Министерство образования Республики Беларусь

Учреждение высшего образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ

по лабораторной работе №5

на тему

**Работа с базой данных**

|  |  |
| --- | --- |
| Студенты группы 150504  Проверила | Шевцов В.С  Крук А.И  Герман Ю.О. |

Минск 2023

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

**Краткие теоретические сведения.**

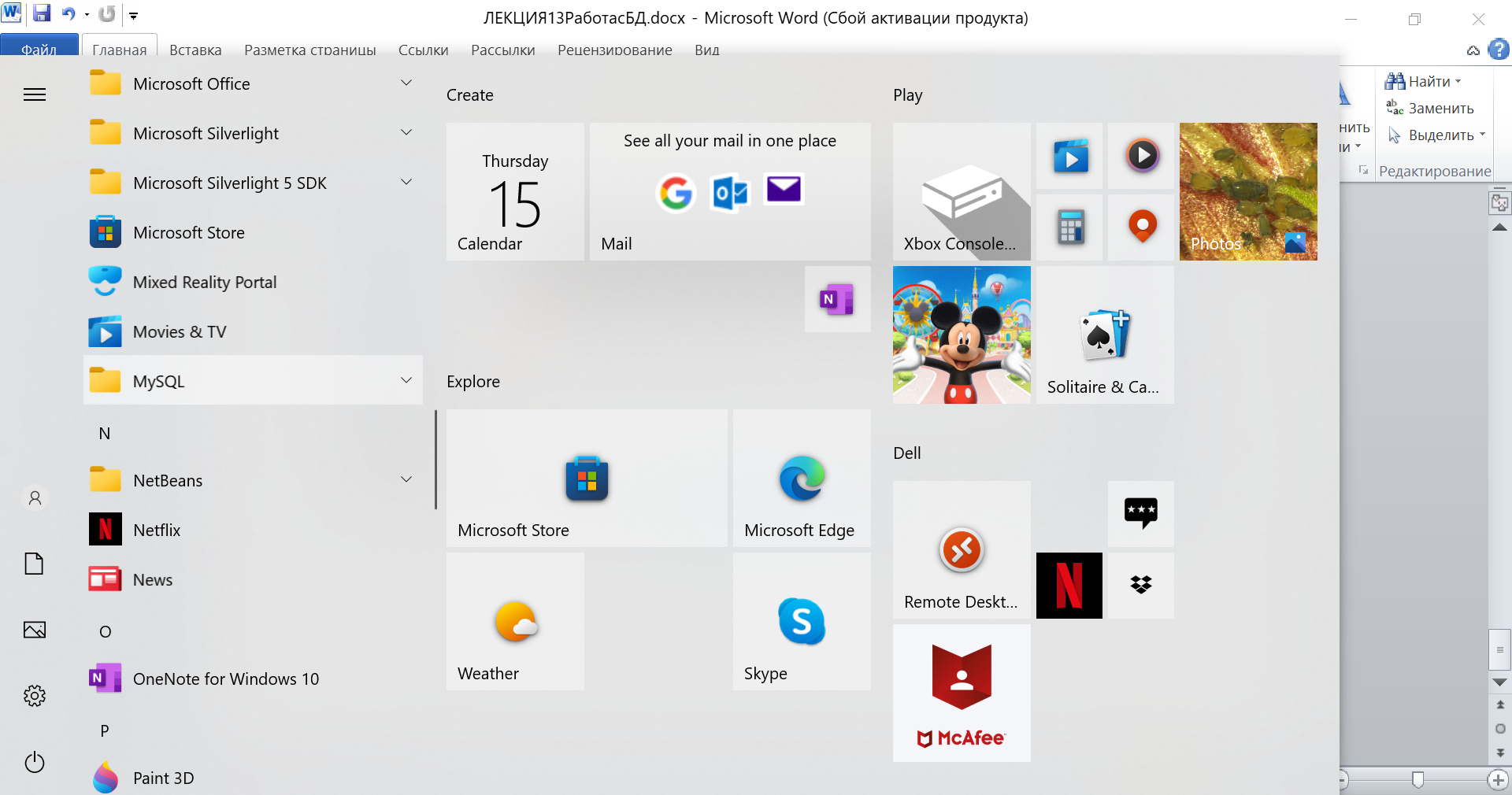
Установить MySql. Возникают вопросы о версии. Лучше использовать устойчивую версию MySql 5.0.

