**Процесс разработки простой GUI программы на языке Java**

Загружаем IntelliJ IDEA

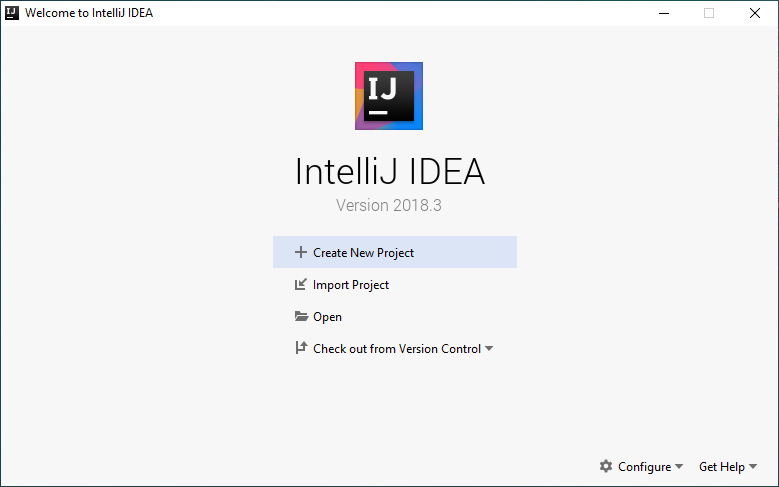


Рис. 1

Создаем новый проект: Create New Project или File, New, Project. Появляется окно (рис.2).

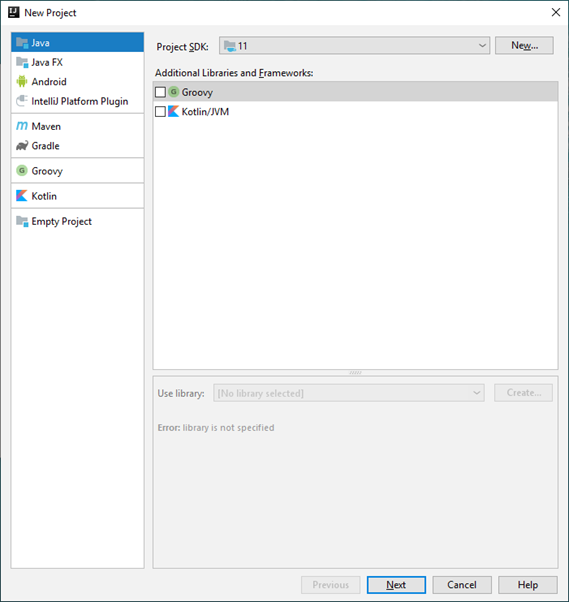


Рис.2

Выделяем Java, нажимаем Next, появляется окно (рис.3).

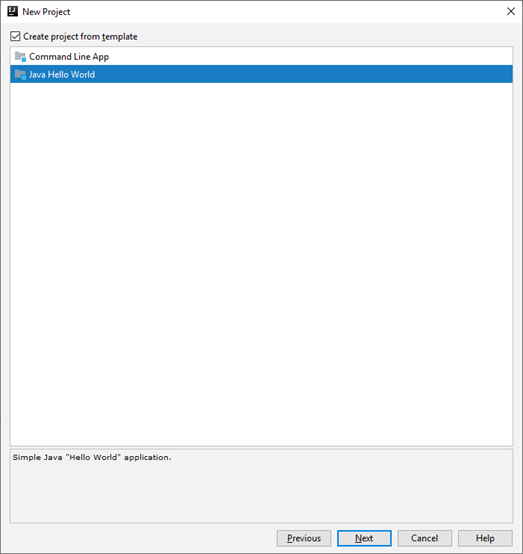


Рис.3

Выбираем “Create Project from template” и “Java Hello Word”, нажимаем Next, появляется окно (рис.4).

