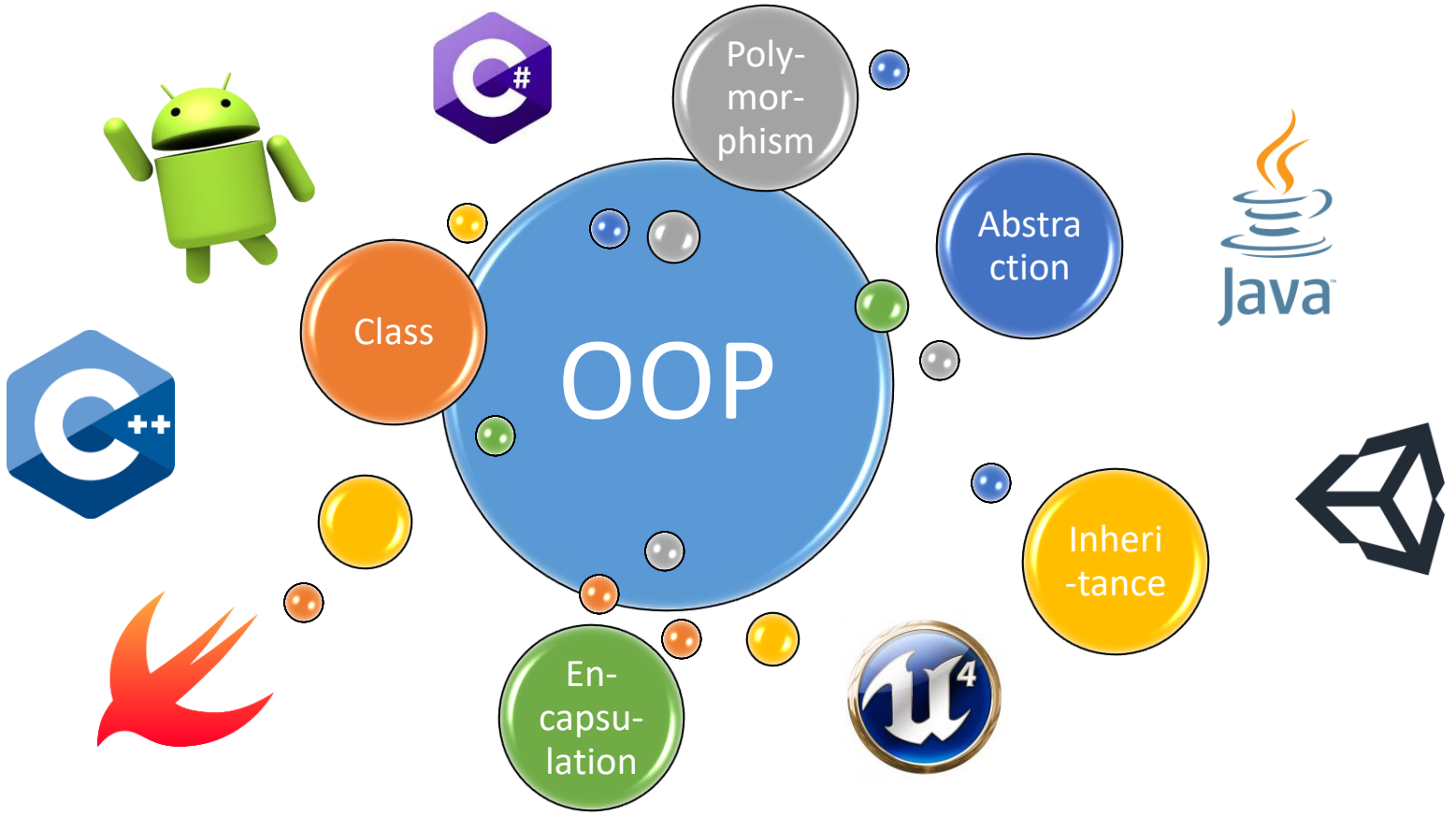


Java Swing

Kuan-Ting Lai

2020/3/21

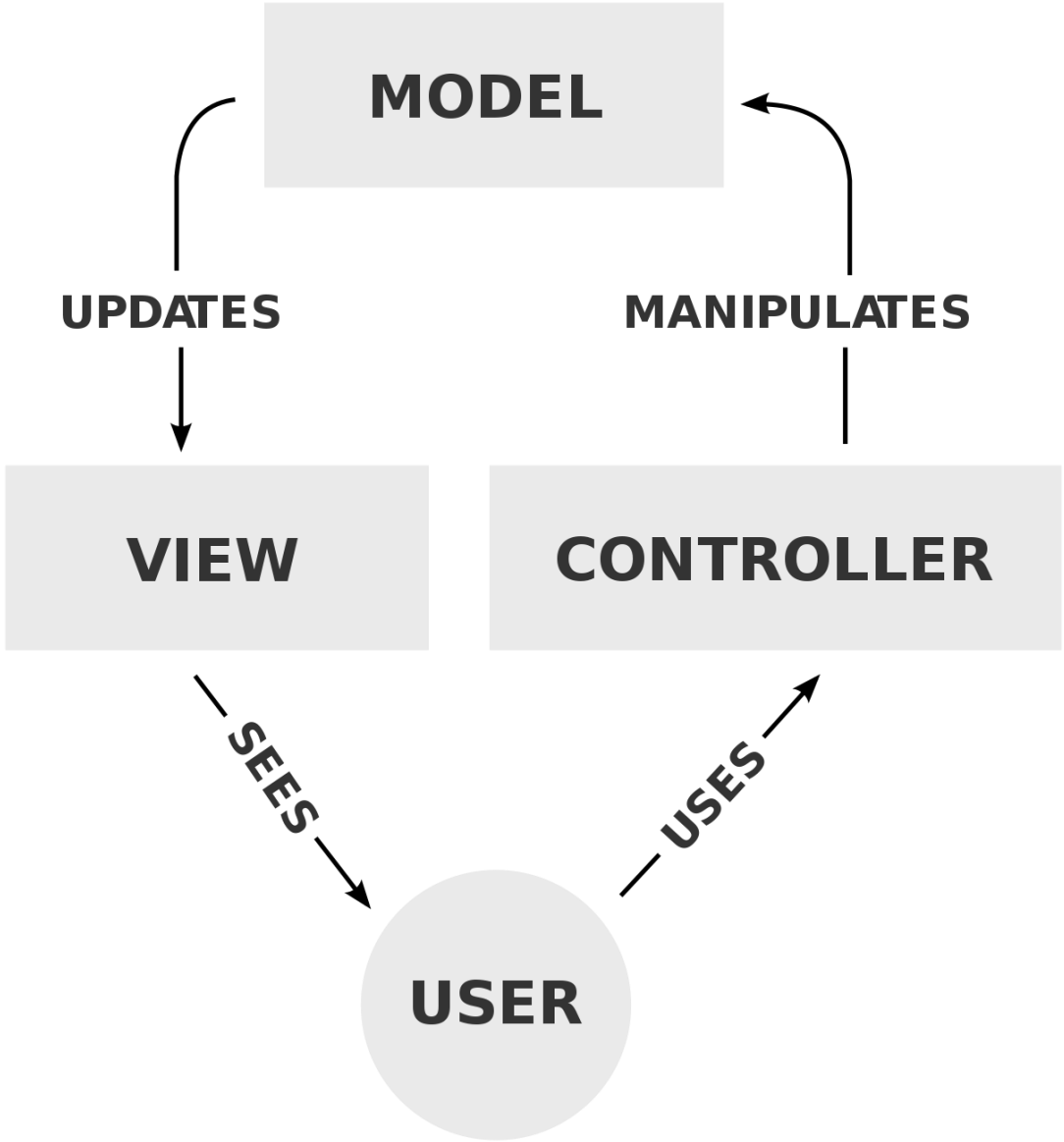
Java Swing

- Used to create **Window-based** applications
- Part of Java Foundation Classes (JFC)
- Platform-independent
- Follow MVC design pattern