Если использовать другие новейшие версии, будут трудности. Нужно пройтись по инсталляции. В списке инсталлированного ПО видим

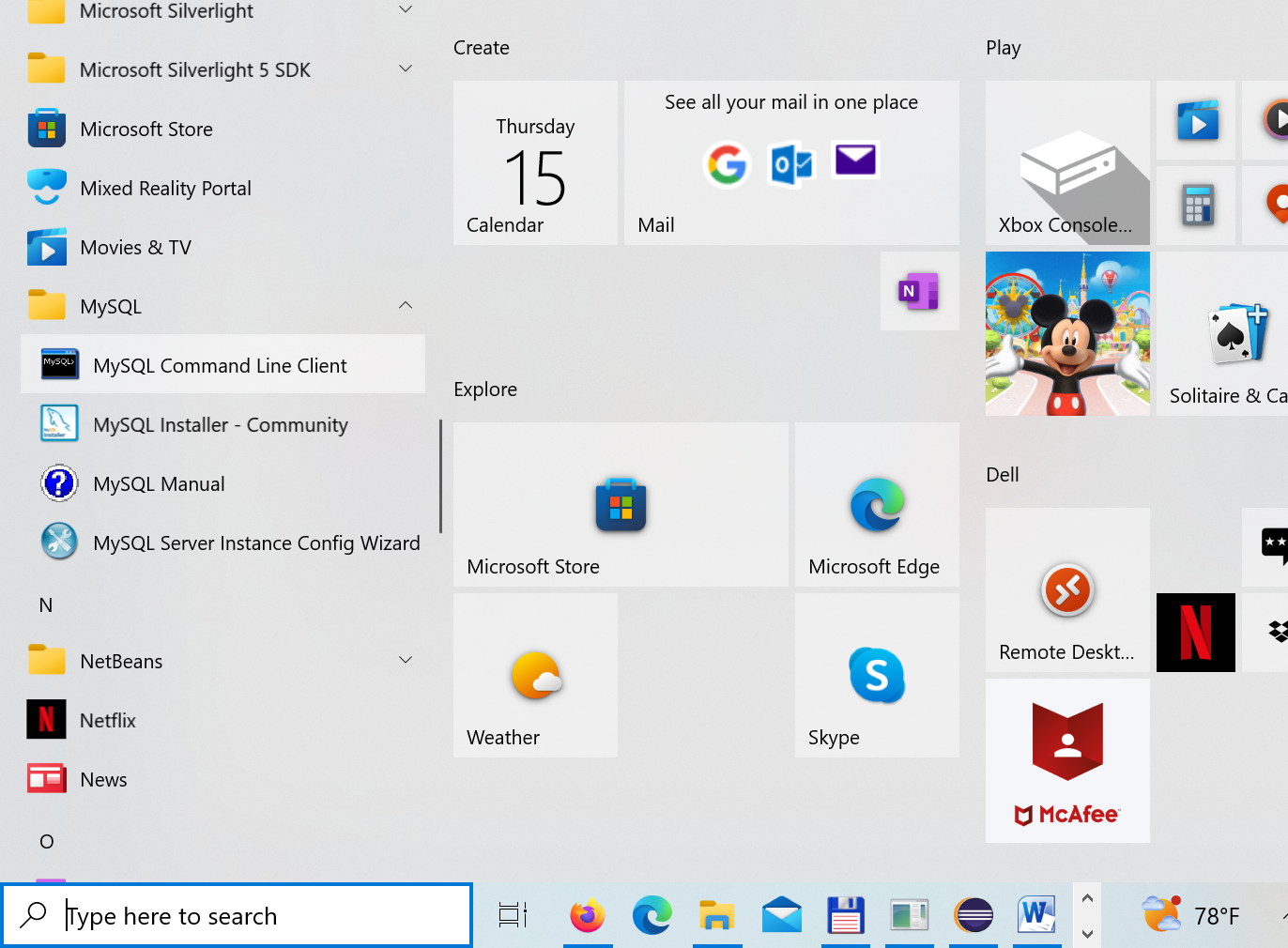
Нужен еще MySQL Connector Java – его нужно отдельно закачать и распаковать.

Итак, создаем базу данных MySQL

Вы видите результат в процессе. Нужно запустить MySQL. Если вы его поставили, то откройте окно пуска Windows, найдите 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.