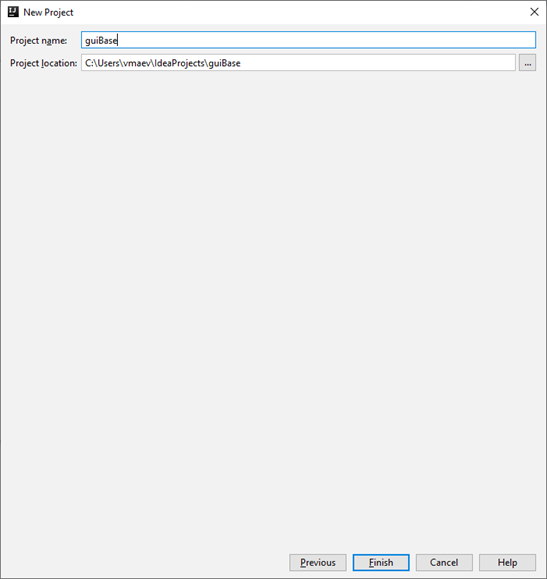


Рис.4

Указываем имя проекта “guiBase” и его место расположения, затем нажимаем кнопку Finish, появляется окно (рис.5).

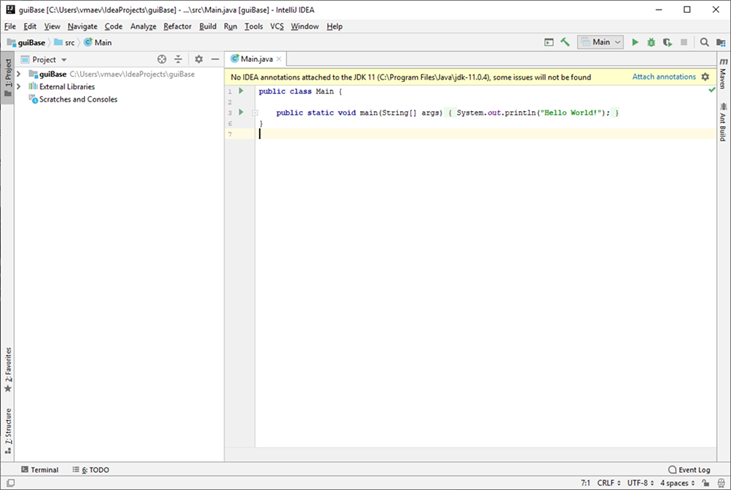


Рис.5

На данном этапе можно запустить проект в консольном режиме и получить в результате сообщение «Hello World!». Но нам нужен проект в графическом режиме. Поэтому продолжим создание проекта. Для этого раскрываем наш проект и выделяем папку src. Далее включаем пункты меню File, New, Gui Form (рис.6).

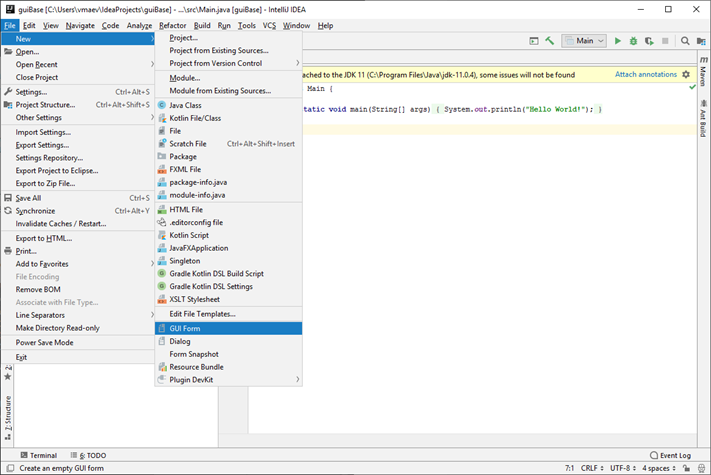


Рис.6

Появляется окно для ввода имени создаваемой формы (рис.7).

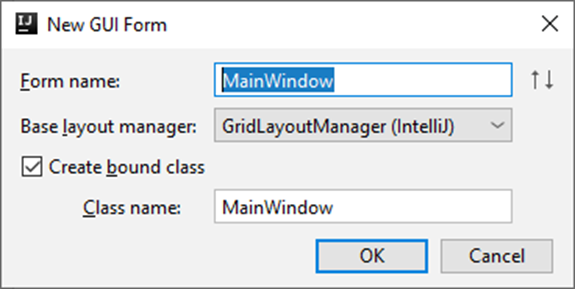


Рис.7

Ввожу имя MainWindow и нажимаю «ok», появляется окно (рис. 8).

