

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Таныгин Максим Олегович  
Должность: и.о. декана факультета фундаментальной и прикладной информатики  
Дата подписания: 21.09.2023 13:09:47  
Уникальный программный ключ:  
65ab2aa0d384efe8480e6a4c688eddbc475e411a

## МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ  
Проректор по учебной работе  
О.П. Фоктионова  
« 24 » 12 (ЮЗГУ) 2017 г.



### ВВОД ДАННЫХ В БАЗУ ДАННЫХ ДЛЯ СУБД SQLITE ИЗ WEB ПРИЛОЖЕНИЯ

Методические указания по выполнению лабораторной работы  
для студентов направления подготовки 09.03.04 «Программная  
инженерия» по дисциплине «Конструирование программного  
обеспечения»

Курск 2017

УДК 004.65

Составители: В.Г. Белов, Т.М. Белова

Рецензент

Кандидат технических наук, доцент кафедры программной инженерии ЮЗГУ И.Н. Ефремова

**Ввод данных в базу данных для СУБД SQLite из Web приложения:** методические указания по выполнению лабораторной работы по дисциплине "Конструирование программного обеспечения" для студентов направления подготовки 09.03.04 "Программная инженерия" / Юго-Зап. гос. ун-т; сост.: В.Г. Белов, Т.М. Белова, – Курск, 2017. – 20 с.: ил. 13, табл. 2.

Изложена последовательность действий с системой Java Eclipse для ввода информации из Web приложения в базу данных при работе с СУБД SQLite.

Материал предназначен для студентов направления подготовки бакалавров 09.03.04 «Программная инженерия», а также будет полезен студентам всех направлений подготовки, изучающим технологии разработки баз данных.

Текст печатается в авторской редакции.

Подписано в печать 27.12.17. Формат 60x84 1/16.  
Усл. печ. л. 10. Уч.-изд. л. 09. Тираж 100 экз. Заказ 4890. Бесплатно.  
Юго-Западный государственный университет  
305040, Курск, ул.50 лет Октября, 94.

## Содержание

1	Цель лабораторной работы .....	4
2	Порядок выполнения лабораторной работы .....	5
2.1	Предварительная подготовка.....	5
2.2.	Пример выполнения лабораторной работы .....	17
3	Содержание отчета по лабораторной работе .....	20
4	Вопросы к защите лабораторной работе .....	21

## **1 Цель лабораторной работы**

Целью лабораторной работы является приобретение знаний и навыков для внесения данных в базу данных SQLite посредством Java Eclipse.

Вы можете использовать полученные навыки для внесения в базу данных клиент-серверного приложения.

## 2 Порядок выполнения лабораторной работы

### 2.1 Предварительная подготовка

Перед тем как перейти к написанию кода для внесения информации в базу данных необходимо выполнить следующие шаги:

1. Создайте новый .html файл в папке с проектом. Для этого перейдите в Package Explorer и на папке с необходимым проектом нажмите правой кнопкой мыши, выберите пункт [New] – [Other] – HTML File (рисунок 1).

