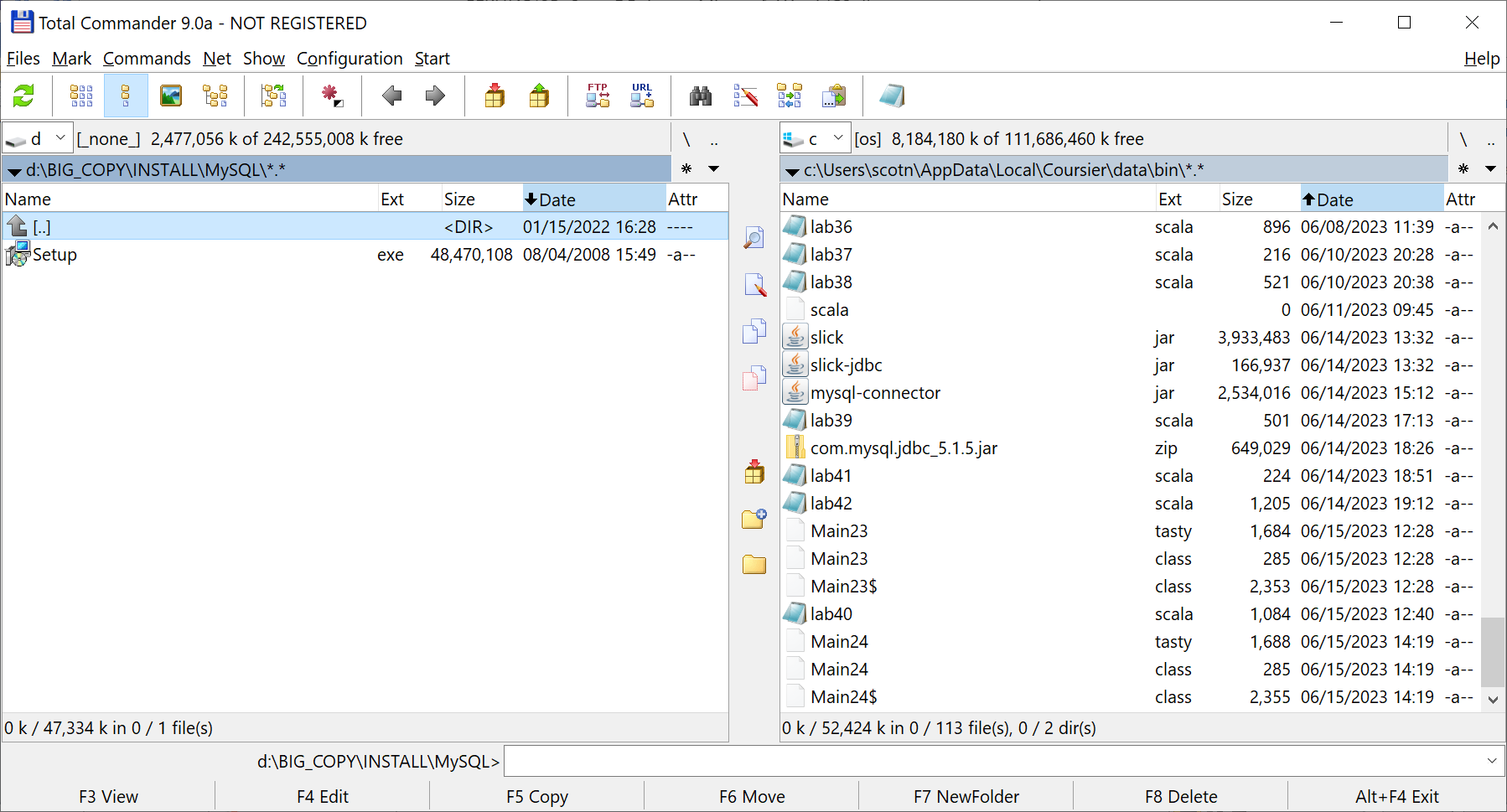
**ЛАБОРАТОРНАЯ РАБОТА 5. Работа с базой данных**

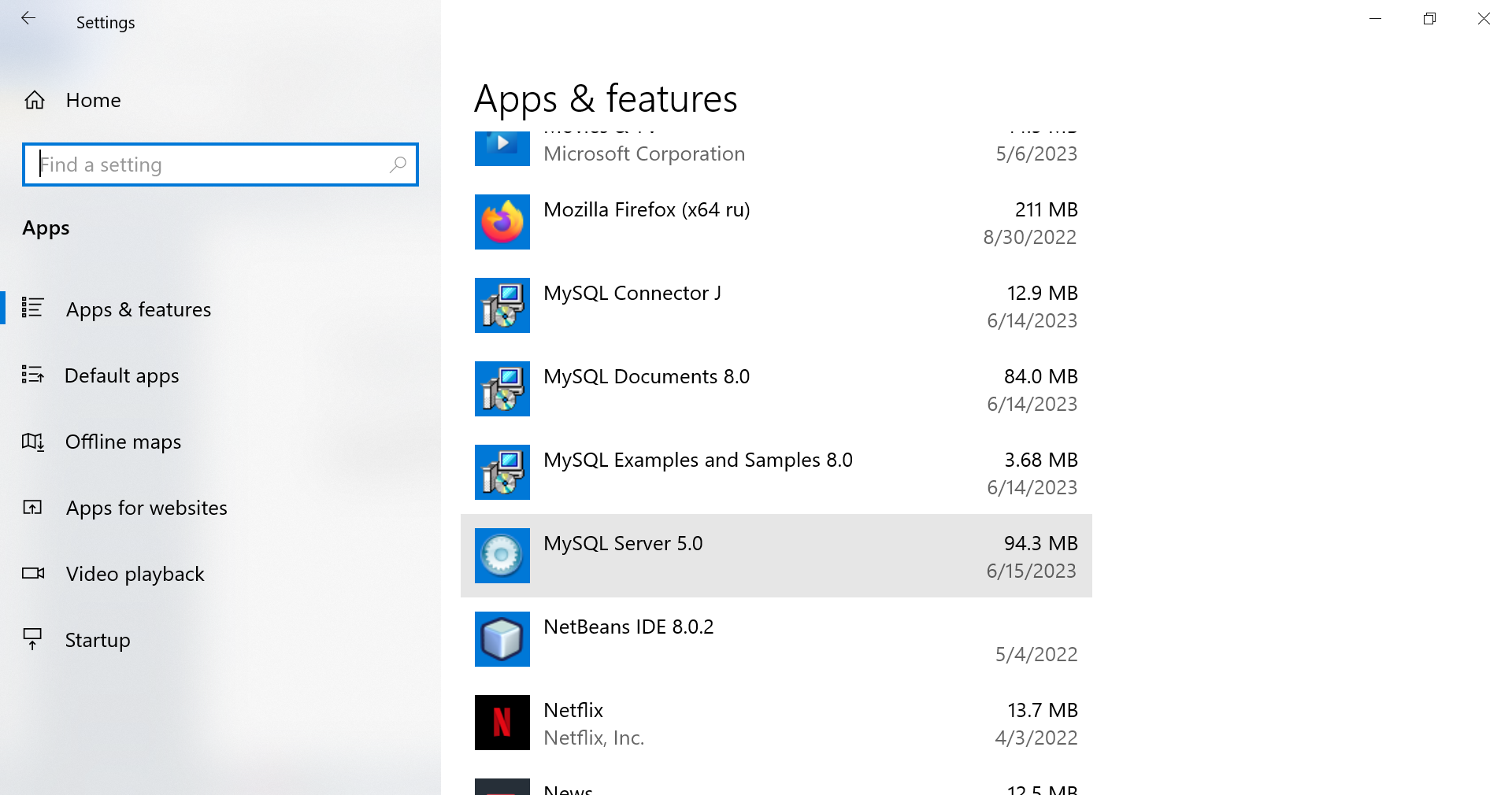
**Цель**. Изучить технику работы с базой данных в Scala.

Краткие теоретические сведения.

