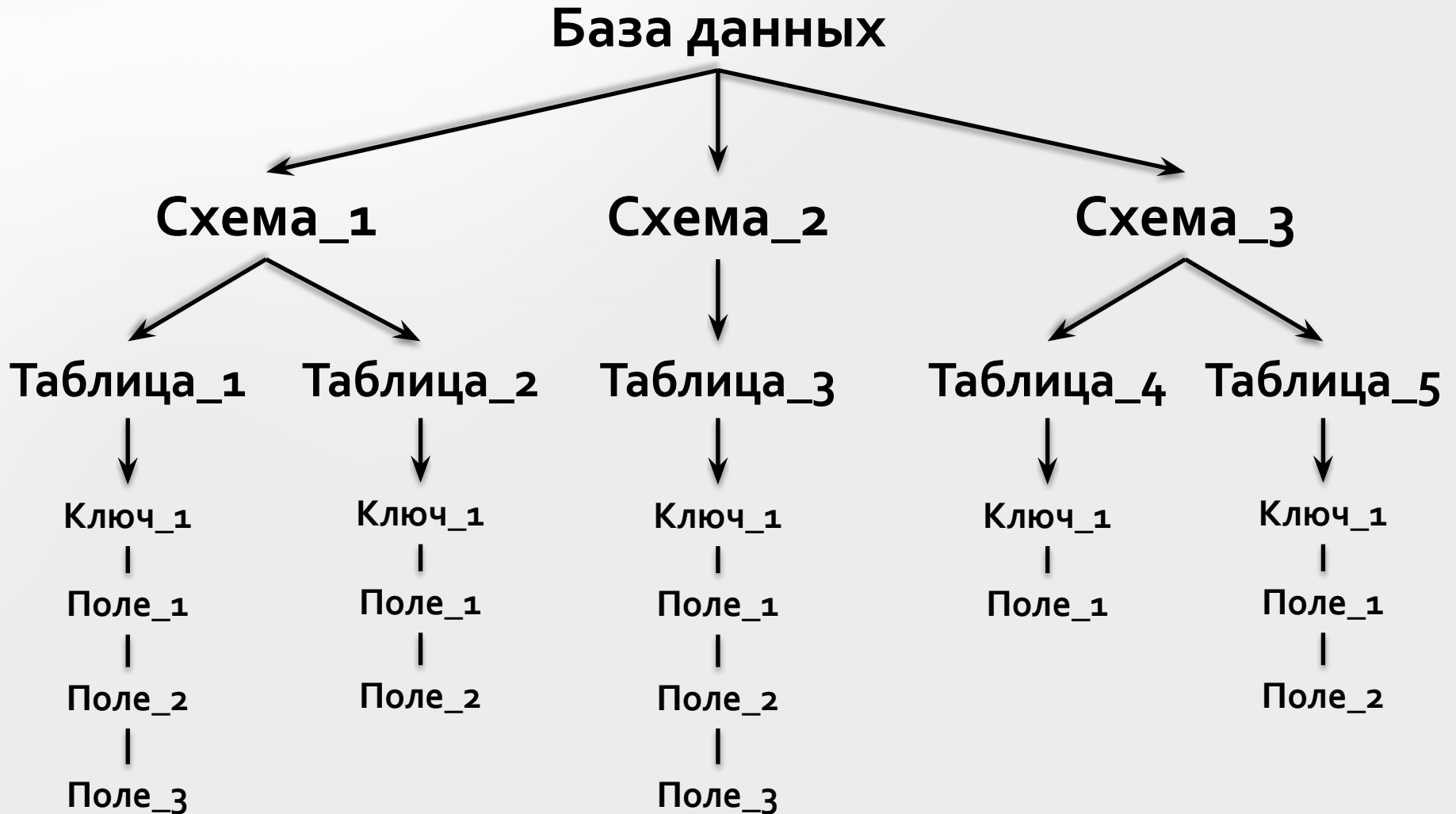
- Изучить особенности извлечения метаданных из баз данных в программе на языке Java;
- Разработать реализацию поставленной цели в программном коде на языке Java;
- Спроектировать пользовательский интерфейс приложения на языке Java;
- Объединить интерфейс и реализацию в единое приложение с использованием шаблона проектирования MVC.

Инструменты исследования

Для реализации задания, благодаря своим преимуществам и отсутствием существенных недостатков, был выбран язык программирования Java, а СУБД, с базами данных которой будет работать приложение, стала DB2. Средой разработки стала Eclipse.