https://www.javatpoint.com/java-swing

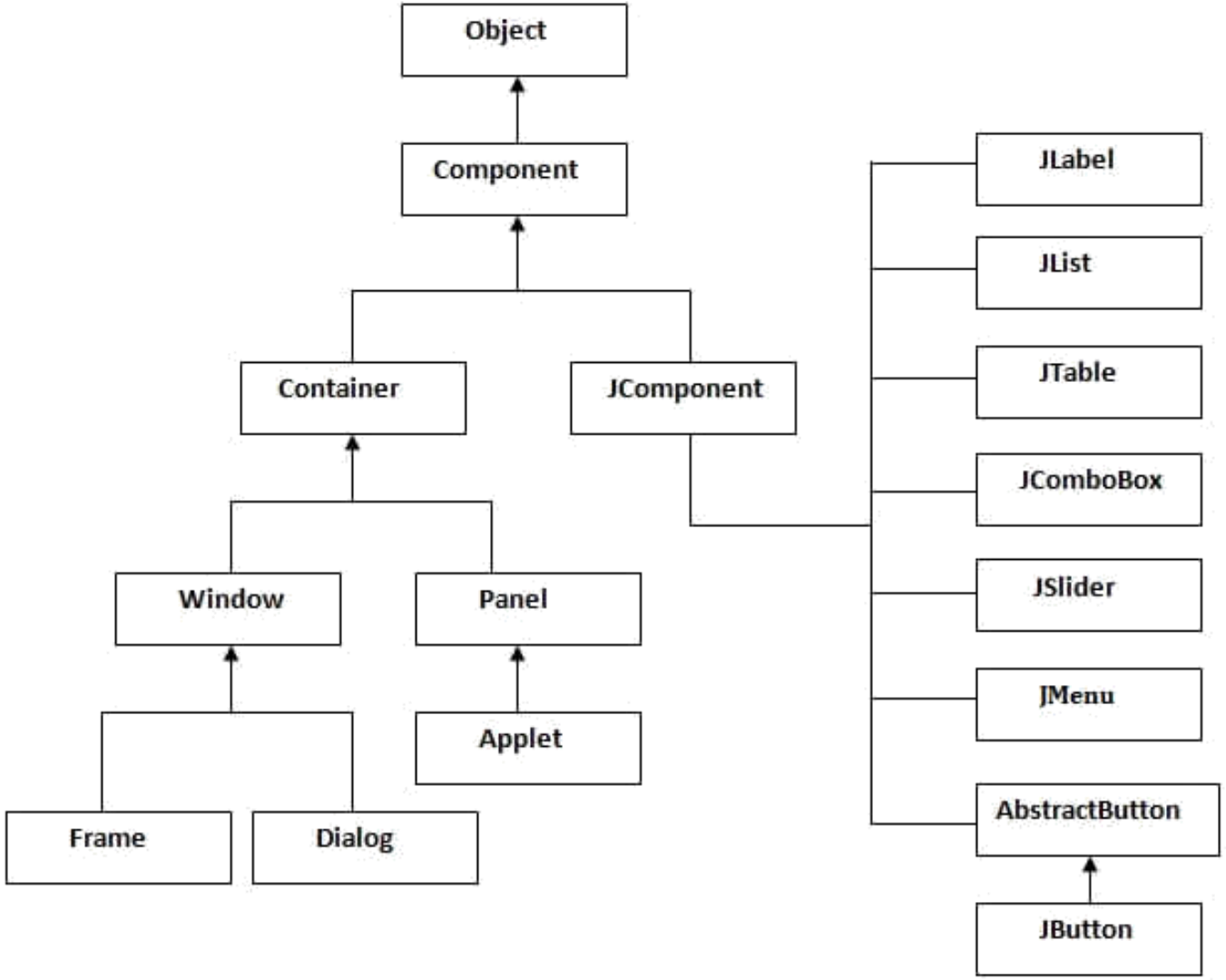
Model-View-Controller (MVC)

- Most common GUI design pattern
- Model
  - Manage the data.
- View
  - Visualize the data, e.g. chart, diagram or table
- Controller
  - Accepts input and converts it to commands for the model or view



https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

Hierarchy of Java Swing Classes



Common Methods

Method	Description
`public void add(Component c)`	Add a component on another component.
`public void setSize(int width, int height)`	Set size of the component.
`public void setLayout(LayoutManager m)`	Set the layout manager for the component.
`public void setVisible(boolean b)`	Set the visibility of the component. It is by default false.

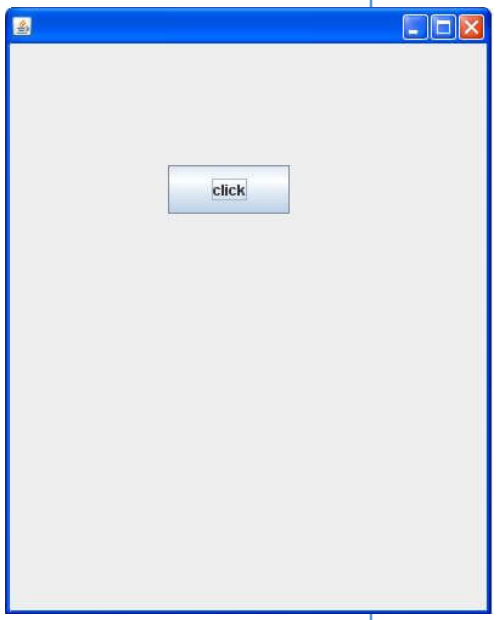
Creating a Java Swing Frame

1. By creating the object of Frame class (association)

2. By extending Frame class (inheritance)

Hello Swing

```
import javax.swing.*;

public class HelloSwing
{
    public static void main(String[] args) {
        JFrame f=new JFrame();           //creating instance of JFrame

        JButton b=new JButton("click"); //creating instance of JButton
        b.setBounds(130,100,100, 40);  //x axis, y axis, width, height

        f.add(b); //adding button in JFrame

        f.setSize(400,500); //400 width and 500 height
        f.setLayout(null);   //using no layout managers
        f.setVisible(true);   //making the frame visible
    }
}
```

