МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра ЭВМ

Отчет по предмету

Функциональное программирование

Лабораторная работа №5

**«**Работа с базой данных**»**

|  |  |
| --- | --- |
| **Выполнили:**  студенты группы 150501  Смоленский Н.О.  Петруненко В.Е. | **Проверила:**  Герман Ю.О. |

Минск

2023

**Цель работы**: Изучить технику работы с базой данных в Scala.

**Теоретические сведения:**

Итак, создаем базу данных MySQL

Откройте и выберите MySQL Command line

Введите ваш пароль (пароль вы создали при установке MySQL). Если пароль правильный, система заработает. Вводите команды MySQL в консоли и ставите в конце точку с запятой. Нажимаете ENTER.

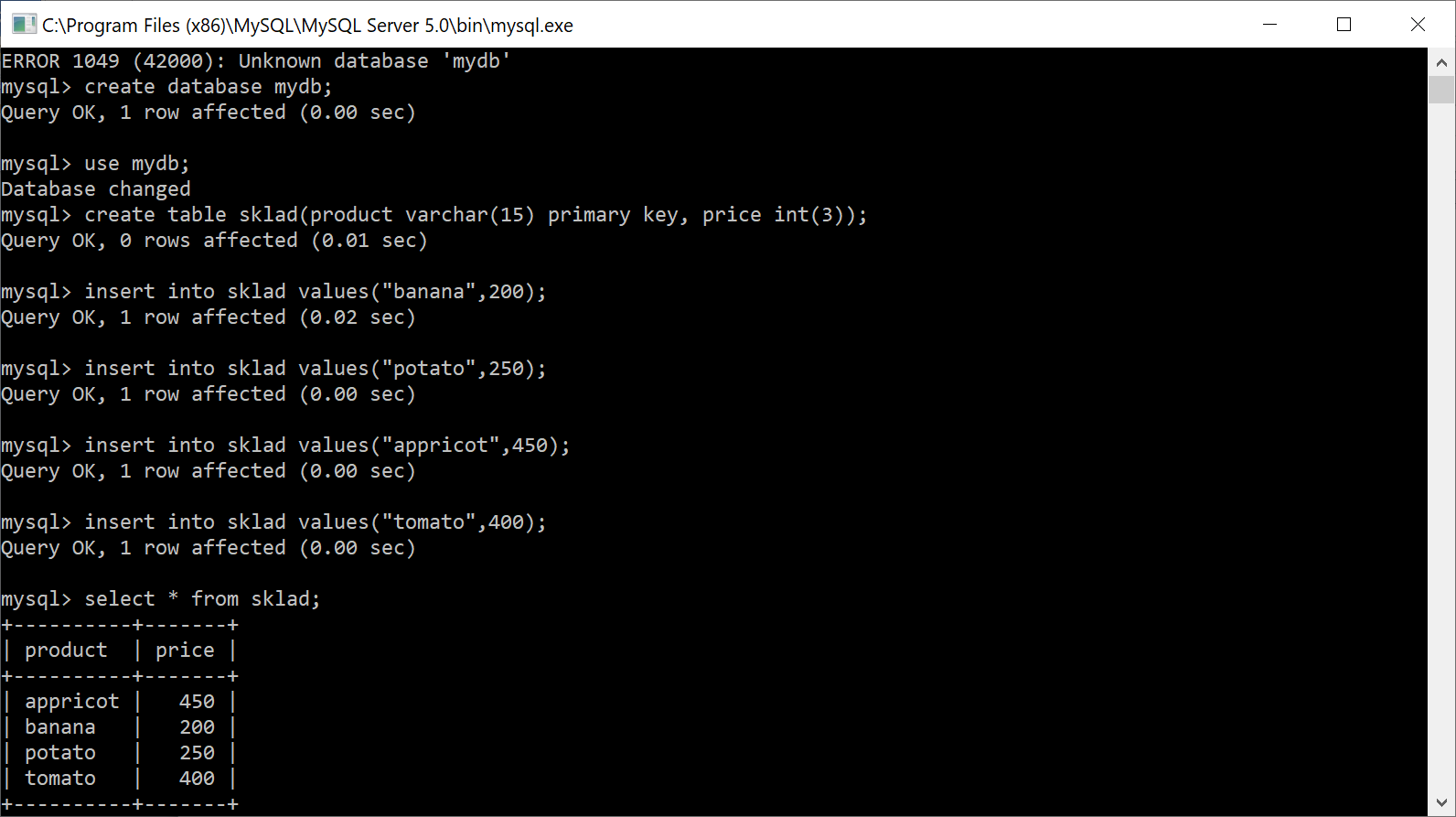
Так, создаете базу командой

* create database mydb;

База имеет имя mydb. Если база создана, она будет храниться на диске. Подключаете ее командой

* use mydb;

Потом создаете простую таблицу (у меня sklad) с двумя столбцами - product и price. Смотрите на экран консоли вверху – там все видно.



Добавляем в проект object (правая кнопка мыши на узле ButtonPr) и пишем под него код (при создании модуля пишем его имя ButtonModule)