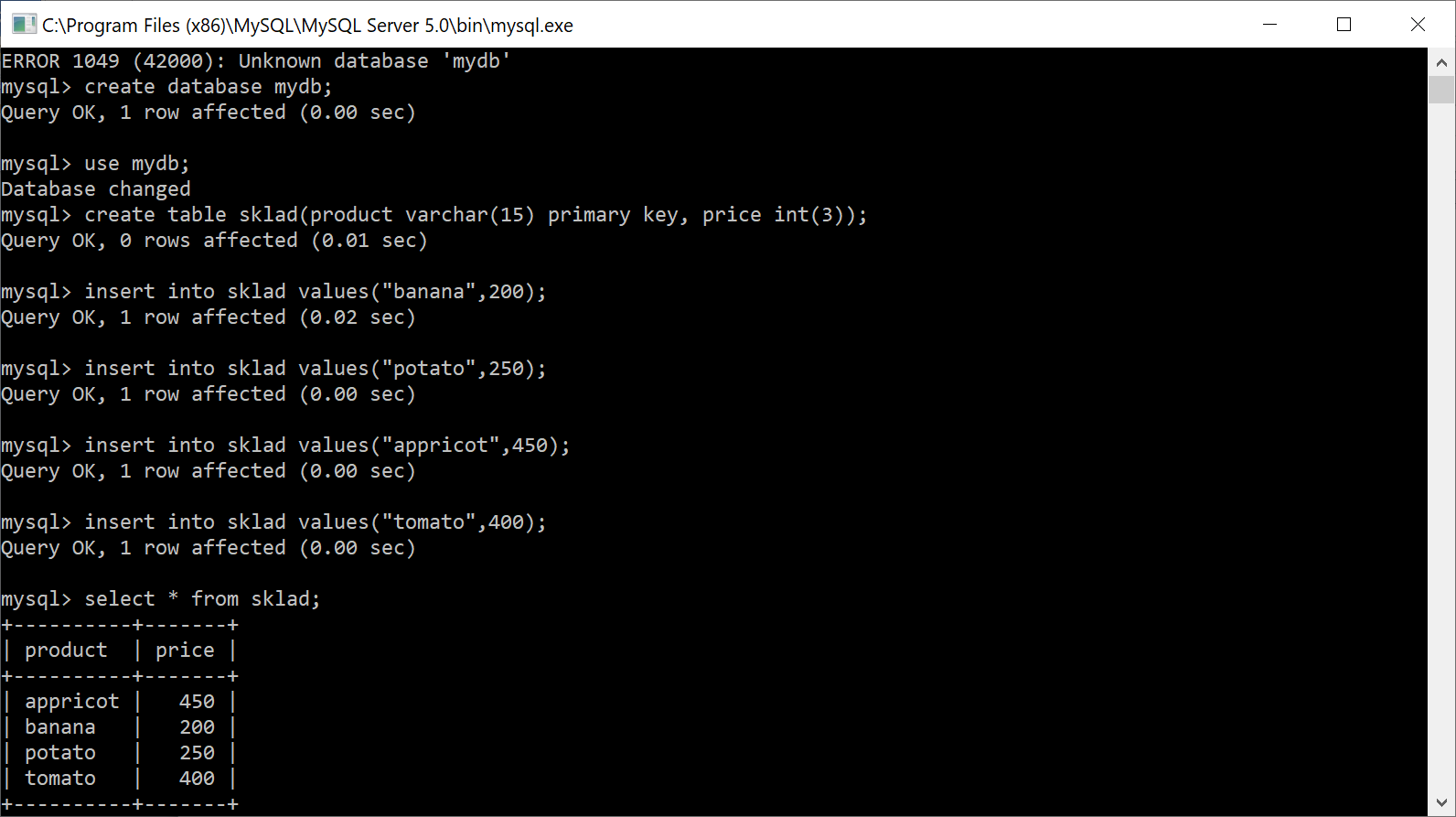
Так, создаете базу командой

* create database mydb;

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

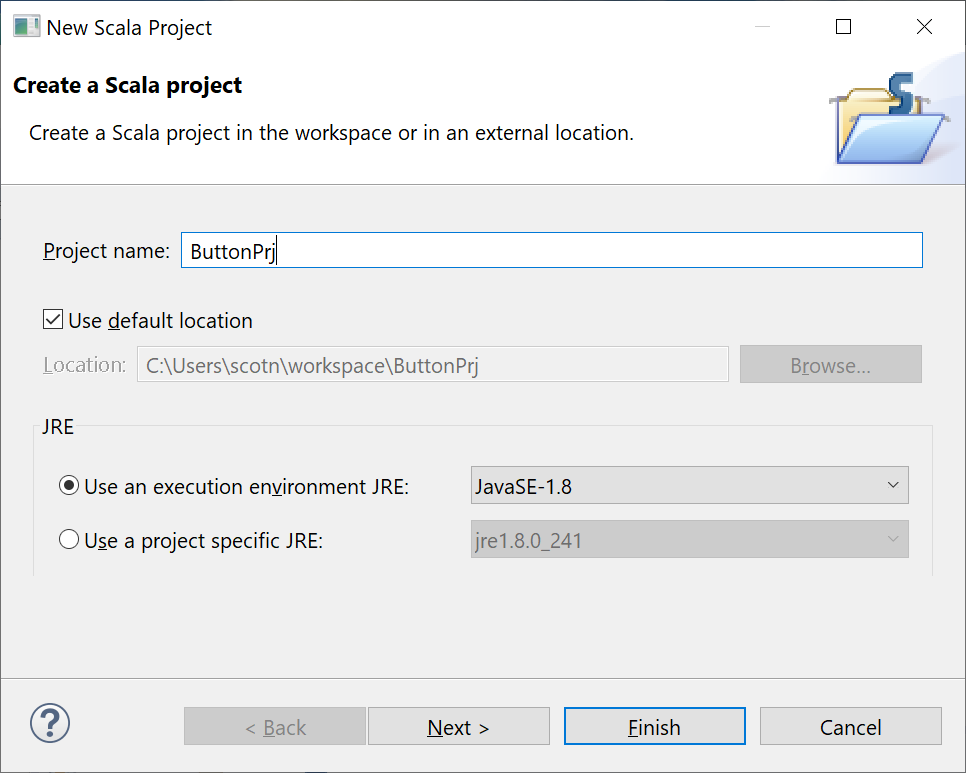
* use mydb;