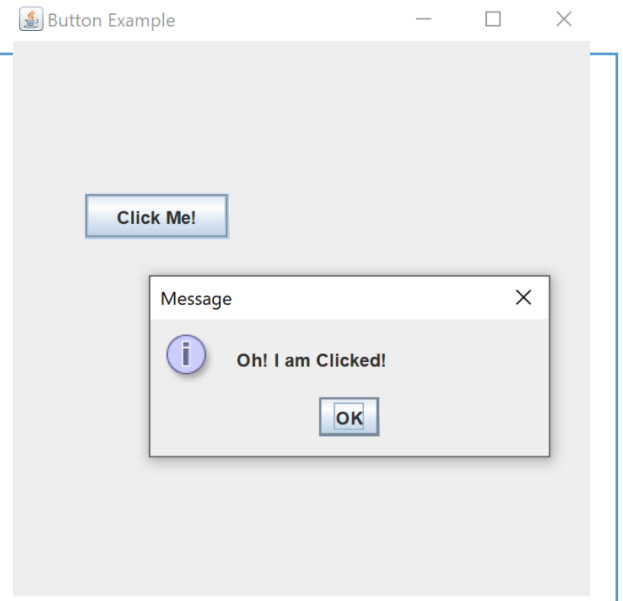


Using Swing by Inheritance

```
import javax.swing.*;

public class HelloSwing2 extends JFrame { //inheriting JFrame

    HelloSwing2() {
        JButton b=new JButton("click");//create button
        b.setBounds(130,100,100, 40);

        add(b);//adding button on frame
        setSize(400,500);
        setLayout(null);
        setVisible(true);
    }

    public static void main(String[] args) {
        new HelloSwing2();
    }
}
```


JButton Example with ActionListener

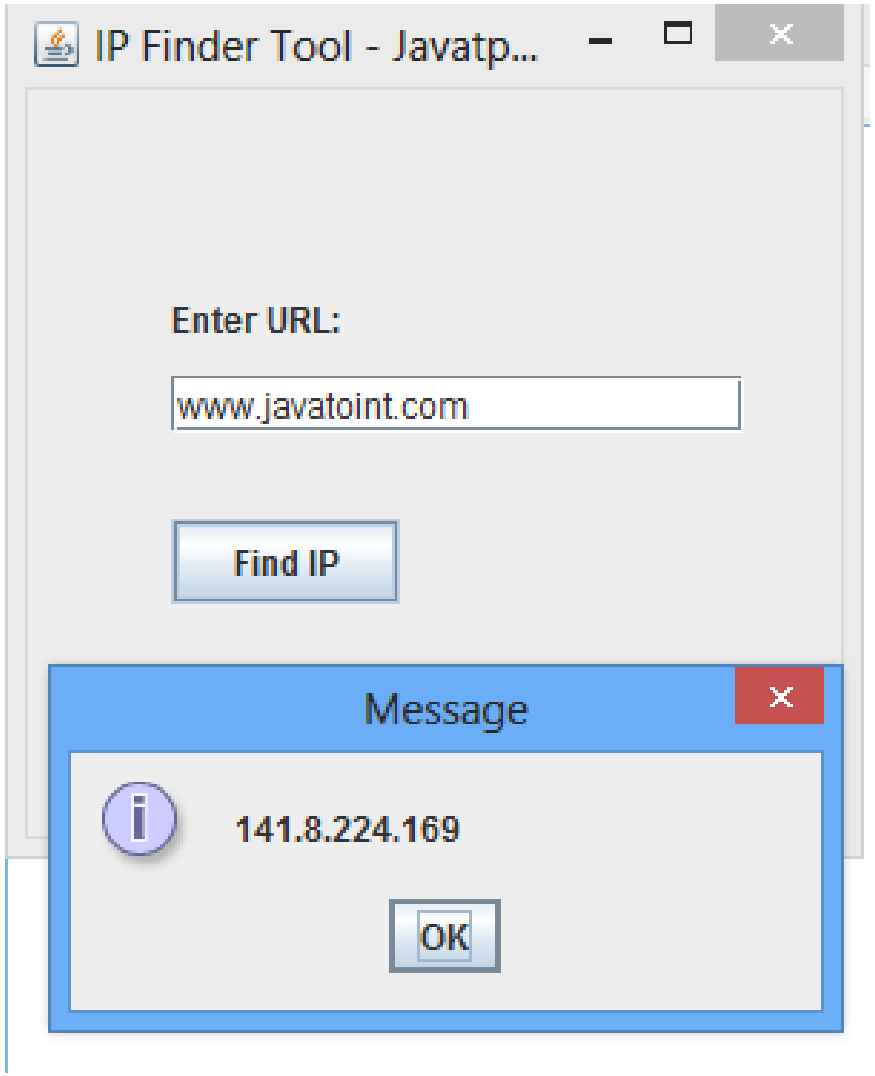
```
import java.awt.event.*;
import javax.swing.*;
public class ButtonExample {
    public static void main(String[] args) {
        JFrame f=new JFrame("Button Example");
        JButton b=new JButton("Click Me!");
        b.setBounds(50,100,95,30);

        b.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    JOptionPane.showMessageDialog(null, "Oh! I am Clicked!");
                }
        });
        f.add(b);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```



JButton Example Implements ActionListener