**import** java.awt.event.{ActionEvent, ActionListener}

**import** javax.swing.{JButton, JFrame, JPanel, SwingUtilities}

**object** ButtonModule {

**def** main(args: Array[*String*]): Unit = {

SwingUtilities.invokeLater(() => {

**val** frame = **new** JFrame("My Application")

frame.setSize(300, 200)

frame.setLocationRelativeTo(**null**)

**val** panel = **new** JPanel()

**val** button = **new** JButton("Click me!")

panel.add(button)

button.addActionListener(**new** ActionListener {

**override** **def** actionPerformed(e: ActionEvent): Unit = {

println("Hello, world!")

}

})

frame.add(panel)

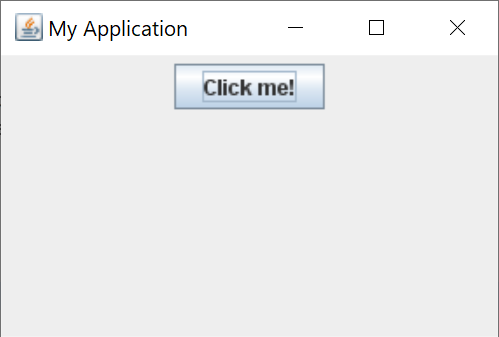
frame.setVisible(**true**)

})

}

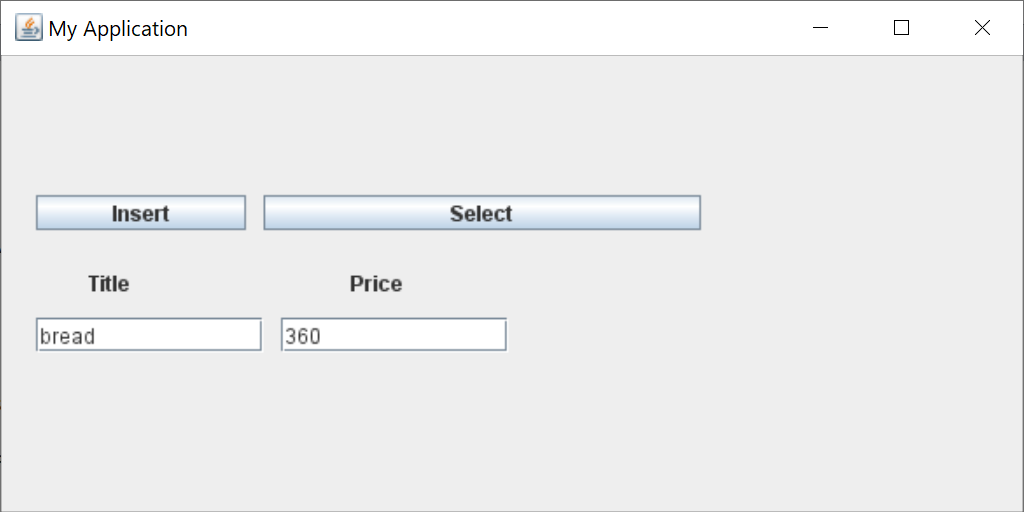
}

На выполнение (правая кнопка на узле проекта) – Run As



При нажатии на кнопку выводится строка в консольном окне среды Eclipse

Мы построим следующий экран



С помощью кнопки Insert добавляем запись в таблицу базы данных, с помощью кнопки Select – выводим содержимое. Чтобы соединиться с базой, нам нужен один импорт.