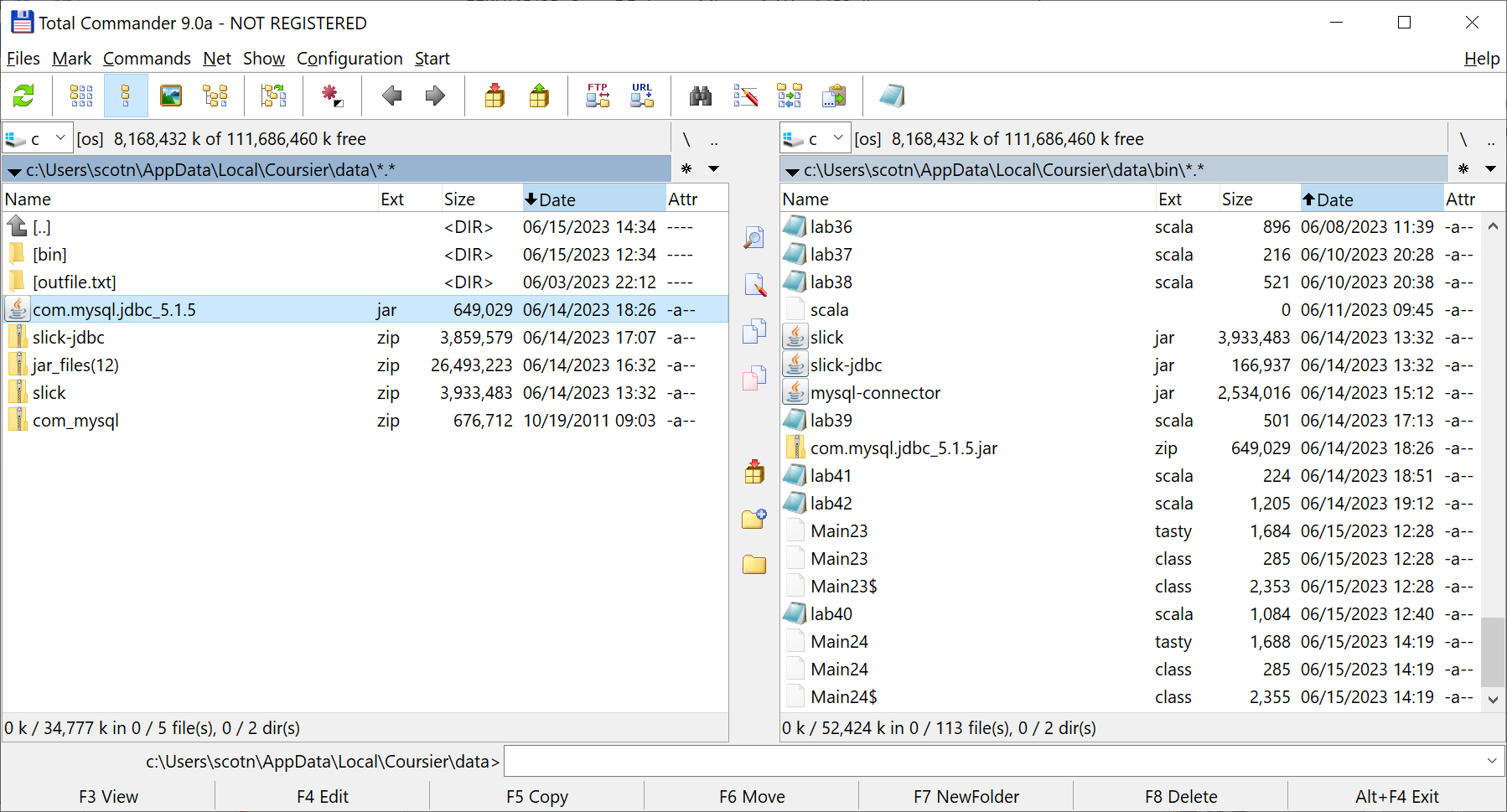
Установить MySql. Возникают вопросы о версии. Лучше использовать устойчивую версию MySql 5.0.



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. Впрочем, это не обязательно. Вам точно понадобится jar-файл с драйвером для MySQL. Вот он



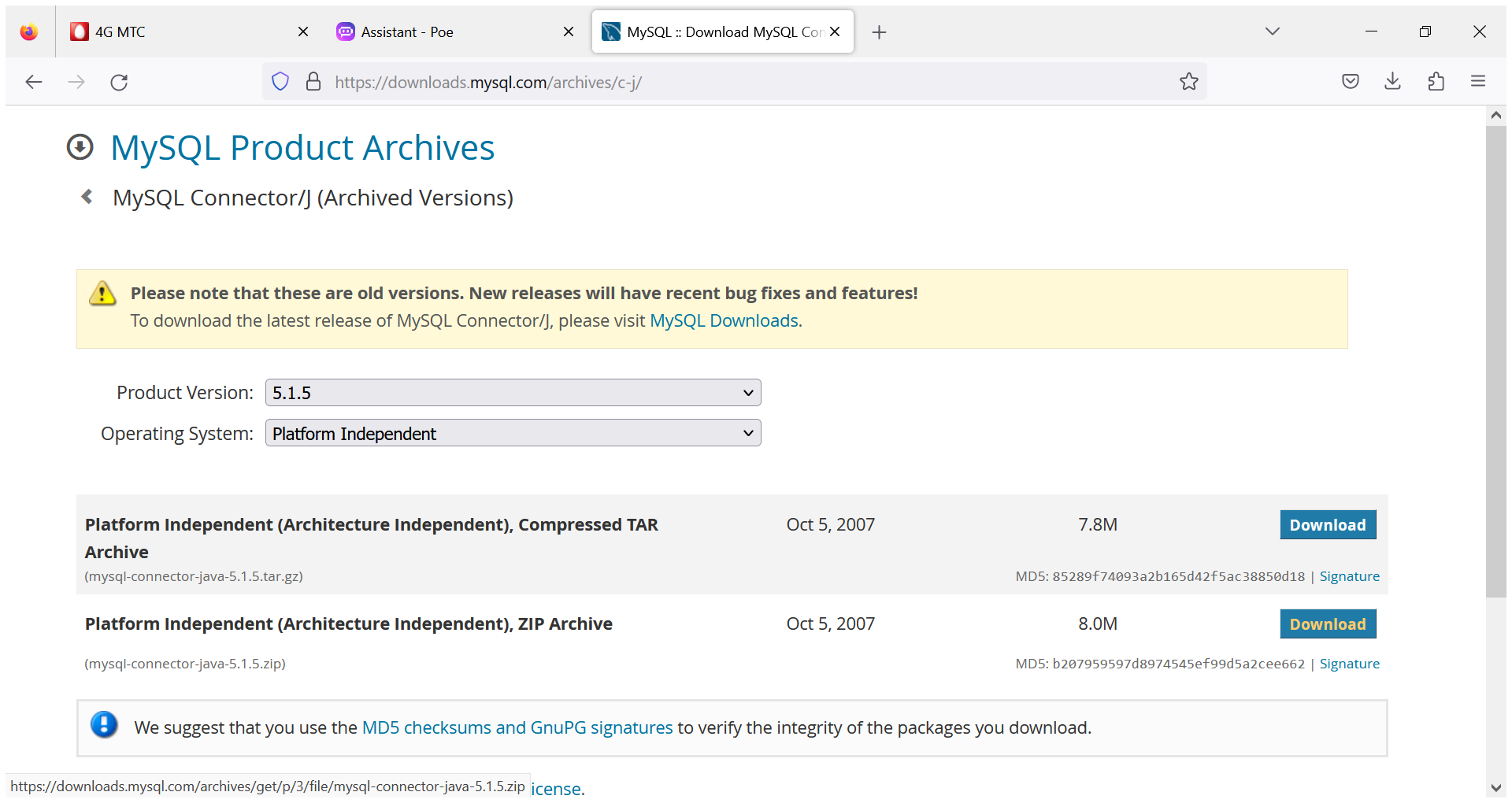
Можно его переименовать в com\_mysql для краткости (необязательно).