```
import java.awt.event.*;
import javax.swing.*;
public class ButtonExample2  extends JFrame implements ActionListener {
    ButtonExample2() {
        JButton b=new JButton("Click Me!");
        b.setBounds(50,100,95,30);
        b.addActionListener(this);
        add(b);
        setSize(400,400);
        setLayout(null);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(null, "Oh! I am Clicked!");
    }
    public static void main(String[] args) {
       new  ButtonExample2();
    }
}
```

IP Finder




```
import javax.swing.*;
import java.awt.event.*;
import java.net.*;
public class IPFinder extends JFrame implements ActionListener {
    JLabel l;
    JTextField tf;
    JButton b;
    IPFinder(){
        super("IP Finder Tool - Javatpoint");
        l=new JLabel("Enter URL:"); l.setBounds(50,70,150,20);;
        tf=new JTextField(); tf.setBounds(50,100,200,20);
        b=new JButton("Find IP");
        b.setBounds(50,150,80,30); b.addActionListener(this);
        add(l); add(tf); add(b);
        setSize(300,300);
        setLayout(null); setVisible(true);
    }
    public void actionPerformed(ActionEvent e){
        String url=tf.getText();
        try {
            InetAddress ia=InetAddress.getByName(url);
            String ip=ia.getHostAddress();
            JOptionPane.showMessageDialog(this,ip);
        } catch (UnknownHostException e1) {
            JOptionPane.showMessageDialog(this,e1.toString());
        }
    }
    public static void main(String[] args) {
        new IPFinder();
    }
}
```

Make an Executable Jar File

- The **jar (Java Archive)** provides the facility to create the executable file
  - java -jar myjar.jar

- Create a manifest file “myfile.mf”

myfile.mf
Main-Class: HelloSwing

Creating Executable Jar File using Jar Tool

- The jar tool provides many switches:
  - **-c** creates new archive file
  - **-v** verbose output.