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

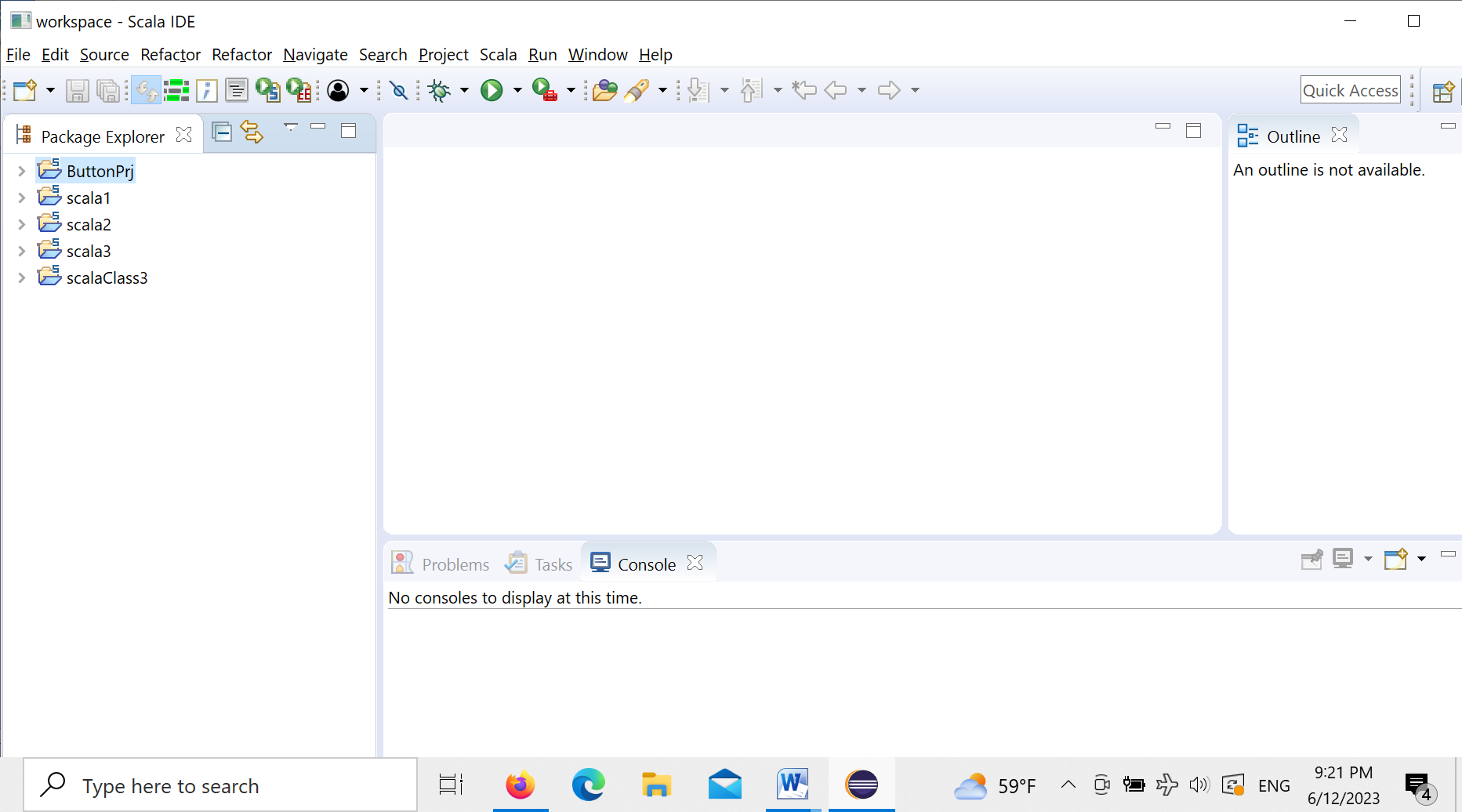


Теперь Scala. MySQL работает. Запускаем Eclipse Oxigen.