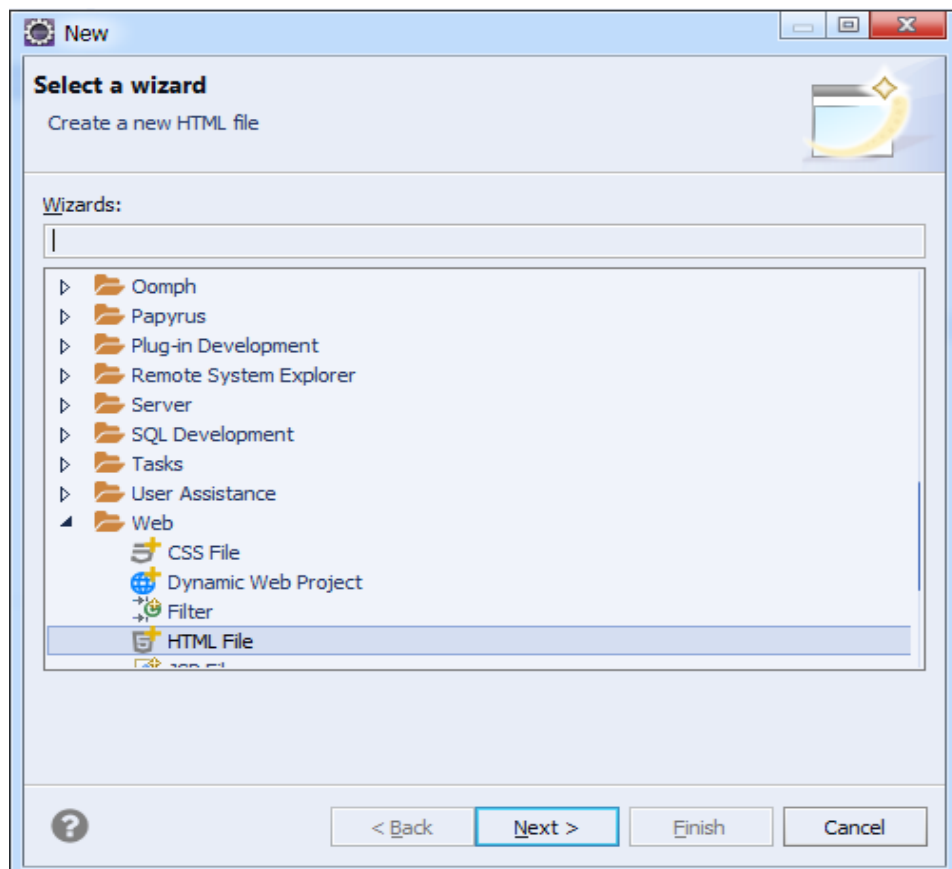
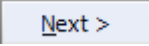


Рисунок 1 – Создание HTML файла

2. Заполните поле «File Name», например, «InsertMySQL» и нажмите кнопку Next  (рисунок 2).

