1. git config -global user.name “name”
2. git config -global user.email “email”
3. git init
4. git remote add origin “ссылка на репозиторий”
5. git add -A
6. git commit -m “описание коммита”
7. git branch session1
8. git checkout session1
9. git push

public BaseForm(){  
setTitle("ООО Фирма");//название приложения  
setMinimumSize(new Dimension(1400,800)); //установка разрешения окна  
setLocationRelativeTo(null); //центрирование окна  
setIconImage(new ImageIcon(getClass().getResource("imgs/school\_logo.png")).getImage()); //добавление иконки  
}

public ClientTableForm() {  
setVisible(true);

pack();

setContentPane(mainPanel); //выбор панели

table.setRowHeight(25); //увеличение строк

// Создание модели данных для таблицы

DefaultTableModel client\_mdl = new DefaultTableModel();

// Указание названий столбцов

client\_mdl.setColumnIdentifiers(new String[]{"ID","Имя","Фамилия","Отчество","Дата рождения","Дата регистрации","Почта","Номер телефона","Пол","Фото"});

// Ловятся исключения, связанные с БД

try {

// Подключение к БД demoexam по адресу 127.0.0.1:3306, пользователь - root

Connection connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/demoexam?serverTimeZone=UTC","root","");

// Подготовка запроса

PreparedStatement preparedStatement = connection.prepareStatement("SELECT \* FROM client");

// Выолнение запроса, в ResultSet помещаются строки из БД

ResultSet resultSet = preparedStatement.executeQuery();

// Прогон по каждой cтроке БД по порядку, пока строки существуют

while (resultSet.next()){

// Добавление в модель строки с данными из БД

client\_mdl.addRow(new String[]{

// Взятие из строки ResultSet'а значения из первого столбца

resultSet.getString(1),

resultSet.getString(2),

resultSet.getString(3),

resultSet.getString(4),

resultSet.getString(5),

resultSet.getString(6),

resultSet.getString(7),

resultSet.getString(8),

resultSet.getString(9).equals("м") ? "Мужской" : "Женский",

resultSet.getString(10)

});

}

connection.close();

} catch (SQLException throwables) {

throwables.printStackTrace();

}

// Установка созданной модели на объект таблицы

table.setModel(client\_mdl);

table.addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

int row = table.getSelectedRow();

if (row != -1 && e.getKeyCode() == KeyEvent.VK\_DELETE)

{

int id = Integer.parseInt((String) table.getValueAt(row, 0));

try {

Connection c = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/demoexam?serverTimeZone=UTC","root","");

String sql = "DELETE FROM client WHERE id=?";

PreparedStatement ps = c.prepareStatement(sql);

ps.setInt(1, id);

ps.executeUpdate();

new ClientTableForm();

dispose();

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

}

});

}

table.setModel(client\_mdl);

table.addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

int row = table.getSelectedRow();

if (row != -1 && e.getKeyCode() == KeyEvent.VK\_DELETE) {

if (JOptionPane.showConfirmDialog

(null, "Вы хотите удалить данную строку?", "Внимание",

JOptionPane.YES\_NO\_OPTION) == JOptionPane.YES\_OPTION) {

int id = Integer.parseInt((String) table.getValueAt(row, 0));

try {

Connection c = DriverManager.getConnection

("jdbc:mysql://127.0.0.1:3306/demoexam?serverTimeZone=UTC", "root", "");

String sql = "DELETE FROM client WHERE id=?";

PreparedStatement ps = c.prepareStatement(sql);

ps.setInt(1, id);

ps.executeUpdate();

new ClientTableForm();

dispose();

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

}

}

});

@Test  
@DisplayName(“Проверка подключения к базе данных”)  
public void testDatabaseConnection() {

Exeption error = null;

Try(Connection…){

}

Assertions.assertEquals(true, error == null);

}