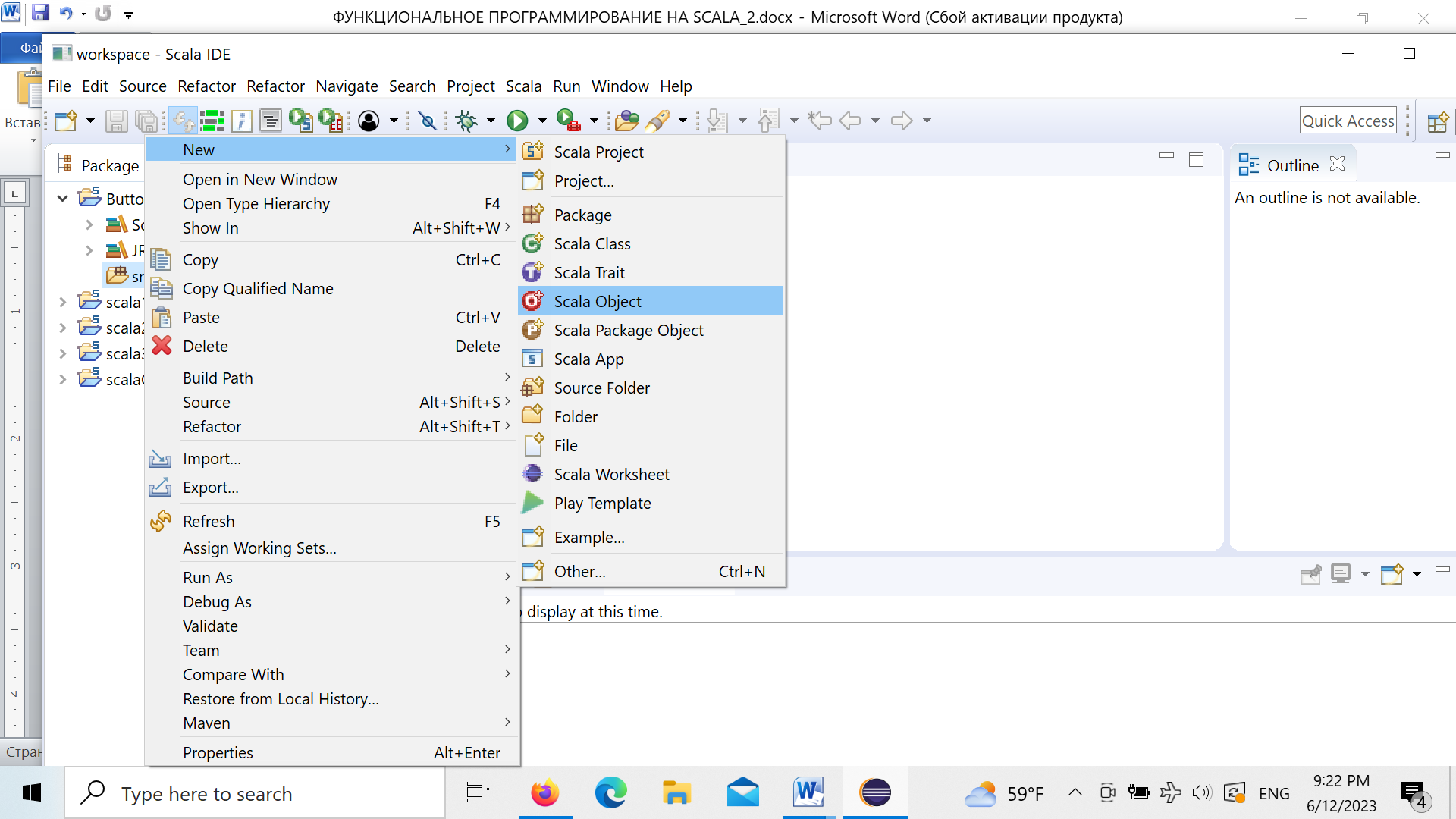
Выбираем File – New Project



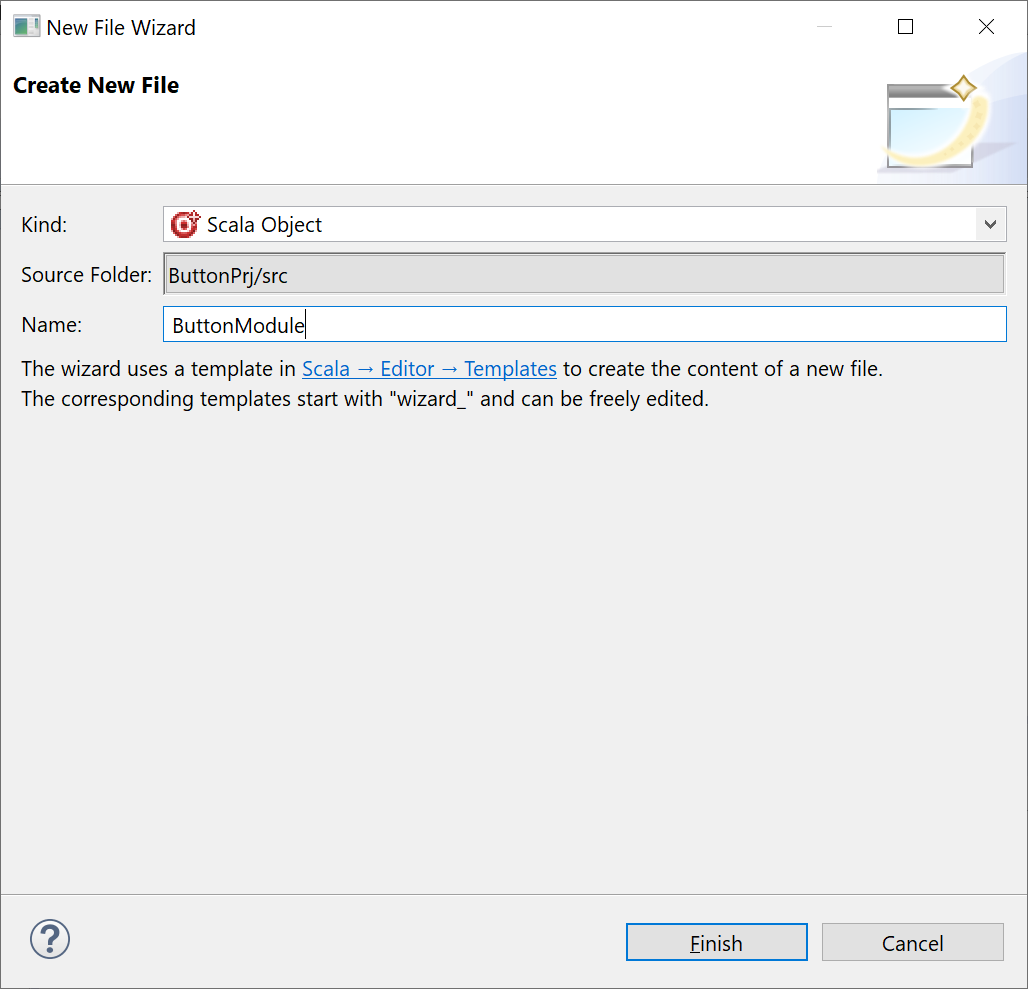
Создается пустой проект