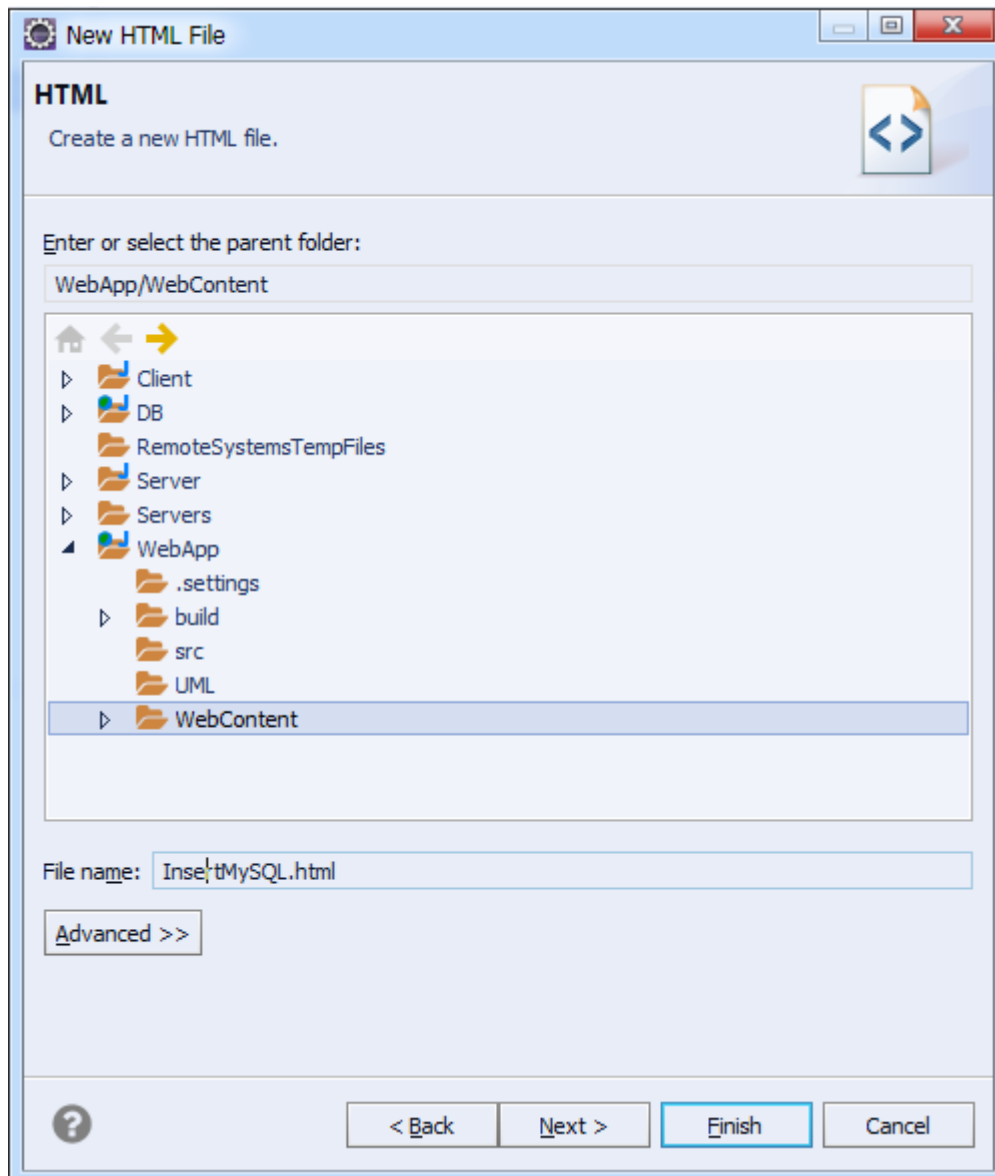


Рисунок 2 – Создание нового HTML файла

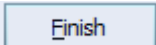
3. Выберите HTML-шаблон, например, «HTML-5» и нажмите Finish  (рисунок 3).



Рисунок 3 – Выбор HTML-шаблона

4. После выполнения предыдущих шагов откроется рабочая область созданного файла (рисунок 4).

```

1 |!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Insert title here</title>
6 </head>
7 <body>
8
9 </body>
10 </html>

```

Рисунок 4 – Рабочая область HTML-файла

В строке:

```
<title>Insert title here</title>
```

Указывается заголовок вкладки, который отображается в браузере. Вставьте другое название, если это необходимо, например, «Test Server» (рисунок 5).

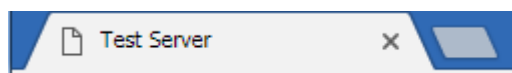


Рисунок 5 – Изменение заголовка вкладки

- Создайте 5 текстовых полей и кнопку для внесения информации в базу данных. Для создания необходимого внесите изменения в тело HTML-файла (body). Пример кода приведен ниже:

```

<form action="SQLiteServlet" method="post" >
ID :<input type="number" name="contactid">
Имя :<input type="text" name="firstname">
Фамилия :<input type="text" name="lastname">
Email :<input type="email" name="email">
Номер телефона :<input type="number" name="phone">
<input type="submit" value="Отправить данные в базу данных SQLite">

```



</form>

В строке:

```
<form action="SQLiteServlet" method="post" >
```

В поле «action» указывается обработчик, к которому обращаются данные формы при их отправке на сервер.

В поле «method» указывается исполняемый метод, ими могут быть doGet или doPost. В качестве примера будет использован метод doPost.

В строках:

```
ID :<input type="number" name="contactid">
```

```
Имя :<input type="text" name="firstname">
```

```
Фамилия :<input type="text" name="lastname">
```

```
Email :<input type="emailr" name="email">
```

```
Номер телефона :<input type="number" name="phone">
```

Создается пять текстовых поля для введения информации в них и последующей отправки на сервер в базу данных.

В поле «type» указывается тип данных создаваемого объекта.

Список возможных параметров для поля «type» приведен ниже в таблице 1:

Таблица 1 – Возможные параметры для поля «type»

button	Кнопка.
checkbox	Флажки. Позволяют выбрать более одного варианта из предложенных.
file	Поле для ввода имени файла, который пересылается на сервер.
hidden	Скрытое поле. Оно никак не отображается на веб-странице.
image	Поле с изображением. При нажатии на рисунок данные формы отправляются на сервер.

password	Обычное текстовое поле, но отличается от него тем, что все символы показываются звездочками. Предназначено для того, чтобы никто не подглядел вводимый пароль.
radio	Переключатели. Используются, когда следует выбрать один вариант из нескольких предложенных.
reset	Кнопка для возвращения данных формы в первоначальное значение.
submit	Кнопка для отправки данных формы на сервер.
text	Текстовое поле. Предназначено для ввода символов с помощью клавиатуры.

В HTML5 были добавлены новые значения, их список приведен ниже в таблице 2.