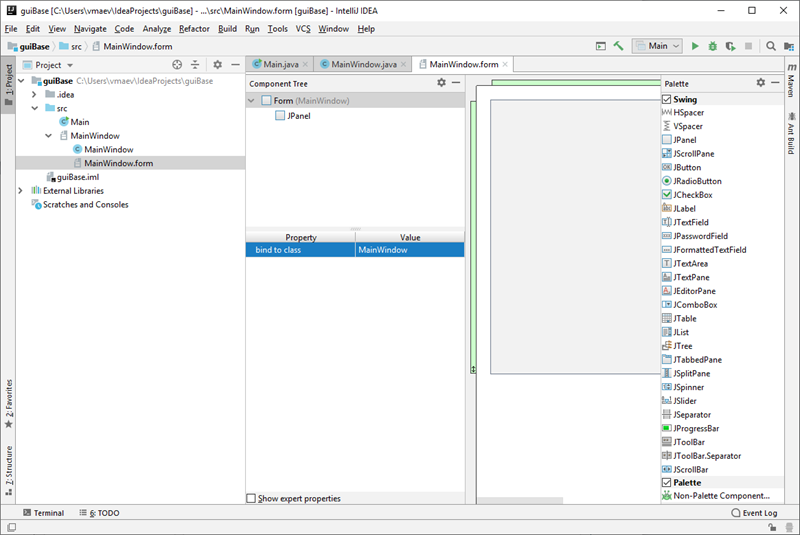


Рис.8

В этом окне появляется дерево компонентов «Component Tree», выделяю на нем компонент JPanel и ввожу значение «panel» его свойства field name (см.рис.9).

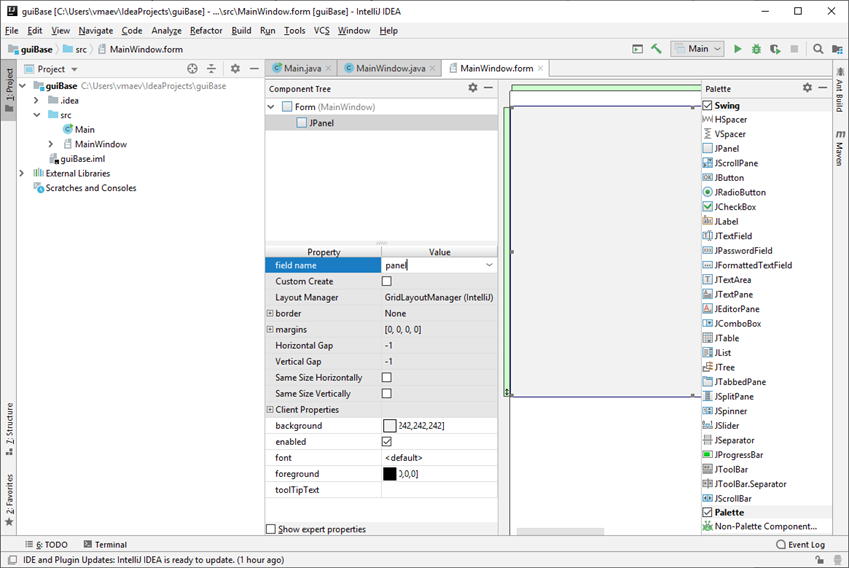


Рис.9

После этого из палитры компонентов (Palette) на форму перетаскиваю компоненты JTextField, PasswordField, JButton (см. рис.10).