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)

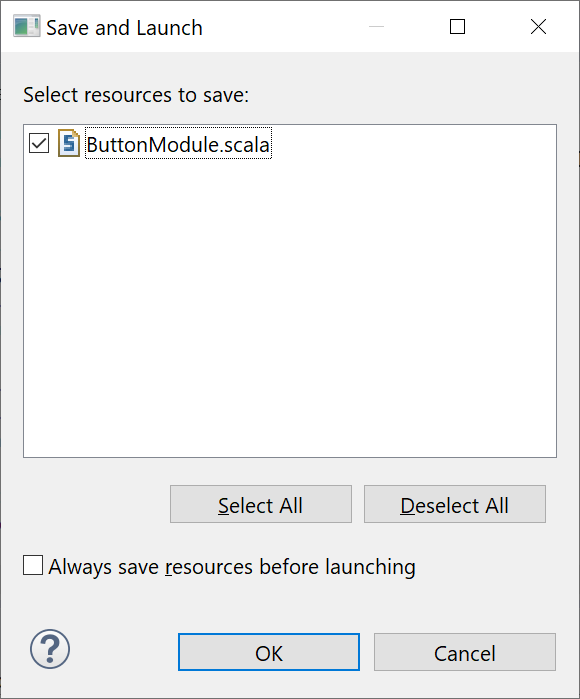
frame.setVisible(true)