Вот шаги для установки коннектора MySQL Java с помощью установщика MSI:

1. Перейдите на страницу загрузки MySQL Connector/J:

https://dev.mysql.com/downloads/connector/j/

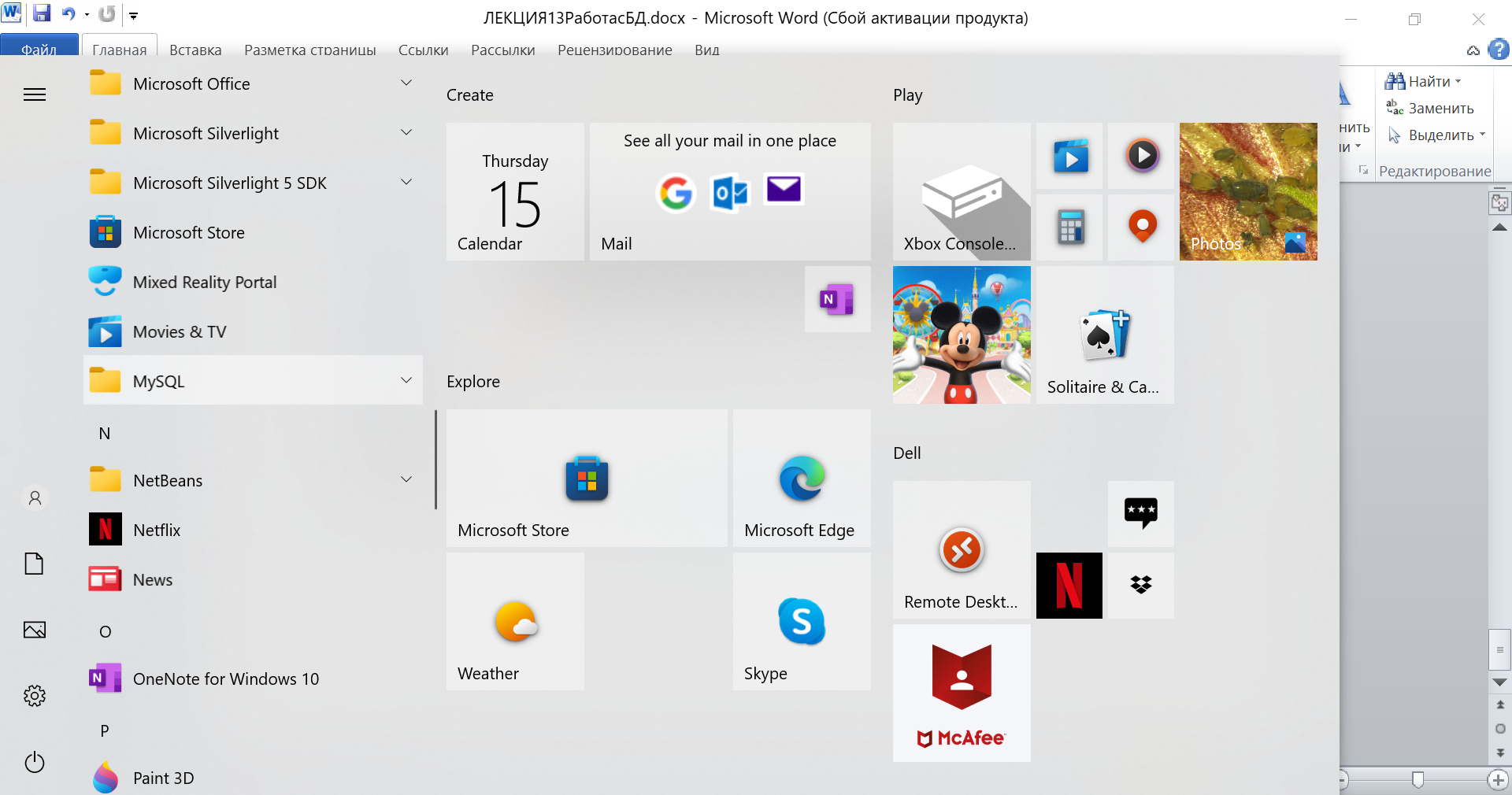
Скачайте версию из архива 5.1.5



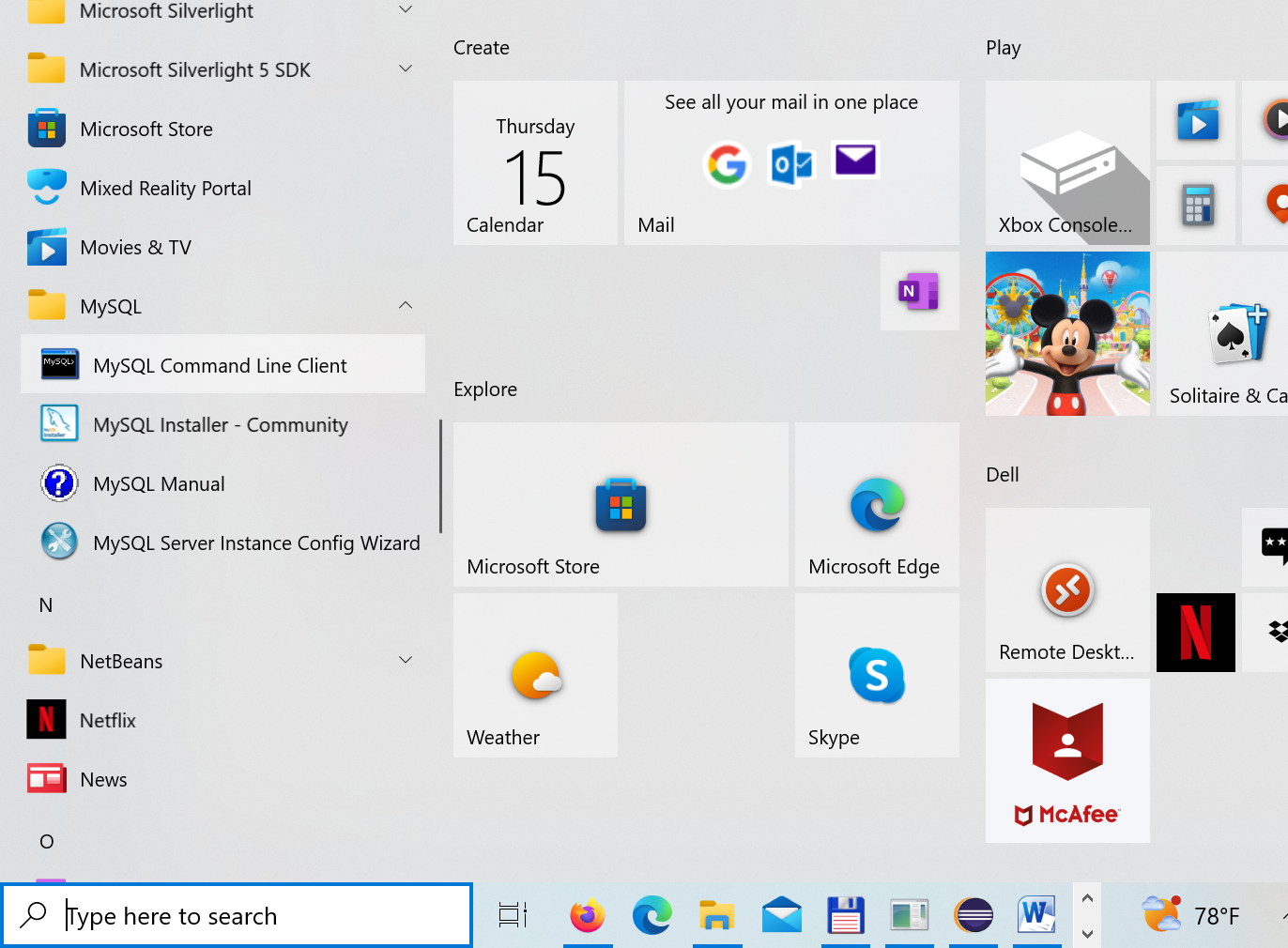
2. Скопируйте архив zip в папку с MySQL и распакуйте.

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.