Таблица 2 – Добавленные параметры в HTML5

color	Виджет для выбора цвета.
date	Поле для выбора календарной даты.
datetime	Указание даты и времени.
datetime-local	Указание местной даты и времени.
email	Для адресов электронной почты.
number	Ввод чисел.
range	Ползунок для выбора чисел в указанном диапазоне.
search	Поле для поиска.
tel	Для телефонных номеров.
time	Для времени.
url	Для веб-адресов.
month	Выбор месяца.
week	Выбор недели.

В поле «name» указывается имя переменной которой будет присвоена введенная информация.

В строке:

```
<input type="submit" value="Отправить данные в базу данных SQLite">
```

Создается кнопка для подтверждения отправки данных. В поле «value» указывается текст который будет отображен на кнопке.

#### 6. Откройте файл web.xml (дескриптор развертывания)

Дескриптор развертывания — это конфигурационный файл артефакта, который будет развернут в контейнере сервлетов. В спецификации Java Platform, Enterprise Edition дескриптор развертывания описывает то, как компонент, модуль или приложение (такое, как веб-приложение или приложение предприятия) должно быть развернуто. Он находится в папке с проектом. Для его открытия разверните папку с проектом, после раскройте папку «WebContent» и «WEB-INF» и откройте файл. (рисунок 6).

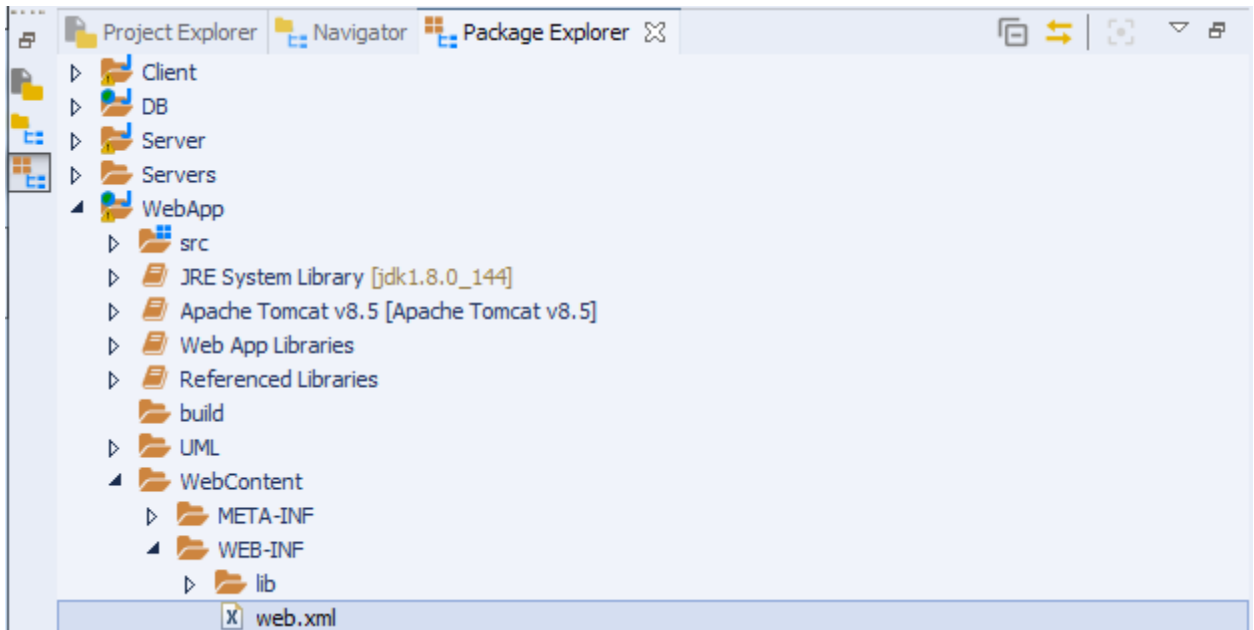


Рисунок 6 – Локализация файла web.xml

7. После открытия файла внизу рабочей области переключитесь на вкладку «Source» (рисунок 7). После чего откроется рабочий код файла web.xml (рисунок 8).