  - **-m** includes manifest information from the given mf file.
  - **-f** specifies the archive file name
  - **-x** extracts files from the archive file

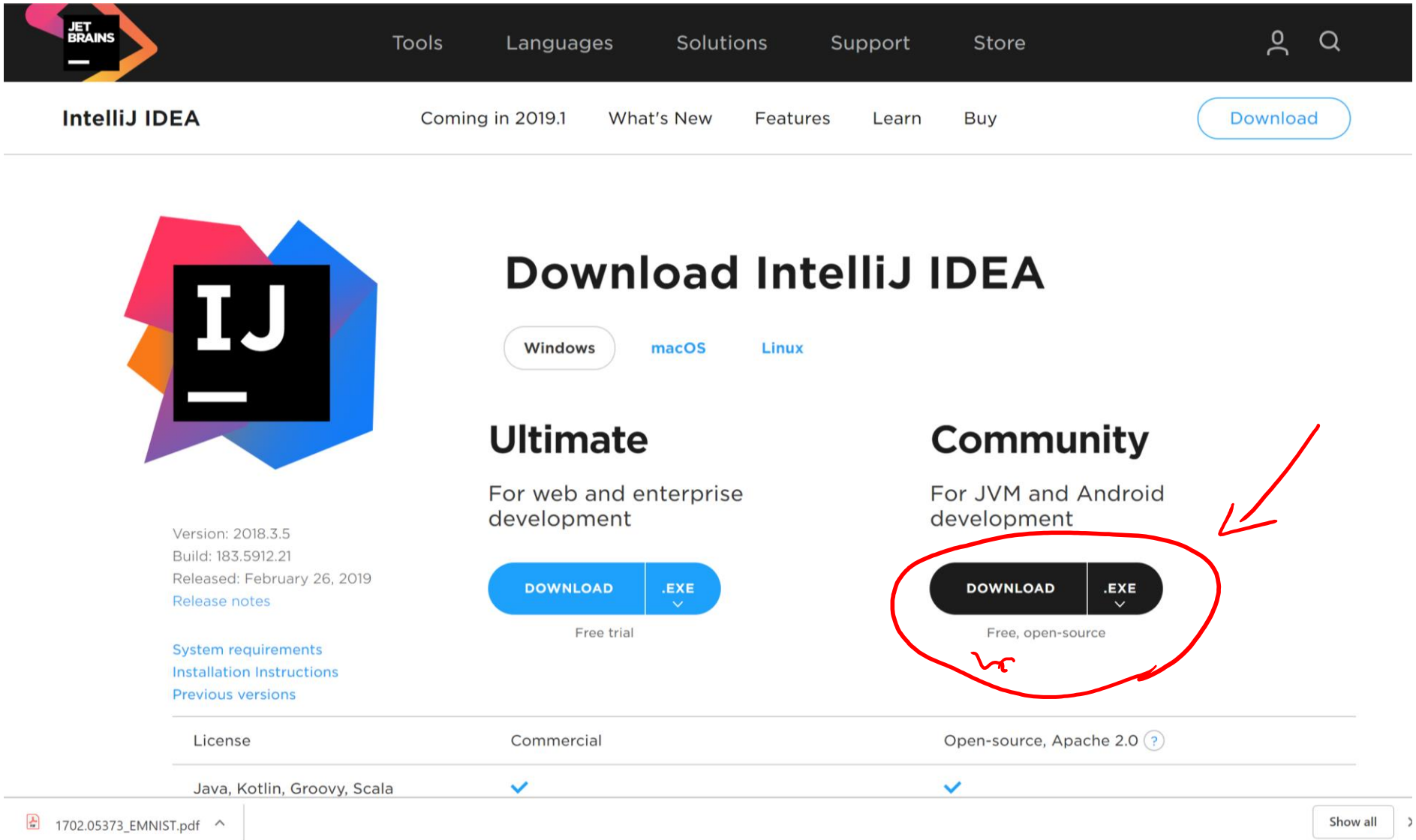
- Create a jar file:

  jar -cvmf myfile.mf myjar.jar First.**class**

- Run a jar file

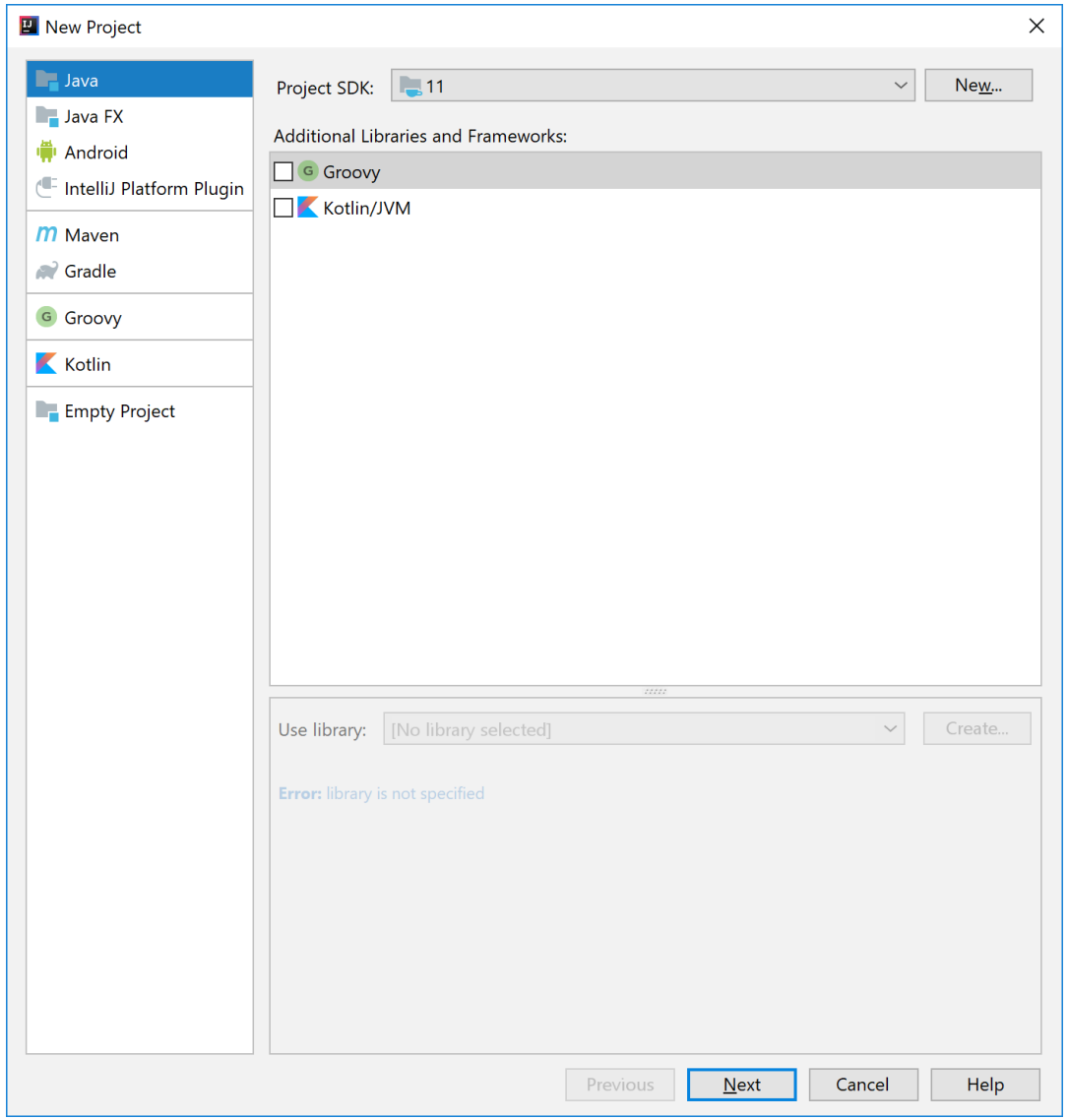
  java -jar myjar.jar

IntelliJ

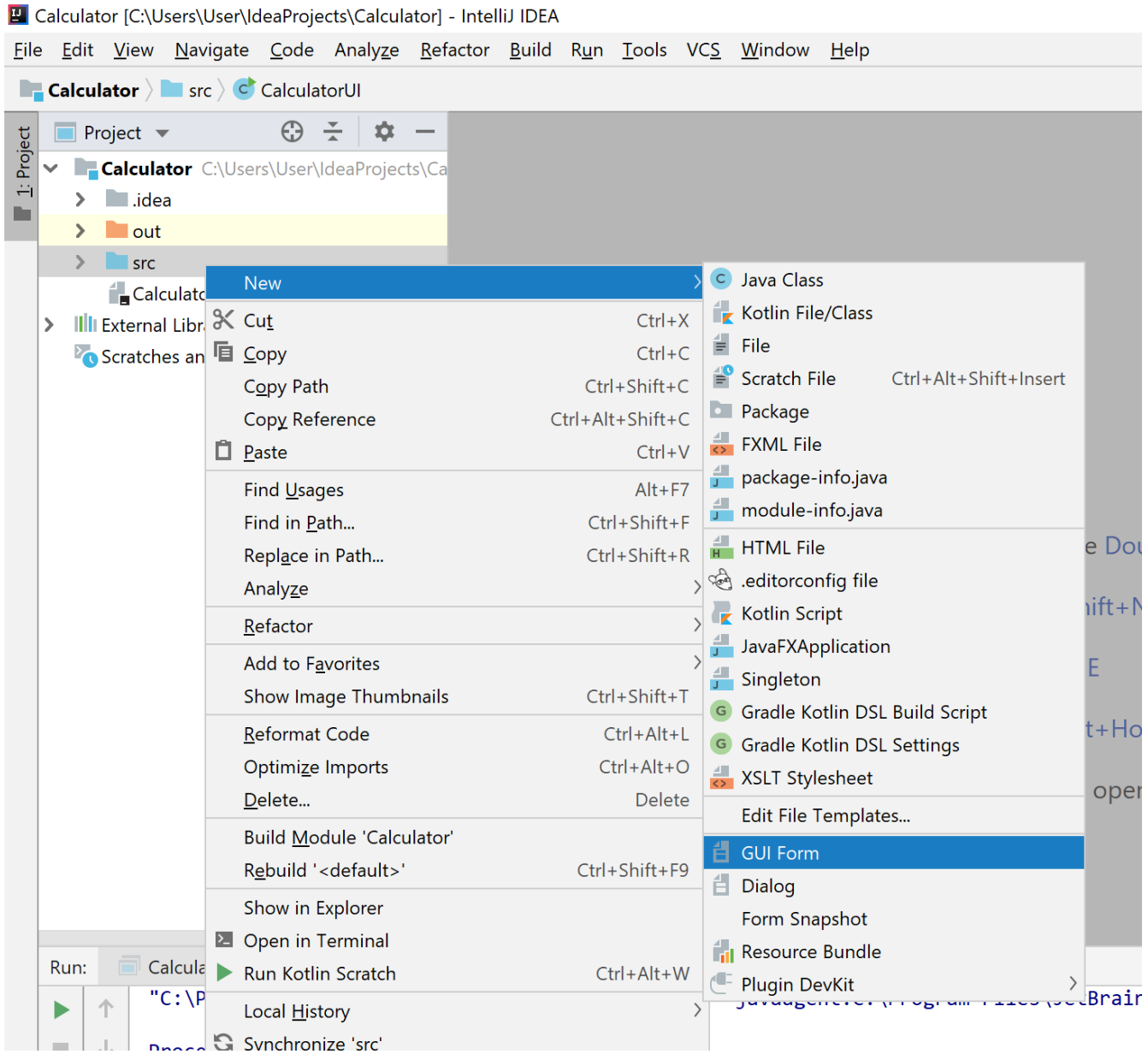

JET BRAINS
Tools
Languages
Solutions
Support
Store
IntelliJ IDEA
Coming in 2019.1
What's New
Features
Learn
Buy
Download
Download IntelliJ IDEA
Windows
macOS
Linux
Ultimate
For web and enterprise development
DOWNLOAD
.EXE
Free trial
Community
For JVM and Android development
DOWNLOAD
.EXE
Free, open-source
Version: 2018.3.5
Build: 183.5912.21
Released: February 26, 2019
Release notes
System requirements
Installation Instructions
Previous versions
License
Commercial
Open-source, Apache 2.0
Java, Kotlin, Groovy, Scala
1702.05373_EMNIST.pdf
Show all

Create a Java Project