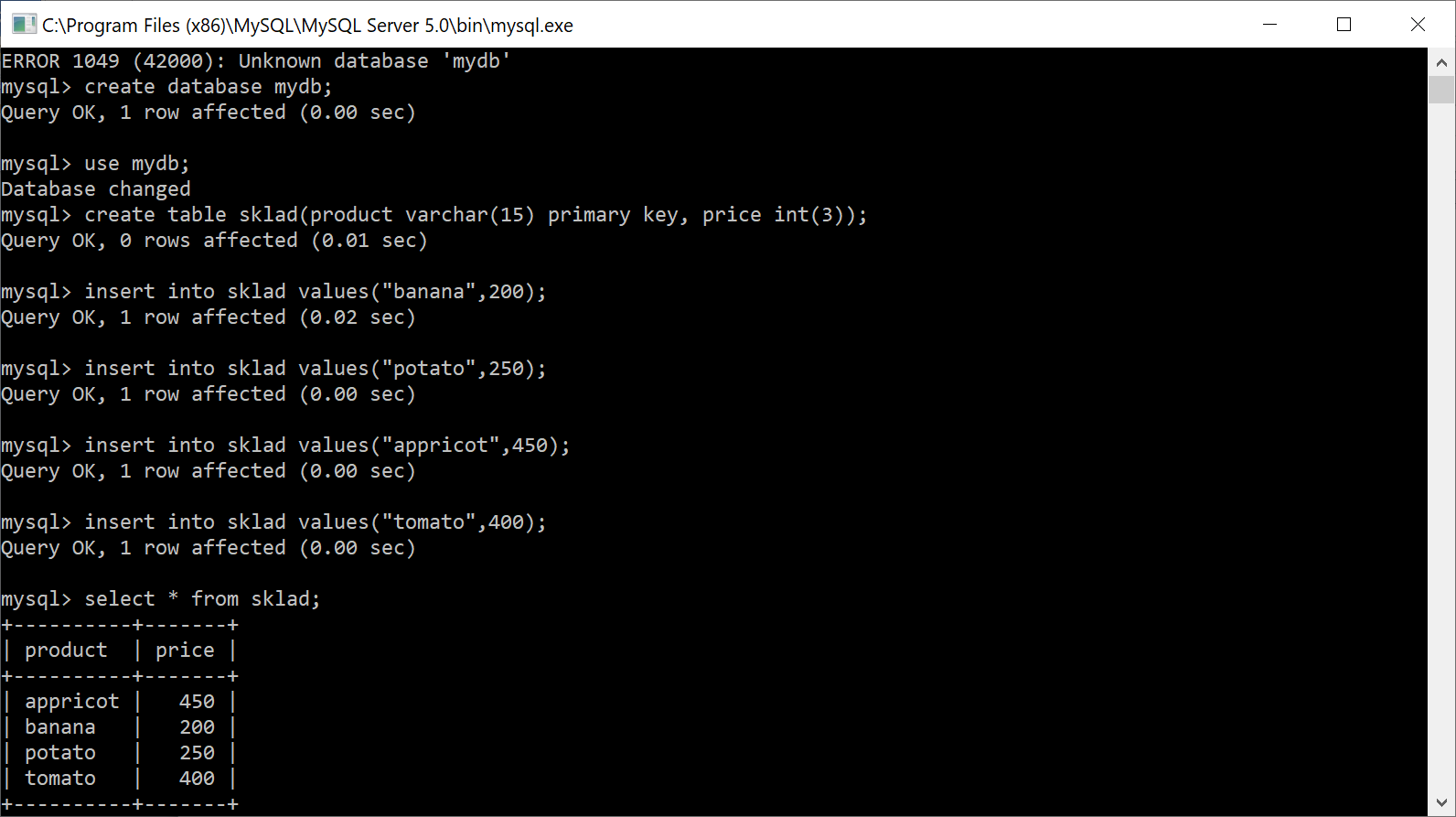
Так, создаете базу командой

* create database mydb;

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

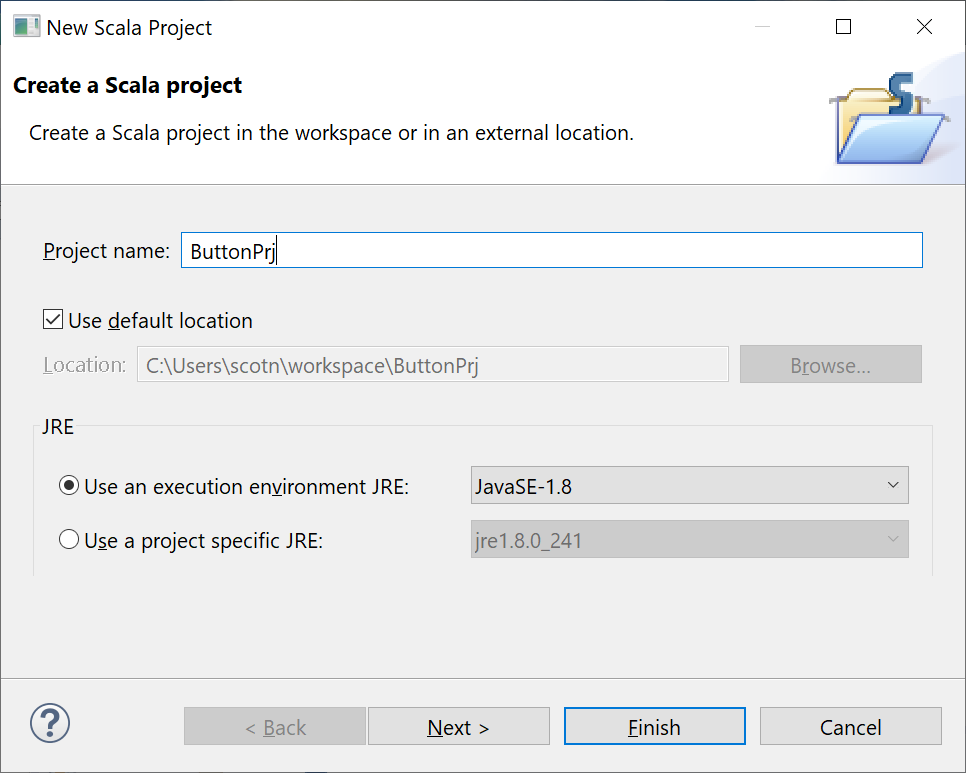
* use mydb;