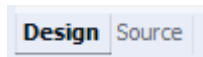


Рисунок 7 – Вкладки Design|Source

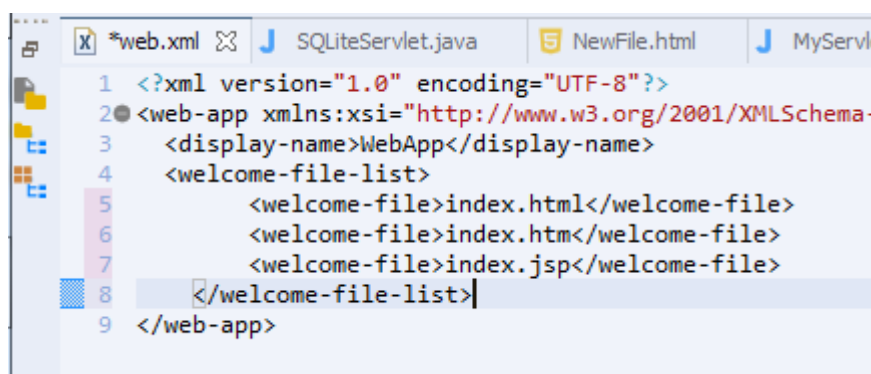


Рисунок 8 – Файл web.xml

Блок `<welcome-file-list>` отвечает за файлы которые будут открыты при обращении к серверу. Если первый файл в первой строке существует то откроется именно он, если нет, сервер будет проходить по телу блока до тех пор пока не найдет существующий файл. Приведите блок `<welcome-file-list>` к следующему виду:

```
<welcome-file-list>

<welcome-file>InsertMySQL.html</welcome-file>

</welcome-file-list>
```

Сервер будет обращаться к файлу «InsertMySQL.html» и откроет его для отображения.

8. После выполнения шагов 4-7 перейдите в файл `SQLiteServlet.java` и в теле файла создайте метод `doPost`.

Пример кода приведен ниже:

```
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException
{
}
}
```

Вставьте в тело метода следующий код:

```
PrintWriter out = response.getWriter();
    try
    {
        Class.forName("org.sqlite.JDBC");
        con = DriverManager.getConnection(url, name, pass);
        response.setContentType("text/html");
        response.setCharacterEncoding("UTF-8");
        String contact_id = request.getParameter("contactid");
        String firstname = request.getParameter("firstname");
        String lastname = request.getParameter("lastname");
        String email = request.getParameter("email");
```

```

String phone = request.getParameter("phone");

PreparedStatement pst;
try {
    pst = con.prepareStatement("insert into
contacts(contact_id, first_name,last_name,email, phone) values(?,?,?,?,?)");

    pst.setString(1,contact_id);
    pst.setString(2,firstname);
    pst.setString(3,lastname);
    pst.setString(4,email);
    pst.setString(5,phone);

    int i = pst.executeUpdate();
    if(i!=0){
        out.println("<br>Запись добавлена в базу данных");
        out.println("<!DOCTYPE html>\r\n" +
            "<html>\r\n" +
            "<head>\r\n" +
            "<meta charset=\"UTF-8\">\r\n" +
            "<title>Test Server</title>\r\n" +
            "</head>\r\n" +
            "<body>\r\n" +
            "\r\n" +
            "<p><a href=\"InsertMySQL.html\">На главную
страницу</a></p>\r\n" +
            "</body>\r\n" +
            "</html>");
        out.println("<!DOCTYPE html>\r\n" +
            "<html>\r\n" +
            "<head>\r\n" +
            "<meta charset=\"UTF-8\">\r\n" +
            "<title>Test Server</title>\r\n" +
            "</head>\r\n" +
            "<body>\r\n" +
            "\r\n" +
            "<p><a href=\"SQLiteServlet\">Посмотреть
обновленную базу данных</a></p>\r\n" +
            "</body>\r\n" +
            "</html>");
    }
    else{
        out.println("Что-то пошло не так");
    }
}
catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

out.close();
}

```

```

catch(SQLException | ClassNotFoundException ex)
{
}
finally
{
    try
    {
        if(con != null)
            con.close();
    }
    catch(SQLException ex)
    {
    }
}
}
}

```

В строках:

```

response.setContentType("text/html");
response.setCharacterEncoding("UTF-8");

```

Задается тип отображаемой страницы и ее кодировка.

В строках:

```

String contact_id = request.getParameter("contactid");
String firstname = request.getParameter("firstname");
String lastname = request.getParameter("lastname");
String email = request.getParameter("email");
String phone = request.getParameter("phone");

```

Создаются строковые переменные для хранения информации из созданных ранее полей для .html страницы.

В строке:

```

PrintWriter out = response.getWriter();

```

Создается экземпляр класса `PrintWriter`, который используется для вывода информации.