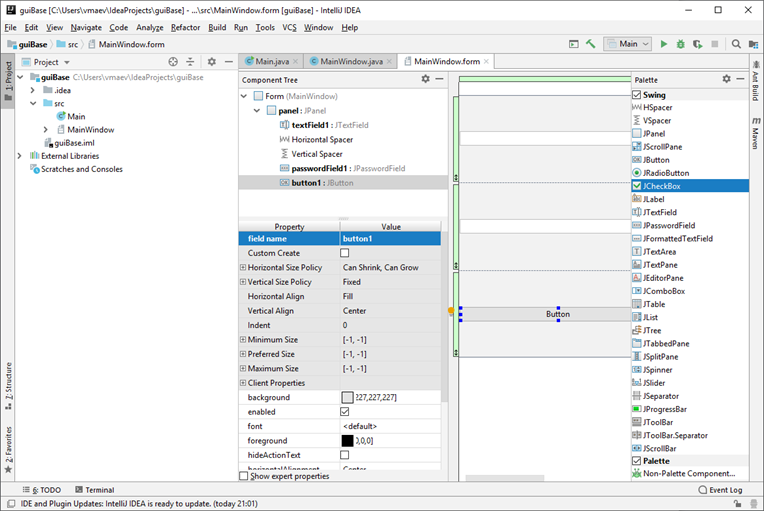


Рис.10

Теперь надо добавить программный код и связать нашу форму с ним. После того, как мы добавили форму MainWindow, автоматически создался класс MainWindow. Этот класс является классом созданной формы, данный класс будет обслуживать все события данной формы. Содержание этого класса будет автоматически редактироваться во время изменения формы. Код класса находится на вкладке MainWindow.java (см. рис.11).

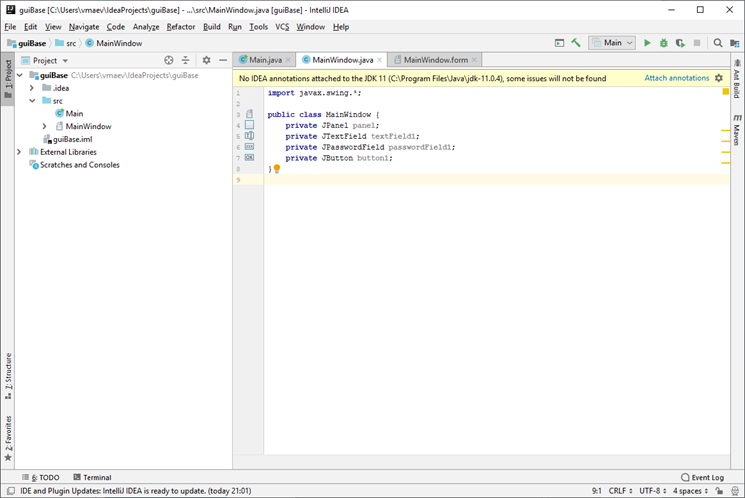


Рис.11

Необходимо данный класс сделать наследником компонента JFrame (расширить его). Для этого в заголовке класса добавляем код extends JFrame (см. рис.12).

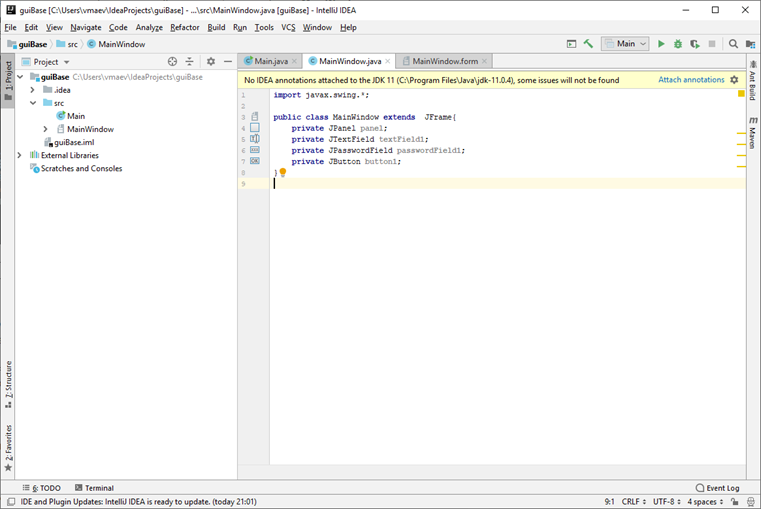


Рис.12

В описание класса добавляем конструктор этого класса, в описание которого вводим строку: this.getContentPane().add(panel) (см. рис. 13).