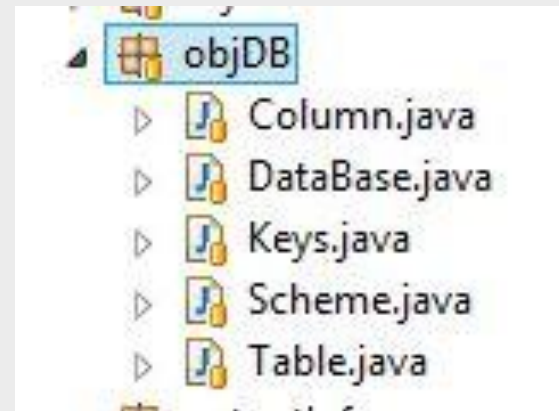


Теоретическая база исследования. Структура БД

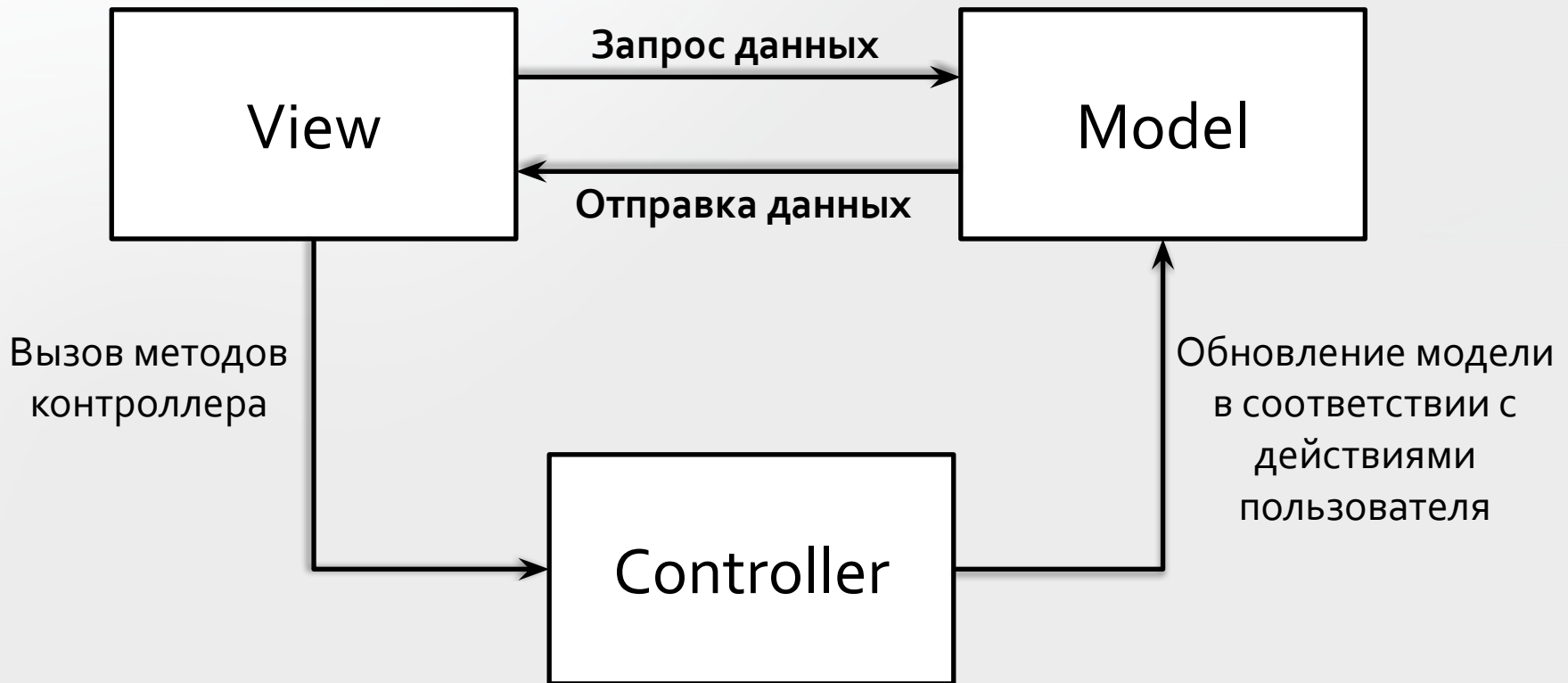


Реализация структуры базы данных в программе

На основе собранных данных была создана следующая структура классов базы данных в программе:



Шаблон проектирования MVC



Пример метода создания файла с метайнформацией

```
public void CreateDBMetaFile(String[] dbInfo) {
    DBWork db = new DBWork();
    DataBase db1 = db.createObjDB(dbInfo);
    if (db.getState()) {
        dirPath += "\\out.dat";
        DBCompareController.AddLogMessage("Сохраняю файл на
диск");
        FileWorkWriteDB fileDB = new FileWorkWriteDB(db1);
        StringBuilder DBString = fileDB.CreateFileDB();
        if (Filework.write(dirPath, DBString)) {
            db.close();
            String str = "Программа успешно завершена!
Созданный файл находится в " + dirPath;
            DBCompareController.AddLogMessage(str);
        }
    }
}
```

Ожидаемые результаты от реализации проекта

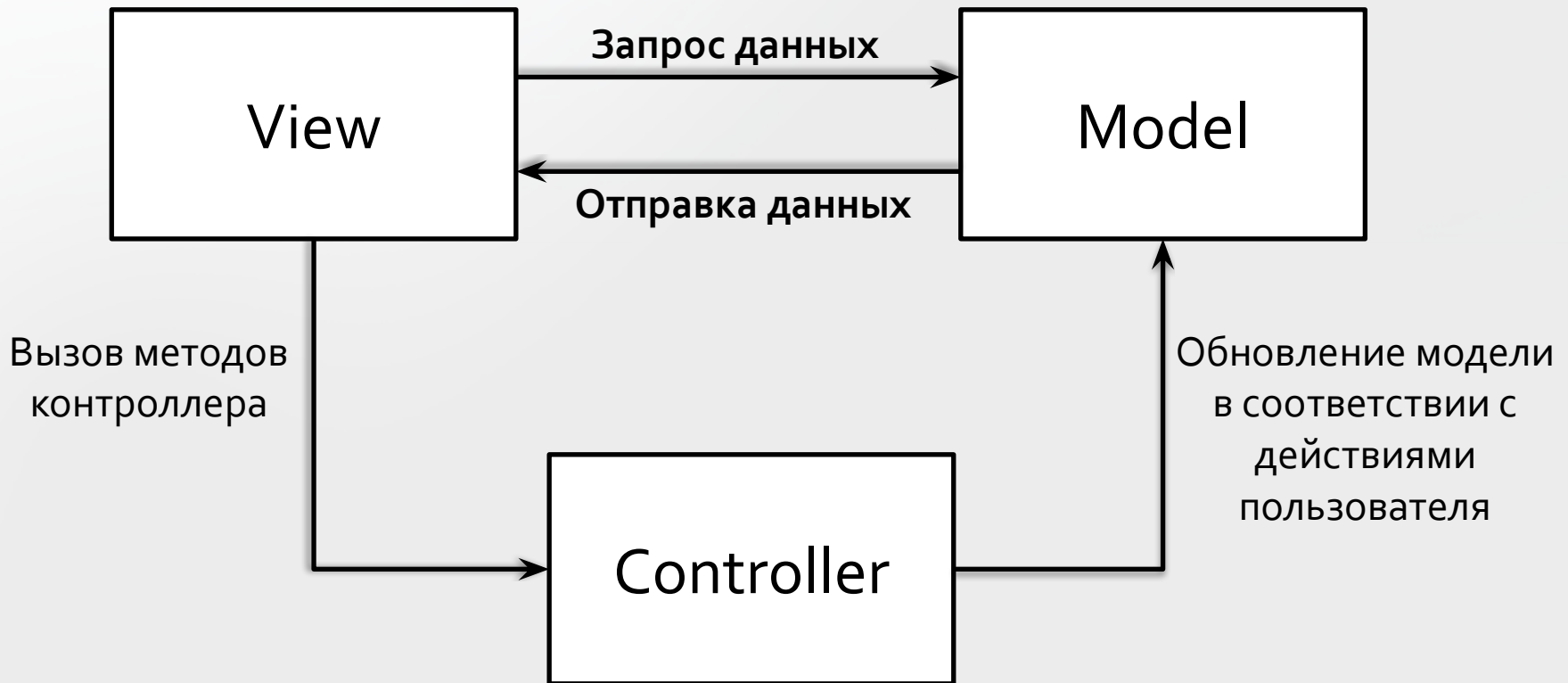
- Время поиска различий в базах данных уменьшается в разы по сравнению с ручным поиском;
- Администратор, создающий файл метаописания не рискует утечкой информации, так как этот файл создается с зашифрованным содержимым;
- Проект пригодится для выполнения любой задачи, где требуется сравнение БД, а возможность несложного расширения функционала позволит еще и автоматизировать процесс.