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

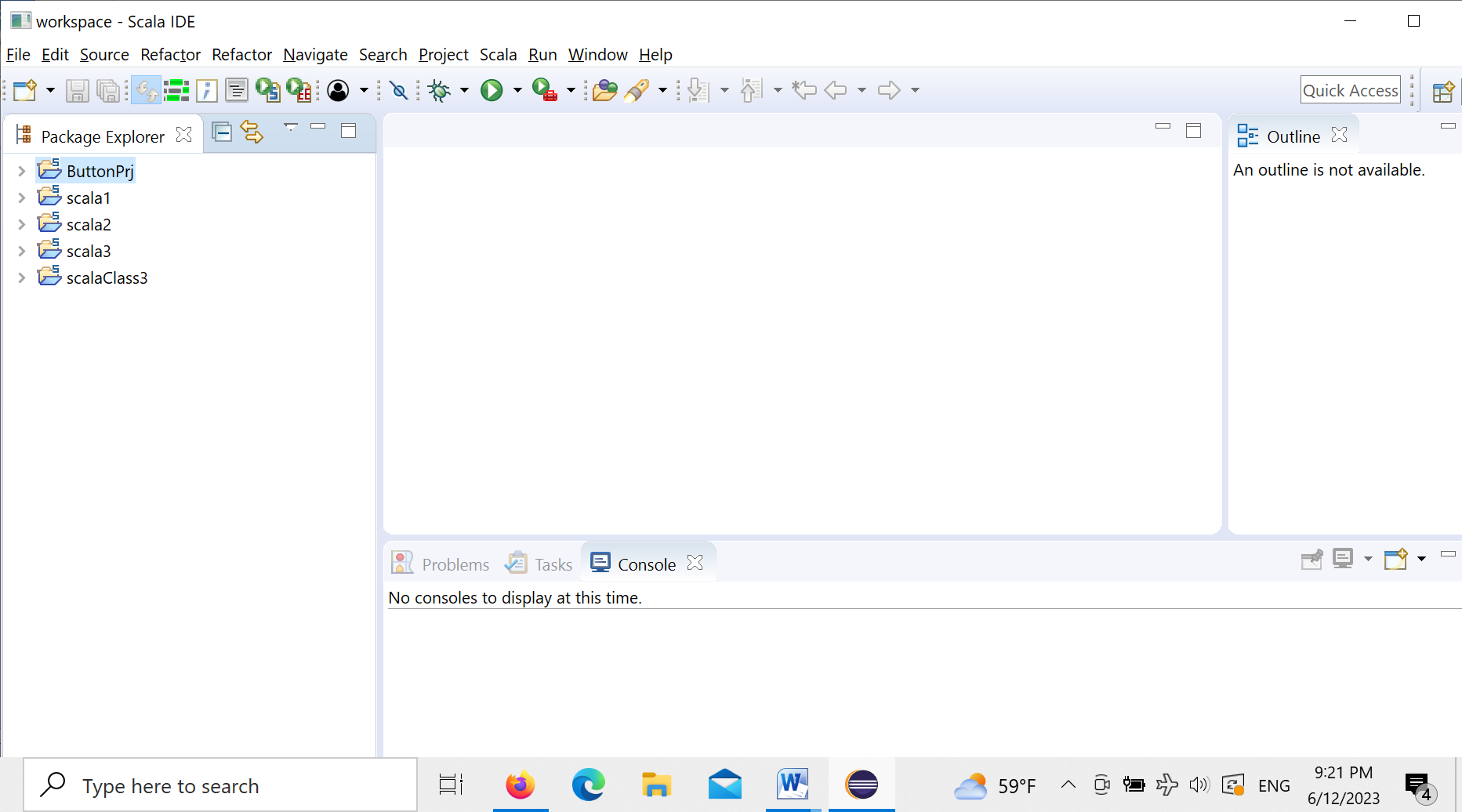


Теперь Scala. MySQL работает. Запускаем Eclipse Oxigen.

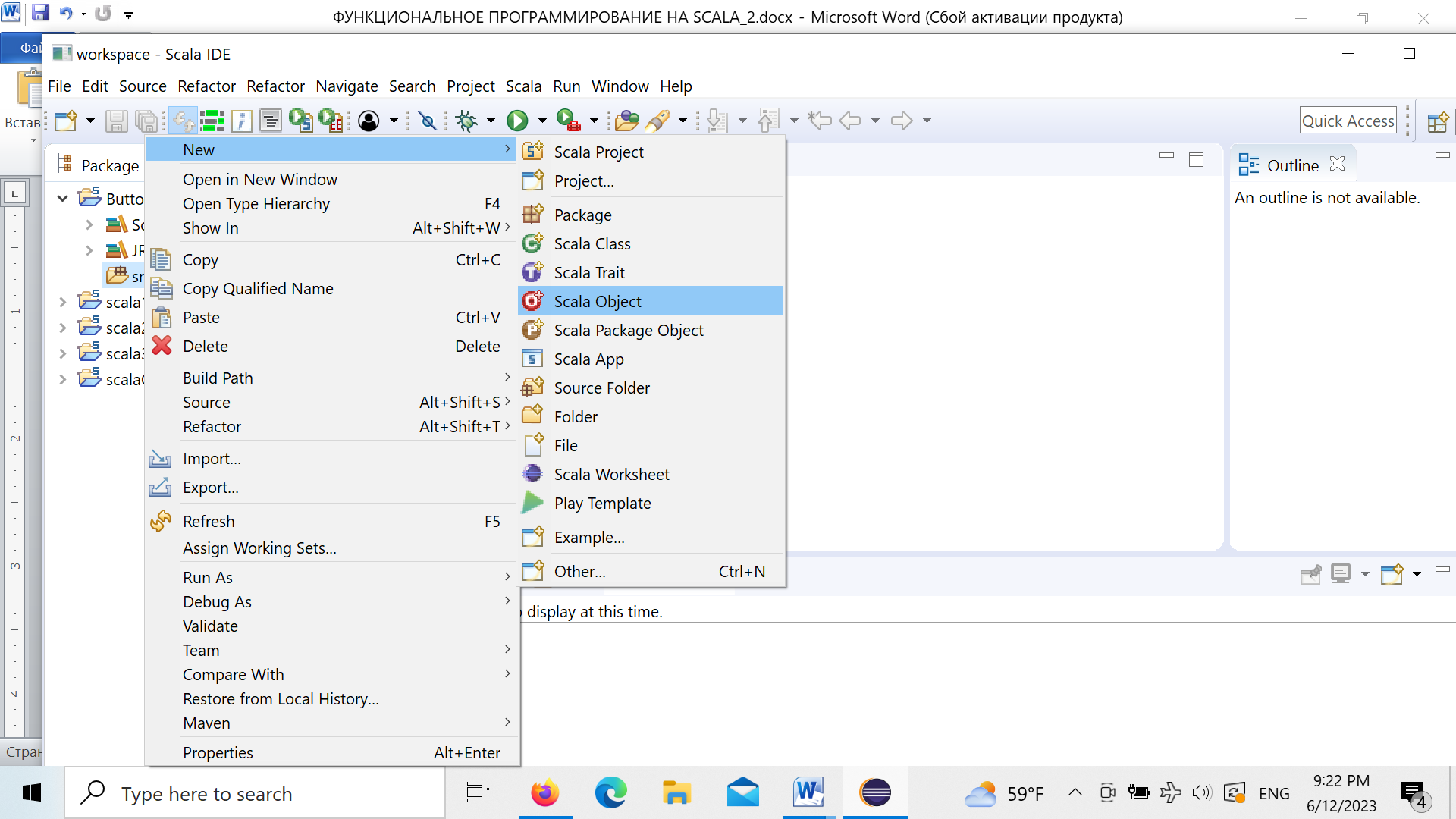
Выбираем File – New Project



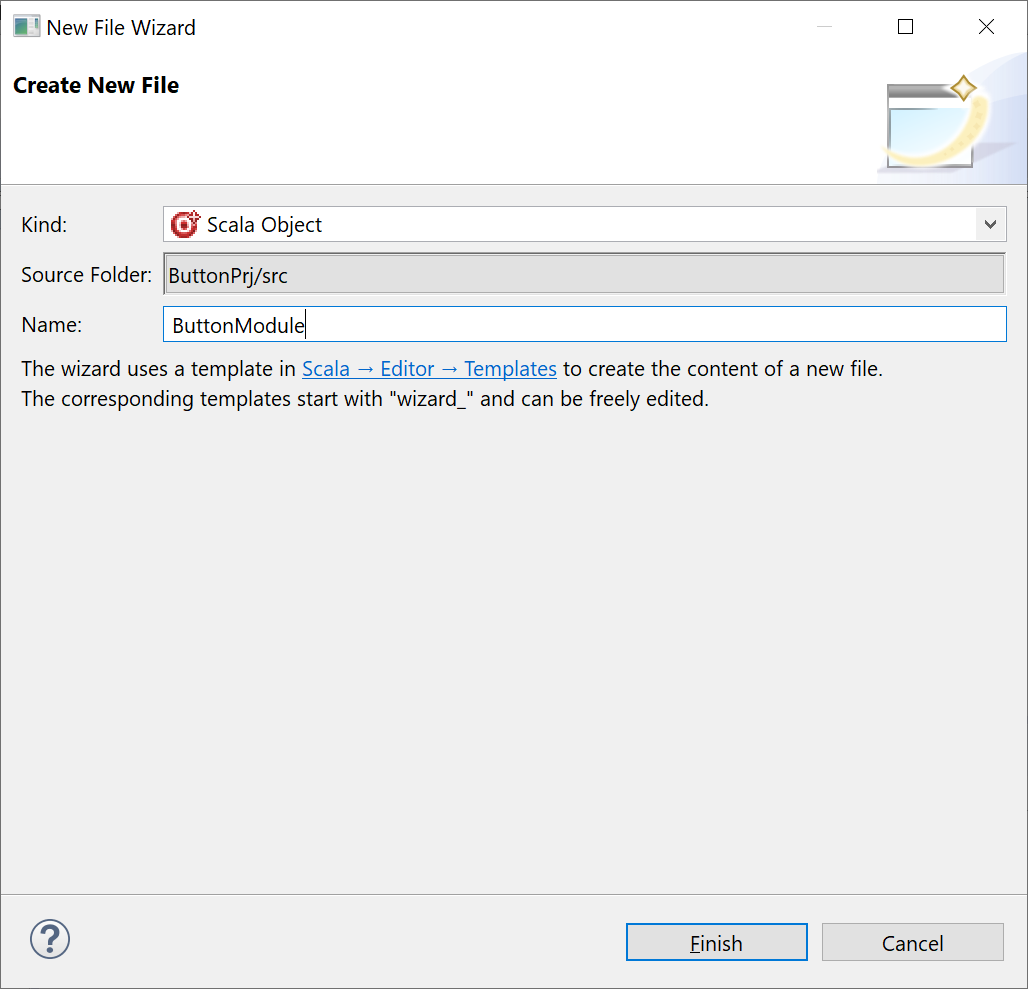
Создается пустой проект



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)