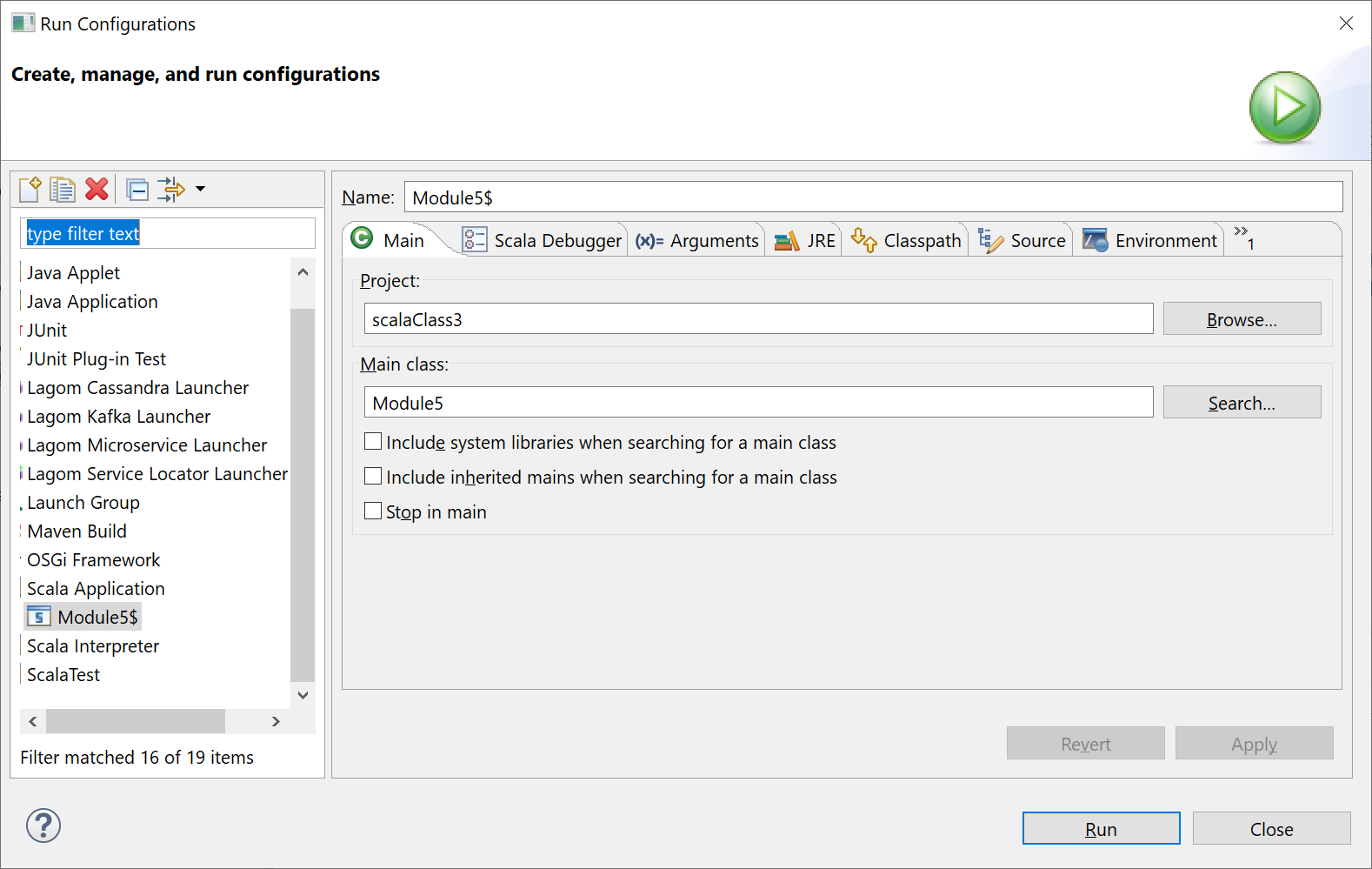
})

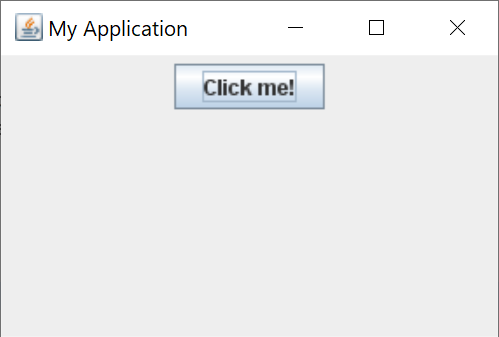
}

}

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

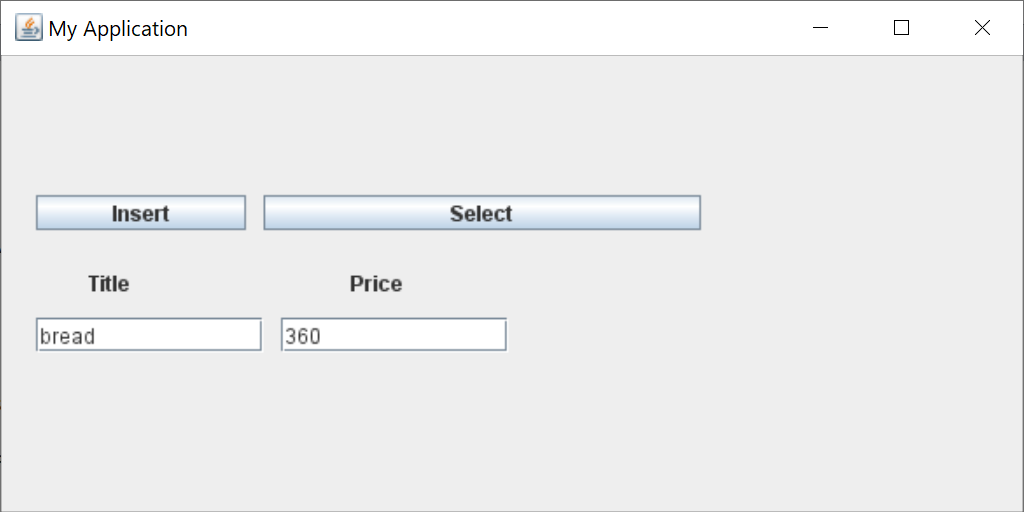






При нажатии на кнопку выводится строка в консольном окне среды Eclipse

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}

Итак, подготовились. MySQL скачали и установили. Версия старая, но проверенная (это версия MySQL 5.0). Создали базу данных и таблицу. MySQL работает запускаем Eclipse Oxigen. Создаем проект. Добавляем объект и пишем такой код