В строке:

```
PreparedStatement pst;
```

Создается экземпляр класса `PreparedStatement`, который используется для хранения и запоминания SQL-запросов.

В теле блока:

```
try {
    pst = con.prepareStatement("insert into
contacts(contact_id, first_name,last_name,email, phone) values(?,?,?,?,?)");

    pst.setString(1,contact_id);
    pst.setString(2,firstname);
    pst.setString(3,lastname);
    pst.setString(4,email);
    pst.setString(5,phone);
```

Помещается SQL-запрос на добавление записи в базу данных.

Обратите внимание, что вместо значений указываются «?».

Сколько необходимо передать значений, столько и символов «?» указывается. За исключением, например, поля «id», если оно является первичным ключом и его значение увеличивается автоматически при добавлении записи в базу данных.

В коде:

```
int i = pst.executeUpdate();
    if(i!=0){
        out.println("<br>Запись добавлена в базу данных");
        out.println("<!DOCTYPE html>\r\n" +
            "<html>\r\n" +
            "<head>\r\n" +
            "<meta charset=\"UTF-8\">\r\n" +
            "<title>Test Server</title>\r\n" +
            "</head>\r\n" +
            "<body>\r\n" +
            "\r\n" +
            "<p><a href=\"InsertMySQL.html\">На главную
страницу</a></p>\r\n" +
            "</body>\r\n" +
```



```

        "</html>");
    out.println("<!DOCTYPE html>\r\n" +
        "<html>\r\n" +
        "<head>\r\n" +
        "<meta charset=\"UTF-8\">\r\n" +
        "<title>Test Server</title>\r\n" +
        "</head>\r\n" +
        "<body>\r\n" +
        "\r\n" +
        "<p><a href=\"SQLiteServlet\">Посмотреть
обновленную базу данных</a></p>\r\n" +
        "</body>\r\n" +
        "</html>");
    }
    else{
        out.println("Что-то пошло не так");
    }
    }
    catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

Создается переменная для проверки выполнения обновления базы данных, а также выводится надпись об успешном добавлении и двух гиперссылок на главную страницу и на просмотр обновленной базы данных.