**import** java.sql.{Connection, DriverManager, ResultSet}

Но нам нужно подключить jar – файл com\_mysql.jar.

Наше приложение таково

**import** java.awt.event.{ActionEvent, ActionListener}

**import** javax.swing.{JButton, JFrame, JPanel, SwingUtilities}

**import** java.awt.Dimension

**import** javax.swing.{JFrame, JLabel,JTextField, JPanel, SwingUtilities}

**import** java.sql.{Connection, DriverManager, ResultSet}

**import** java.awt.Color

**object** ButtonModule {

**val** frame = **new** JFrame("My Application")

frame.setSize(800, 600)

frame.setLayout(**null**)

**def** main(args: Array[*String*]): Unit = {

SwingUtilities.invokeLater(() => {

// val panel = new JPanel()

**val** label1=**new** JLabel("Title")

**val** label2=**new** JLabel("Price")

label1.setBounds(50,120,100,20)

label2.setBounds(200,120,70,20)

**val** button1 = **new** JButton("Insert")

**val** button2 = **new** JButton("Select")

**val** textField = **new** JTextField(20)

**val** textField2 = **new** JTextField(20)

button1.setBounds(20,80,120,20)

button2.setBounds(150,80,250,20)

textField.setBounds(20,150,130,20)

textField2.setBounds(160,150,130,20)

frame.add(button1)

frame.add(button2)

frame.add(label1)

frame.add(textField)

frame.add(label2)

frame.add(textField2)

button1.addActionListener(**new** ActionListener {

**override** **def** actionPerformed(e: ActionEvent): Unit = {

**val** url = "jdbc:mysql://localhost:3306/mydb"

**val** username = "root"

**val** password = "1"

Class.forName("com.mysql.jdbc.Driver")

// Class.forName("org.gjt.mm.mysql.Driver")

**val** conn = DriverManager.getConnection(url, username, password)

**try** {

**val** stmt = conn.createStatement()

**val** rs = stmt.execute("INSERT INTO sclad VALUES ('" + textField.getText + "'," + textField2.getText + ")")

textField2.setText("")

textField.setText("Added record")

}

**finally** {

conn.close()

}

// textField.setText("problems")

}

})

button2.addActionListener(**new** ActionListener {

**override** **def** actionPerformed(e: ActionEvent): Unit = {

////////

{

**val** url = "jdbc:mysql://localhost:3306/mydb"

**val** username = "root"

**val** password = "1"

Class.forName("com.mysql.jdbc.Driver")

**val** conn = DriverManager.getConnection(url, username, password)

**try** {

**val** stmt = conn.createStatement()

**val** prod\_name= textField.getText().toString().trim()

**val** rs = stmt.executeQuery("SELECT \* FROM sclad WHERE product = '"+prod\_name+"'")

**while** (rs.next()) {

**val** name = rs.getString("product")

**val** price = rs.getInt("price")

// println(s"name=$name, price=$price")

textField2.setText(""+price)

}

} **finally** {

conn.close()

}

}

///////

// textField.setText("Hello World, You say")

}

})

frame.setBackground(Color.BLUE)

frame.setLocationRelativeTo(**null**)

frame.setVisible(**true**)

})