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)

})

}

}

***Задание***

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

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

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

import com.mysql.jdbc.exceptions.MySQLIntegrityConstraintViolationException

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

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

import java.awt.Dimension

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)

frame.getContentPane().setBackground(Color.CYAN)

frame.setLocationRelativeTo(null)

frame.setVisible(true)

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

SwingUtilities.invokeLater(() => {

// val panel = new JPanel()

val label1 = new JLabel("Title")

val label2 = new JLabel("Price")

val amountLabel = new JLabel("Amount")

label1.setBounds(235, 120, 100, 20)

label2.setBounds(375, 120, 100, 20)

amountLabel.setBounds(510, 120, 100, 20)

val setButton = new JButton("Insert")

val selectButton = new JButton("Select")

val groupButton = new JButton("Group by product")

val likeButton = new JButton("Select like")

val nameField = new JTextField(20)

val priceField = new JTextField(20)

val amountField = new JTextField(20)

val infoField = new JTextArea()

setButton.setBackground(Color.BLUE)

setButton.setForeground(Color.PINK)

setButton.setBounds(105, 80, 120, 20)

selectButton.setBounds(255, 80, 120, 20)

groupButton.setBounds(405, 80, 150, 20)

likeButton.setBounds(585, 80, 150, 20)

nameField.setBounds(200, 150, 100, 20)

priceField.setBounds(340, 150, 100, 20)

amountField.setBounds(480, 150, 100, 20)

infoField.setBounds(200, 200, 400, 200)

frame.add(setButton)

frame.add(selectButton)

frame.add(label1)

frame.add(nameField)

frame.add(label2)

frame.add(priceField)

frame.add(amountLabel)

frame.add(amountField)

frame.add(infoField)

frame.add(groupButton)

frame.add(likeButton)

setButton.addActionListener(new ActionListener {

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

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

val username = "root"

val password = "root"

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 podval VALUES ('" + nameField.getText + "'," + priceField.getText + "," + amountField.getText + ")")

priceField.setText("")

nameField.setText("")

amountField.setText("")

} finally {

conn.close()

}

}

}

)

selectButton.addActionListener(new ActionListener {

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

////////

{

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

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 prod\_name = nameField.getText().trim()

var rs: java.sql.ResultSet = null

if (!prod\_name.isBlank && priceField.getText.isBlank && amountField.getText.isBlank) {

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

while (rs.next()) {

val name = rs.getString("product")

val price = rs.getInt("price")

val amount = rs.getInt("amount")

priceField.setText("" + price)

amountField.setText("" + amount)

}

}

if (prod\_name.isBlank && !priceField.getText.isBlank && amountField.getText.isBlank) {

rs = stmt.executeQuery("SELECT ALL product, price, amount FROM podval WHERE price > " + priceField.getText + " GROUP BY price, product")

infoField.setText("product\tprice\tamount\n")

while (rs.next()) {

val name = rs.getString("product")

val price = rs.getString("price")

val amount = rs.getString("amount")

infoField.setText(infoField.getText + name + "\t" + price + "\t" + amount + "\n")

}

}

if (prod\_name.isBlank && priceField.getText.isBlank && amountField.getText.isBlank) {

var maxPrice = 0

rs = stmt.executeQuery("SELECT ALL product, price, amount FROM podval")

while (rs.next()) {

val name = rs.getString("product")

val price = rs.getString("price")

val amount = rs.getString("amount")

if (Integer.parseInt(price) \* Integer.parseInt(amount) > maxPrice) {

maxPrice = Integer.parseInt(price) \* Integer.parseInt(amount)

nameField.setText(name)

priceField.setText(price)

amountField.setText(amount)

}

}

}

}

finally {

conn.close()

}

}

}

}

)

groupButton.addActionListener(new ActionListener {

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

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

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 prod\_name = nameField.getText().trim()

var rs: java.sql.ResultSet = null

rs = stmt.executeQuery("SELECT product, price, amount FROM podval2 where product = '" + prod\_name + "'" + "group by product, id")

infoField.setText("product\tprice\tamount\n")

while (rs.next()) {

val name = rs.getString("product")

val price = rs.getInt("price")

val amount = rs.getInt("amount")

infoField.setText(infoField.getText + name + "\t" + price + "\t" + amount + "\n")

}

} finally {

conn.close()

}

}

}

)

likeButton.addActionListener(new ActionListener {

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

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

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 prod\_name = nameField.getText().trim()

var rs: java.sql.ResultSet = null

rs = stmt.executeQuery("SELECT \* FROM podval WHERE product LIKE '" + prod\_name + "%'")

infoField.setText("product\tprice\tamount\n")

while (rs.next()) {

val name = rs.getString("product")

val price = rs.getInt("price")

val amount = rs.getInt("amount")

infoField.setText(infoField.getText + name + "\t" + price + "\t" + amount + "\n")

}

} finally {

conn.close()

}

}

}

)

}

)

}

}