## 2.2. Пример выполнения лабораторной работы

1. Запустите файл InsertMySQL.html на Apache Tomcat сервере (рисунок 9). В появившемся окне нажмите кнопку «Finish» после чего откроется веб-страница в браузере (рисунок 10).

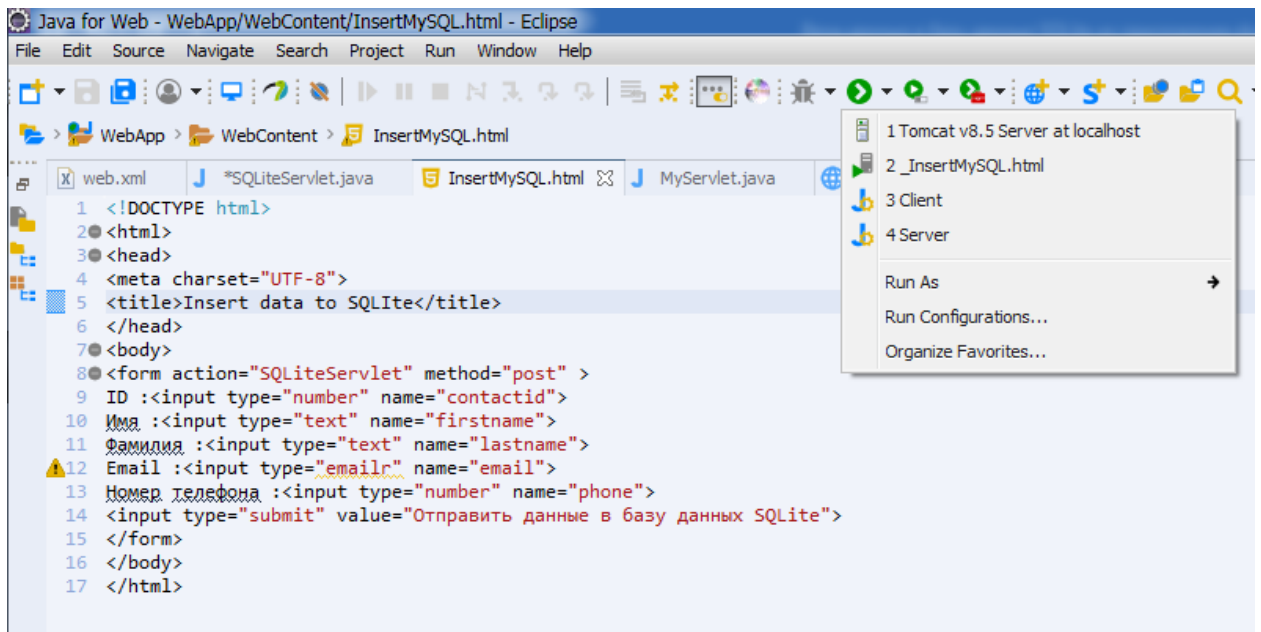


Рисунок 9 – Запуск InsertMySQL.html

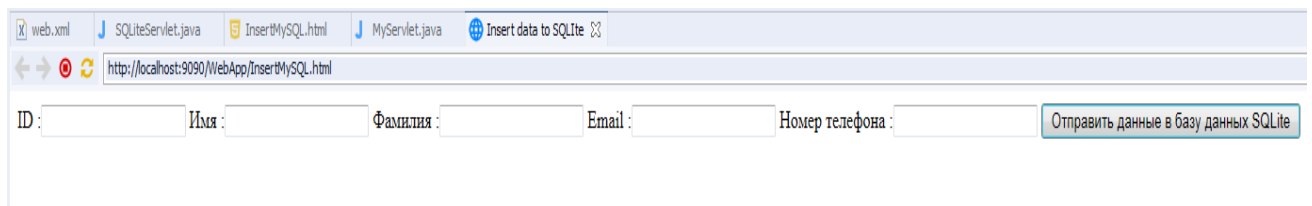


Рисунок 10 – InsertMySQL.html в браузере

2. Введите ID, Имя, Фамилию, Email и телефон пользователю которого необходимо добавить в базу данных, например, «2», «John», «Дое», «johndoe@crim.eu», «874512584123» (рисунок 11).

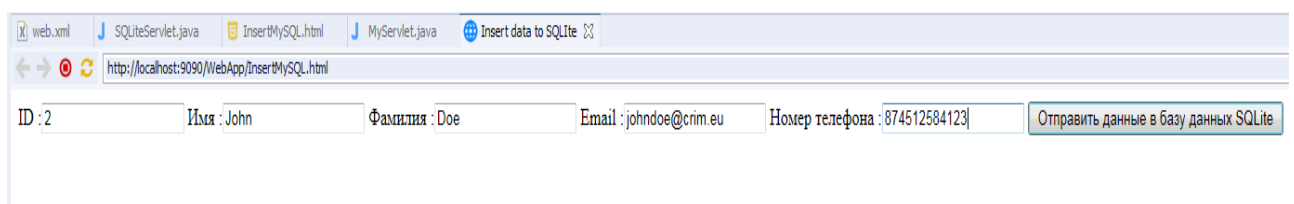


Рисунок 11 – Заполнение полей для внесения информации

3. Нажмите на исполняемую кнопку для внесения информации в базу данных, после чего будет выполнен переход на страницу с подтверждением об успешности внесения данных (рисунок 12).

Запись добавлена в базу данных

[На главную страницу](#)

[Посмотреть обновленную базу данных](#)

Рисунок 12 – Успешное добавление записи в базу данных

4. Для того чтобы посмотреть обновленную базу данных нажмите на гиперссылку «Посмотреть обновленную базу данных» (рисунок 13).

79996085964 874512584123

[На главную страницу](#)

Рисунок 13 – Обновленная база данных

### **3 Содержание отчета по лабораторной работе**

В сводный отчет по лабораторным работам в качестве одного из разделов или подразделов включаются скриншоты, показывающие создание необходимых файлов для внесения информации в базу данных и их использование.

#### **4 Вопросы к защите лабораторной работе**

1. Расскажите о возможных параметрах для поля «type»
2. Что такое web.xml и для чего он нужен?
3. Какова суть метода doPost?
4. За что отвечает блок <welcome-file-list>?
5. Если в блоке <welcome-file-list> имеется несколько файлов, какой файл будет открыт первым? И что будет происходить если первого файла не обнаружится?
6. Почему в SQL-запросе в качестве значений передаются «?»?
7. Что случится в SQL-запросе указать большее или меньшее количество «?»?