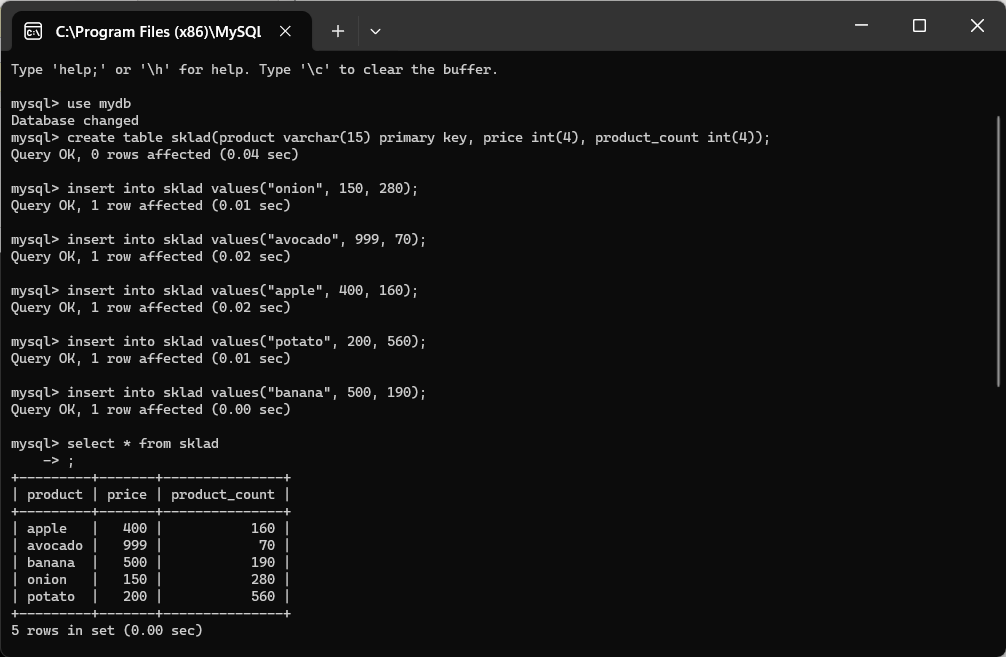
}

}

Разберитесь с этим приложением.

**Код программы:**

1.Расширить количество столбцов таблицы sclad, добавив поле количество товара на складе.



2.Вывести товар, дающий максимальную прибыль (количество\*цену).

buttonProfit.addActionListener(**new** ActionListener{

**override** **def** actionPerformed(e: ActionEvent): Unit = {

**val** url = "jdbc:mysql://localhost:3306/mydb"

**val** username = "root"

**val** password = "root"

Class.forName("com.mysql.jdbc.Driver")

**val** conn = DriverManager.getConnection(url, username, password)

**try** {

**val** stmt = conn.createStatement()

**val** rs = stmt.executeQuery("SELECT \* FROM sklad ORDER BY (product\_count \* price) DESC LIMIT 1")

**val** resultText = **new** *StringBuilder*()

**while** (rs.next()){

**val** name = rs.getString("product")

**val** price = rs.getInt("price")

**val** count = rs.getInt("product\_count")

**val** maxProfit = price \* count

resultText.append(s"Product: $name \nPrice: $price \nCount: $count \nMax Profit: $price \* $count = $maxProfit\n")

}

textFieldPrice.setText("")

textFieldProduct.setText("")

textFieldCount.setText("")

**if** (resultText.isEmpty)

resultText.append(s"Error: \nDatabase table is empty")

outputArea.setText(resultText.toString())