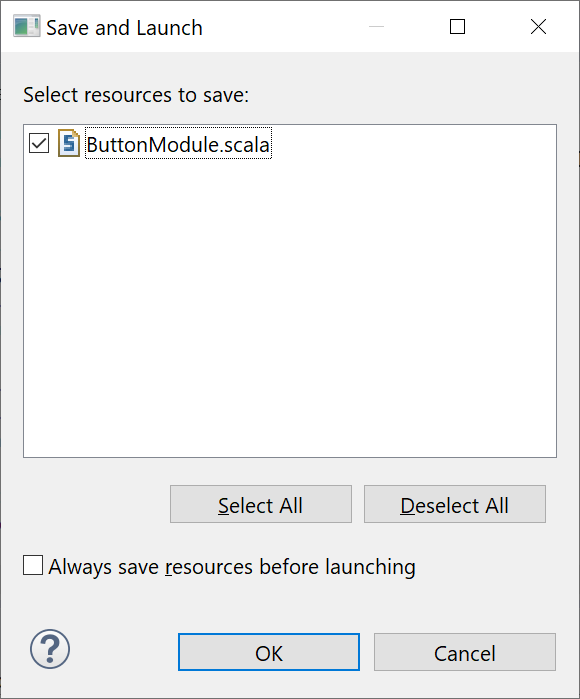
frame.setVisible(**true**)

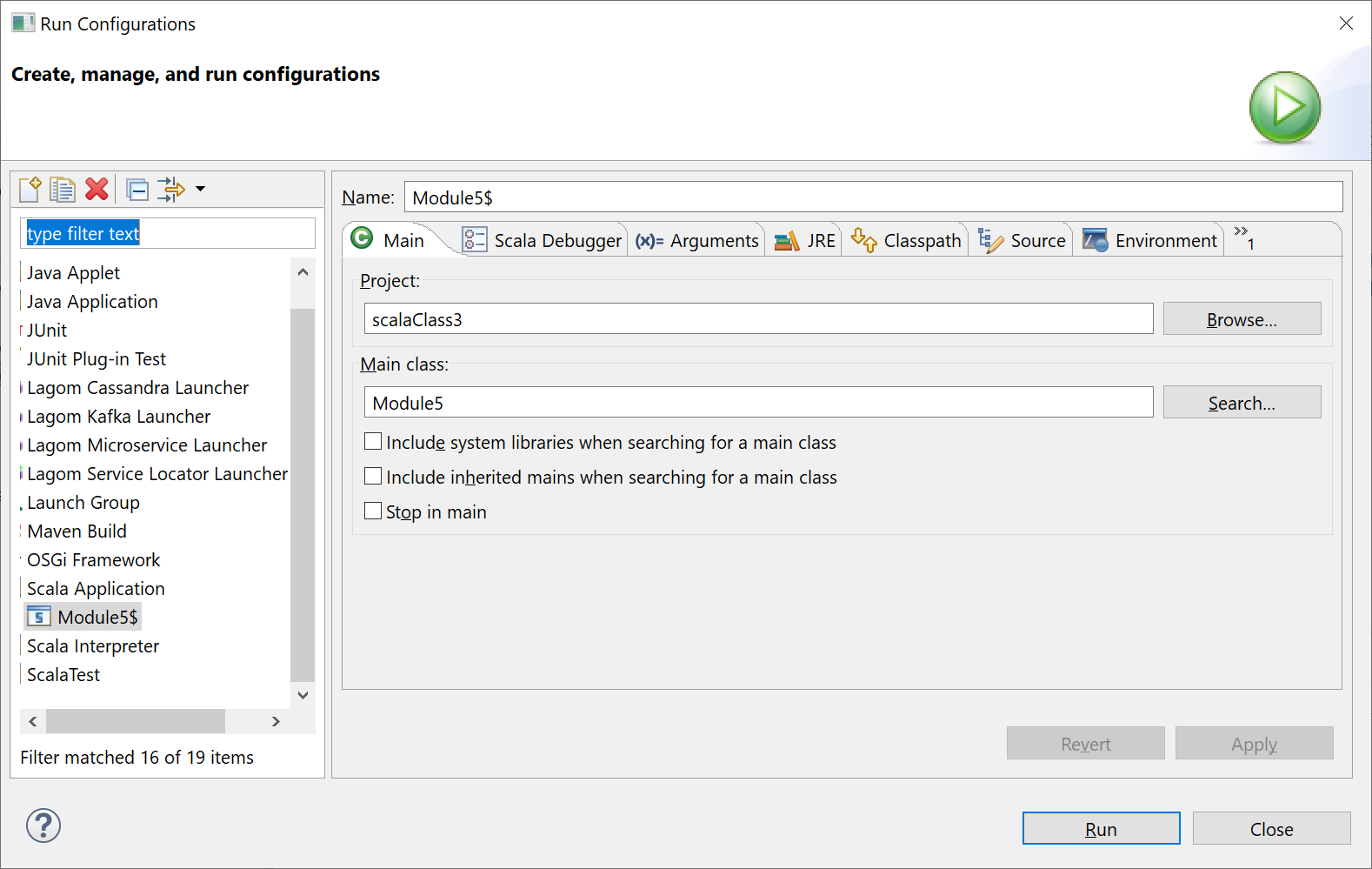
})