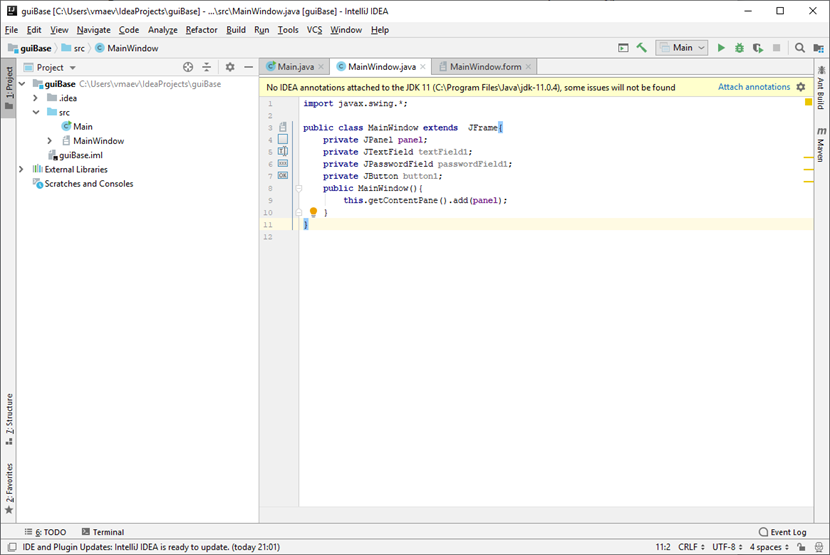


Рис.13

Сейчас необходимо в код метода «main» нашего проекта добавить код создания нашей формы. Для этого выделяем вкладку Main.java и вводим следующий код:

//Создание окна формы

MainWindow mainWindow= new MainWindow();

// Упаковка всех элементов нашей формы

mainWindow.pack();

//Установка размеров окна

mainWindow.setSize(new Dimension(200,200));

mainWindow.setVisible(true);

Удаляем строку «System.out.println("Hello World!");»

В результате, получаем: (см. рис.14).

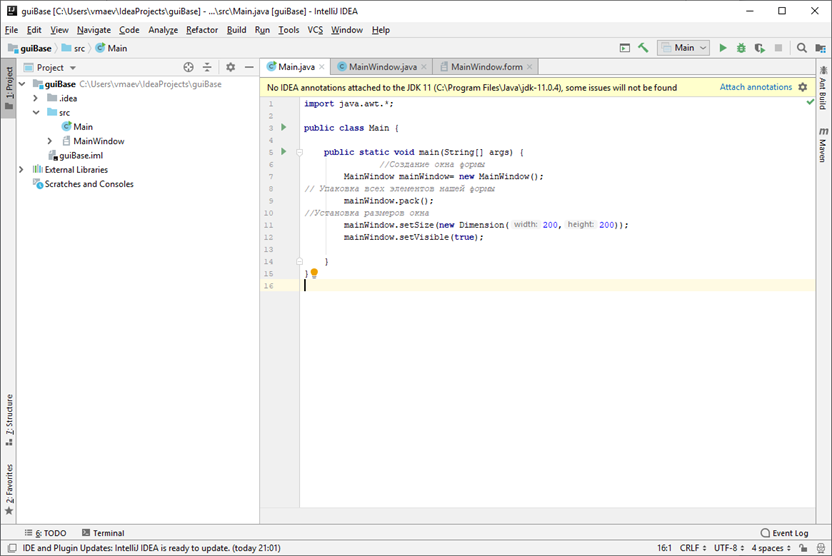


Рис.14.