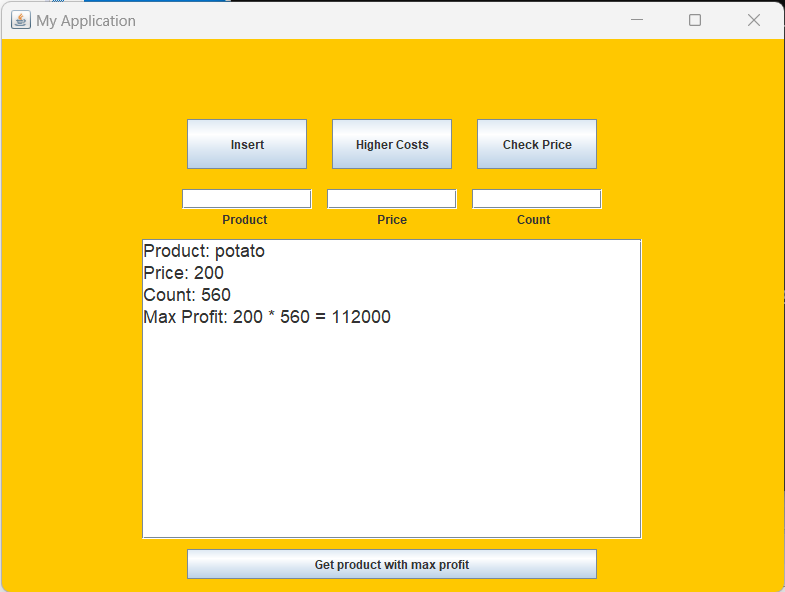
} **finally** {

conn.close()

}

}

})



3.Выбрать товар по ограничению (не меньше такой-то величины).

buttonHigher.addActionListener(**new** ActionListener{

**override** **def** actionPerformed(e: ActionEvent): Unit = {

**if** (textFieldPrice.getText.isEmpty) {

outputArea.setText("Error: \nPrice field is empty")

**return**

}

**val** url = "jdbc:mysql://localhost:3306/mydb"

**val** username = "root"

**val** password = "root"

Class.forName("com.mysql.jdbc.Driver")

**val** conn = DriverManager.getConnection(url, username, password)

**try** {

**val** stmt = conn.createStatement()

**val** lowPrice = textFieldPrice.getText().toInt

**val** query = "SELECT \* FROM sklad WHERE price >= ?"

**val** pstmt = conn.prepareStatement(query)

pstmt.setInt(1, lowPrice)

**val** rs = pstmt.executeQuery()

**val** resultText = **new** *StringBuilder*()

**while** (rs.next()){

**val** name = rs.getString("product")

**val** price = rs.getInt("price")

**val** count = rs.getInt("product\_count")

resultText.append(s"Product: $name \nPrice: $price \nCount: $count\n\n")

}

outputArea.setText(resultText.toString())

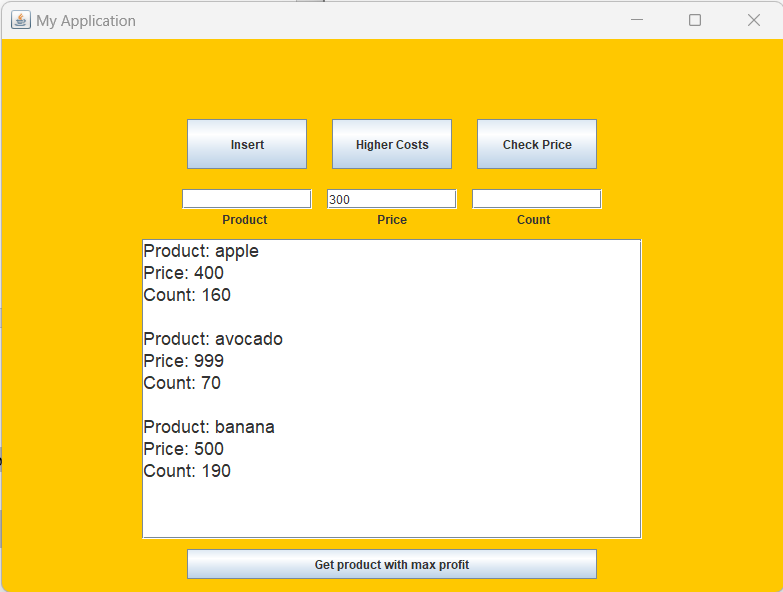
} **finally** {

conn.close()

}

}

})

****

**Вывод**

В ходе выполнения лабораторной работы было проведено ознакомление с работой с базами данных в Scala, различными SQL-запросами. Используя полученную теоретическую информацию, была написана программа, использующая графический интерфейс.