Основные источники информации

- Фаулер М. Рефакторинг: улучшение существующего кода/Пер. с англ. – СПб: Символ-плюс, 2003. – 432с., ил.
- Макконнелл С. Совершенный код. Мастер-класс / Пер. с англ. — М. : Издательство «Русская редакция», 2010. — 896 с.
- <http://www-01.ibm.com/support/knowledgecenter/>
(Дата обращения: 21 апреля 2015)
- <https://ru.wikipedia.org/wiki/Model-View-Controller>
(Дата обращения: 15 мая 2015)

Спасибо за просмотр!

Шаблон проектирования MVC



View (Представление)

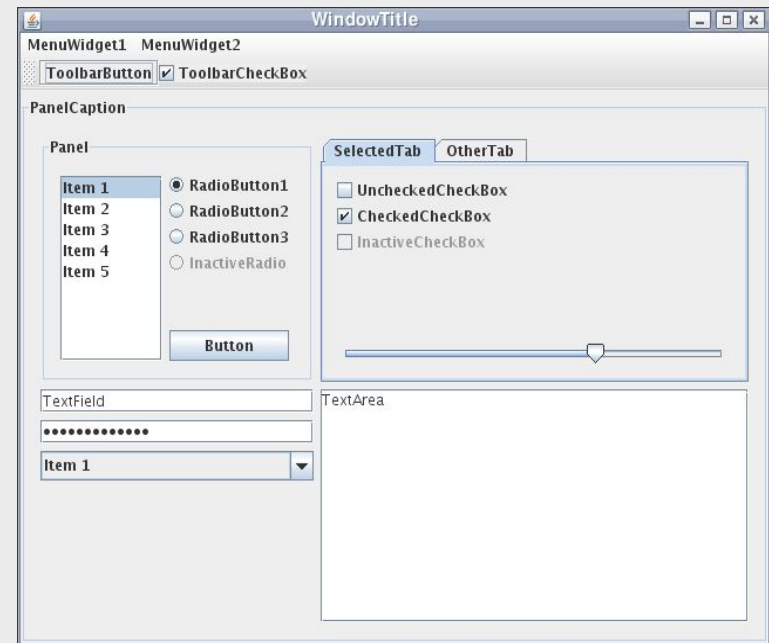
Представление используется для описания внешнего отображения данных. Оно может напрямую обращаться к Модели или Контроллеру за получением готовых к показу данных.

Представление не может напрямую обращаться к базе данных и не должно работать с данными, полученными напрямую от пользователей.

Описание графического интерфейса в Java. Библиотека Swing.