Затем необходимо в тело класса MainWindow добавить обработчик щелчка мыши на кнопке JButton:

private class MyButtonListener inplements ActionListener{

public void actionPerformed(ActionEvent actionEvent){

}

}

В результате получаем следующее содержимое класса MainWindow:

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class MainWindow extends JFrame{

private JPanel panel;

private JTextField textField1;

private JPasswordField passwordField1;

private JButton button1;

private class MyButtonListener implements ActionListener {

public void actionPerformed(ActionEvent actionEvent){

}

}

public MainWindow(){

this.getContentPane().add(panel);

}

}

Метод actionPerformed() будет обрабатывать все события кнопки button1. Кнопке необходимо указать какой класс будет обрабатывать. Поэтому необходимо в конструктор класса MainWindow добавить следующий код:

this.button1.addActionListener(new MybuttonListener()); Затем в обработчик щелчка мыши на кнопке добавим следующий код:

if (textfield1.getText().equals(passwordField1.getText())){

JOptionPane.showMessageDialog(null, “Yes”);}

else {

JOptionPane.showMessageDialog(null, “No”);}

В результате, код на вкладке MainWindow.java будет следующим:

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class MainWindow extends JFrame{

private JPanel panel;

private JTextField textField1;

private JPasswordField passwordField1;

private JButton button1;

private class MyButtonListener implements ActionListener {

public void actionPerformed(ActionEvent actionEvent){

if (textField1.getText().equals(passwordField1.getText())){

JOptionPane.showMessageDialog(null, "Yes");}

else {

JOptionPane.showMessageDialog(null, "No");}

}

}

public MainWindow(){

this.getContentPane().add(panel);

this.button1.addActionListener(new MyButtonListener());

}

}(см. рис. 15)

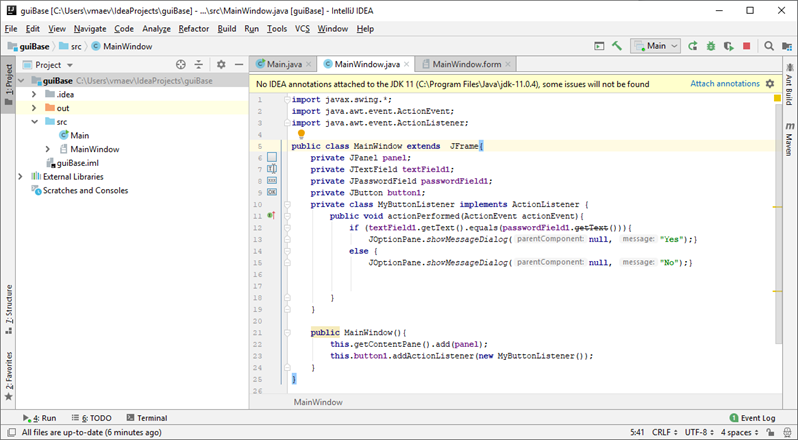


Рис.15

Запускаем наш проект (выделяем имя проекта в окне Project и включаем пункты меню Run, Run ‘Main’). Появляется окно (см. рис.16).

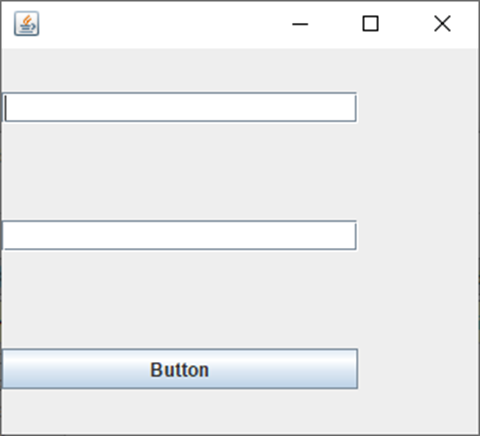


Рис.16

Если в оба поля ввести одну и туже информацию, например число 12, и щелкнуть мышкой по кнопке, то появляется сообщение «Yes» (см. рис.17), а если ввести в эти поля разную информацию, например, числа 12 и 14, то появляется сообщение «No» (см. рис. 18).

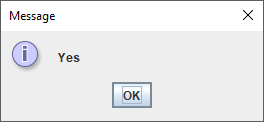


Рис.17

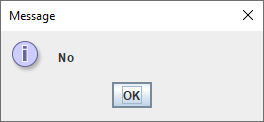


Рис.18