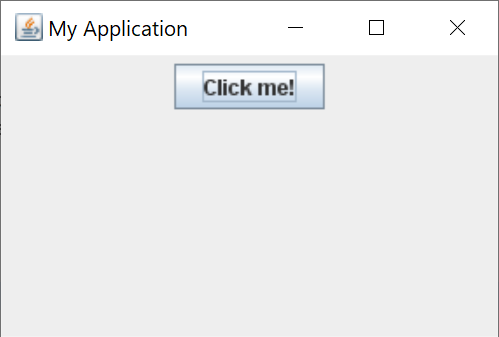
}

}

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

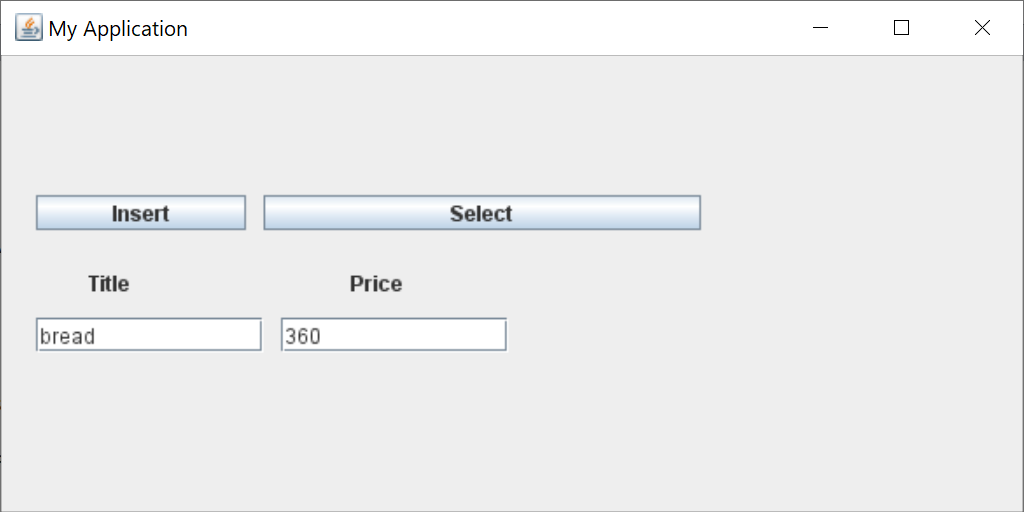






При нажатии на кнопку выводится строка в консольном окне среды Eclipse

Мы построим следующий экран

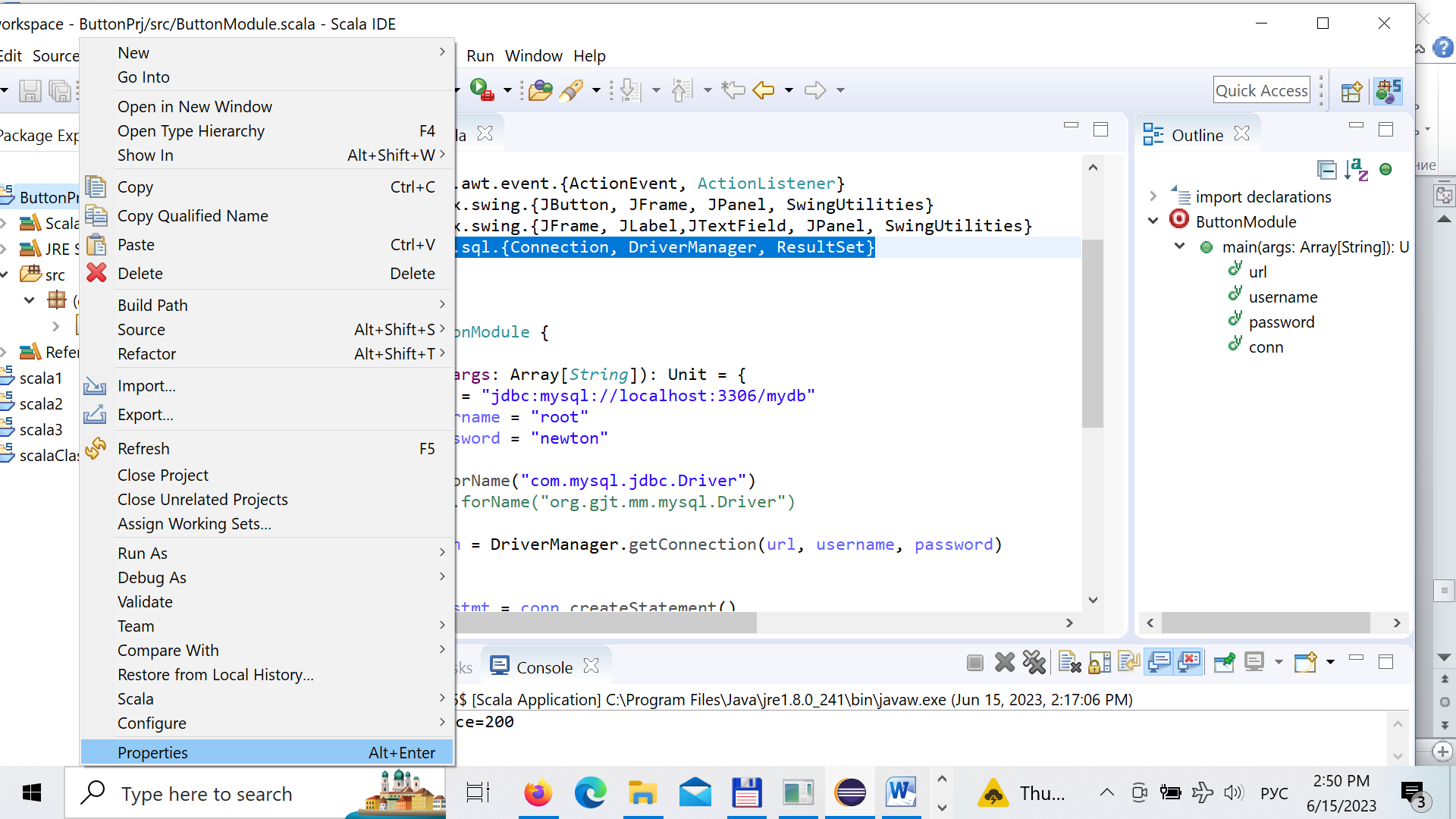


С помощью кнопки Insert добавляем запись в таблицу базы данных, с помощью кнопки Select – выводим содержимое. Чтобы соединиться с базой, нам нужен один импорт.

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

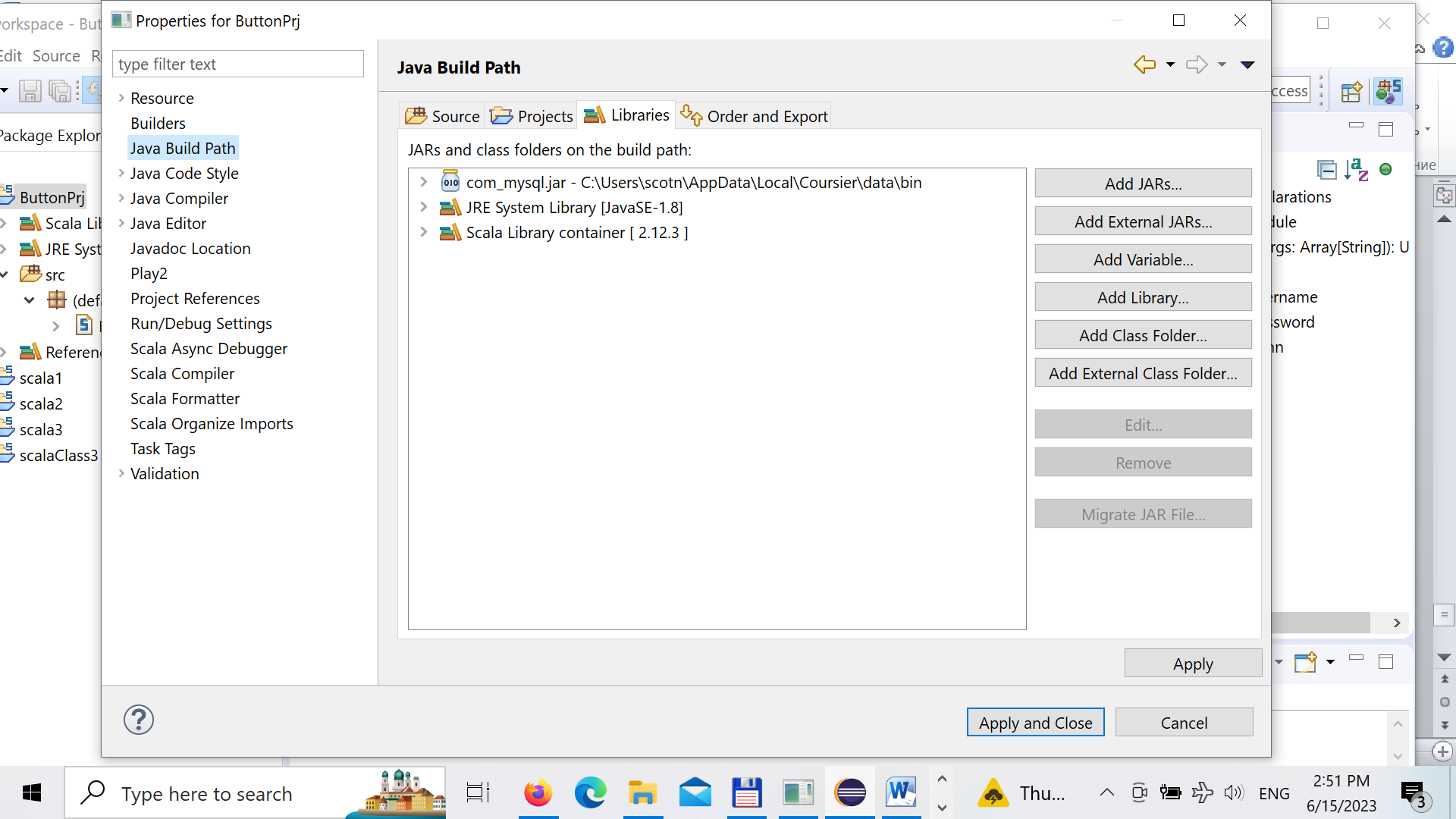
Но нам нужно подключить jar – файл com\_mysql.jar. Это делается так