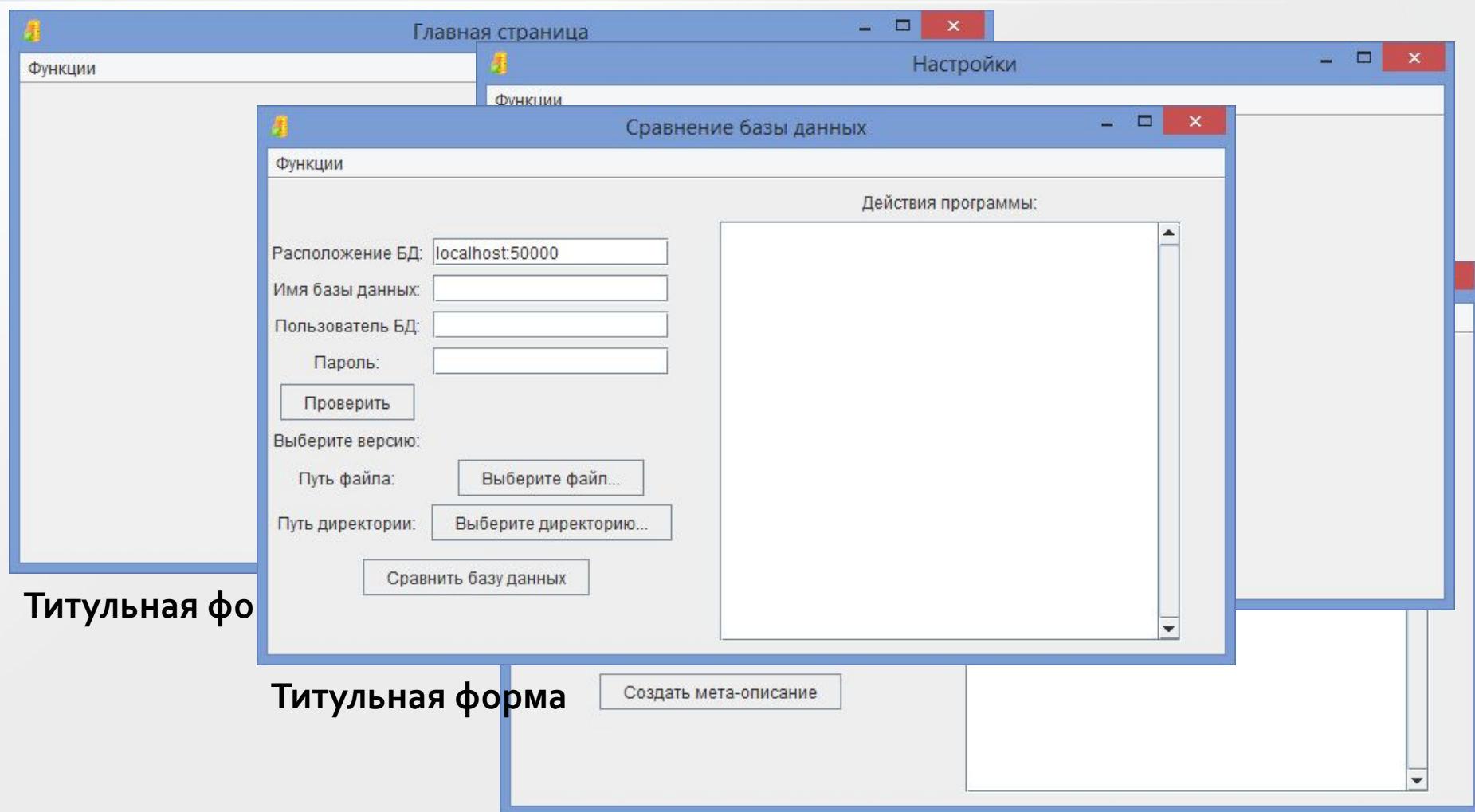
Swing — библиотека для создания графического интерфейса для программ на языке Java.

```
private void CreateFirstPanel() {  
    BlockMenuItem(o);  
    centerIcon.setIcon(createIcon("/image/ProgramLogo.png"));  
    Font font = new Font("Colibri", 1, 16);  
    nameLabel.setFont(font);  
  
    firstPanel.setLayout(new GridBagLayout());  
  
    GridBagConstraints gbc = new GridBagConstraints();  
    gbc.gridwidth = GridBagConstraints.REMAINDER;  
    gbc.insets = new Insets(15, 1, 1, 1);  
  
    firstPanel.add(centerIcon, gbc);  
    firstPanel.add(nameLabel, gbc);  
  
}
```



Примеры графических элементов

Формы приложения



Титульная фо

Титульная форма

Форма создания файла с метаинформацией

Взаимодействие с пользователем.

Контроллер приложения

Контроллер — связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение. Контроллер отвечает за обработку запросов пользователя.

Контроллер, также как и Представление, не может работать с базой данных, но и выводить информацию на форму приложения тоже не способен.

Примеры методов контроллера

```
class ChooseFileActionListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        String filepath;
        JFileChooser fileOpen = new JFileChooser();
        fileOpen.setSelectionMode(JFileChooser.FILES_ONLY);
        int ret = fileOpen.showDialog(null, "Открыть файл");
        if (ret == JFileChooser.APPROVE_OPTION) {
            File file = fileOpen.getSelectedFile();
            filepath = file.getPath();
            theModel.setFilePath(filepath);
            Object button = e.getSource();
            theView.SetDirPathOnButton(button, filepath);
            AddLogMessage("Выбран входной файл:");
            AddLogMessage(" " + filepath);
            filePathFlag = true;
        }
    }
}
```

```
class ExitMenuItemActionListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        theView.dispose();
    }
}
```

Основные источники информации

- Карлсон Д. Eclipse / Пер. с англ. – М.:Лори, 2008. – 336с.
- Шилдт Г. Java 8. Полное руководство. 9-е издание / Пер. с англ. – М.:Вильямс, 2015. – 1377с.
- <https://javaswing.wordpress.com/> (Дата обращения: 17 мая 2015)
- <https://ru.wikipedia.org/wiki/Swing> (Дата обращения: 17 мая 2015)

Спасибо за просмотр!