Щелкаем правой кнопкой мыши на узле проекта



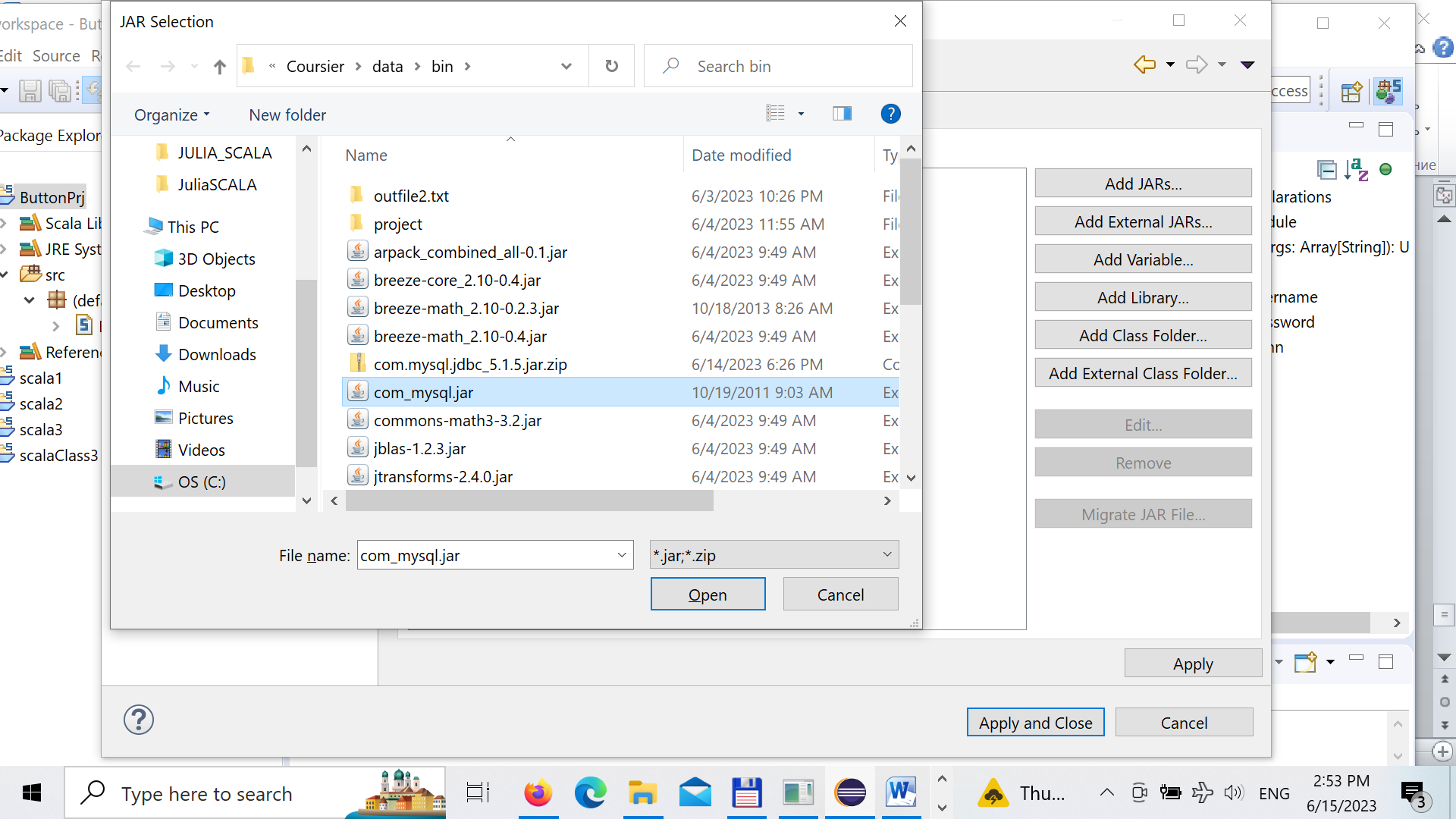
Выбираем пункт Properties

Выбираем пункт Java Build Path

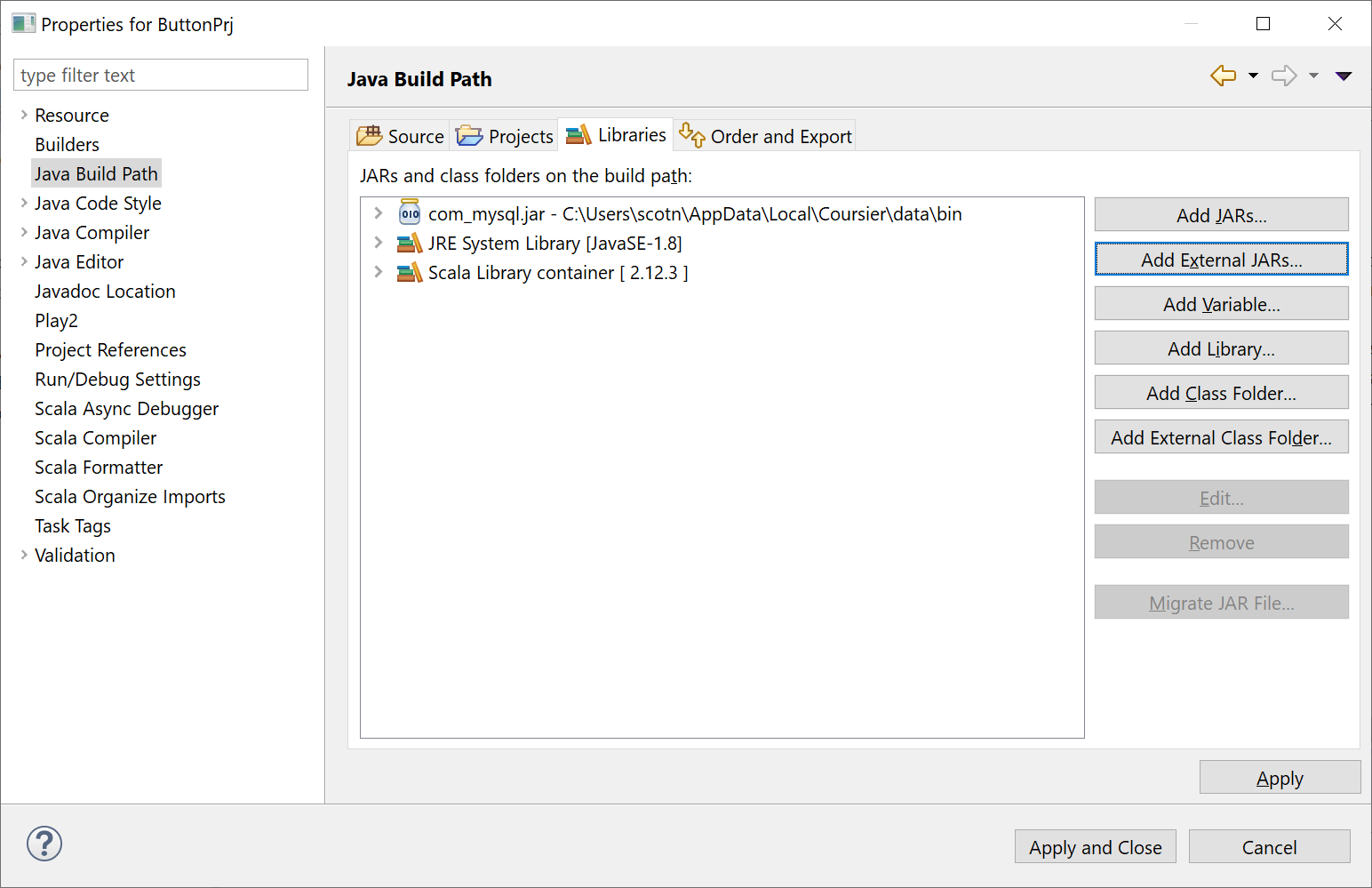


Выбираем пункт Add External Jars:

Находим наш jar-файл и жмем Open



Нажимаем Apply and Close



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**)

})

}

}

Разберитесь с этим приложением.

***Порядок выполнения работы***

1. Изучить теоретическую часть.
2. Создать и отладить приложение в среде Eclipse Oxygen.
3. Выполнить задание.
4. Написать отчет.
5. Защитить работу.

***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).

***Контрольные вопросы***

1. Что такое драйвер с базой данной? Как он подключается?
2. Что такое соединение, как его открыть?
3. Какие sql-запросы использовались?