- Press Next
- Don’t select any template

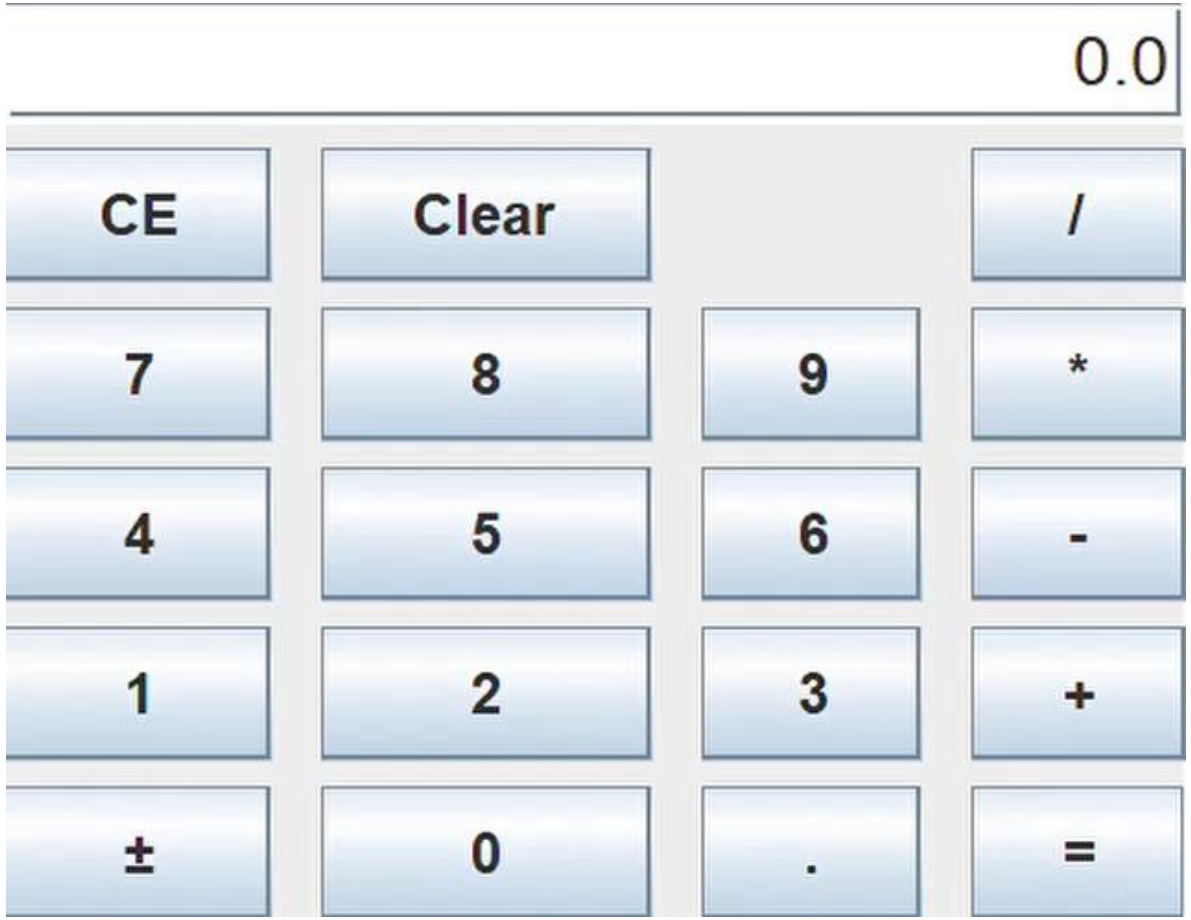


Add GUI Form


Calculator [C:\Users\User\IdeaProjects\Calculator] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Calculator src CalculatorUI
Project
Calculator C:\Users\User\IdeaProjects\Ca
.idea
out
src
External Libr
Scratches an
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show Image Thumbnails Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Build Module 'Calculator'
Rebuild '<default>' Ctrl+Shift+F9
Show in Explorer
Open in Terminal
Run Kotlin Scratch Ctrl+Alt+W
Local History
Synchronize 'src'
Java Class
Kotlin File/Class
File
Scratch File Ctrl+Alt+Shift+Insert
Package
FXML File
package-info.java
module-info.java
HTML File
.editorconfig file
Kotlin Script
JavaFXApplication
Singleton
Gradle Kotlin DSL Build Script
Gradle Kotlin DSL Settings
XSLT Stylesheet
Edit File Templates...
GUI Form
Dialog
Form Snapshot
Resource Bundle
Plugin DevKit
Run:

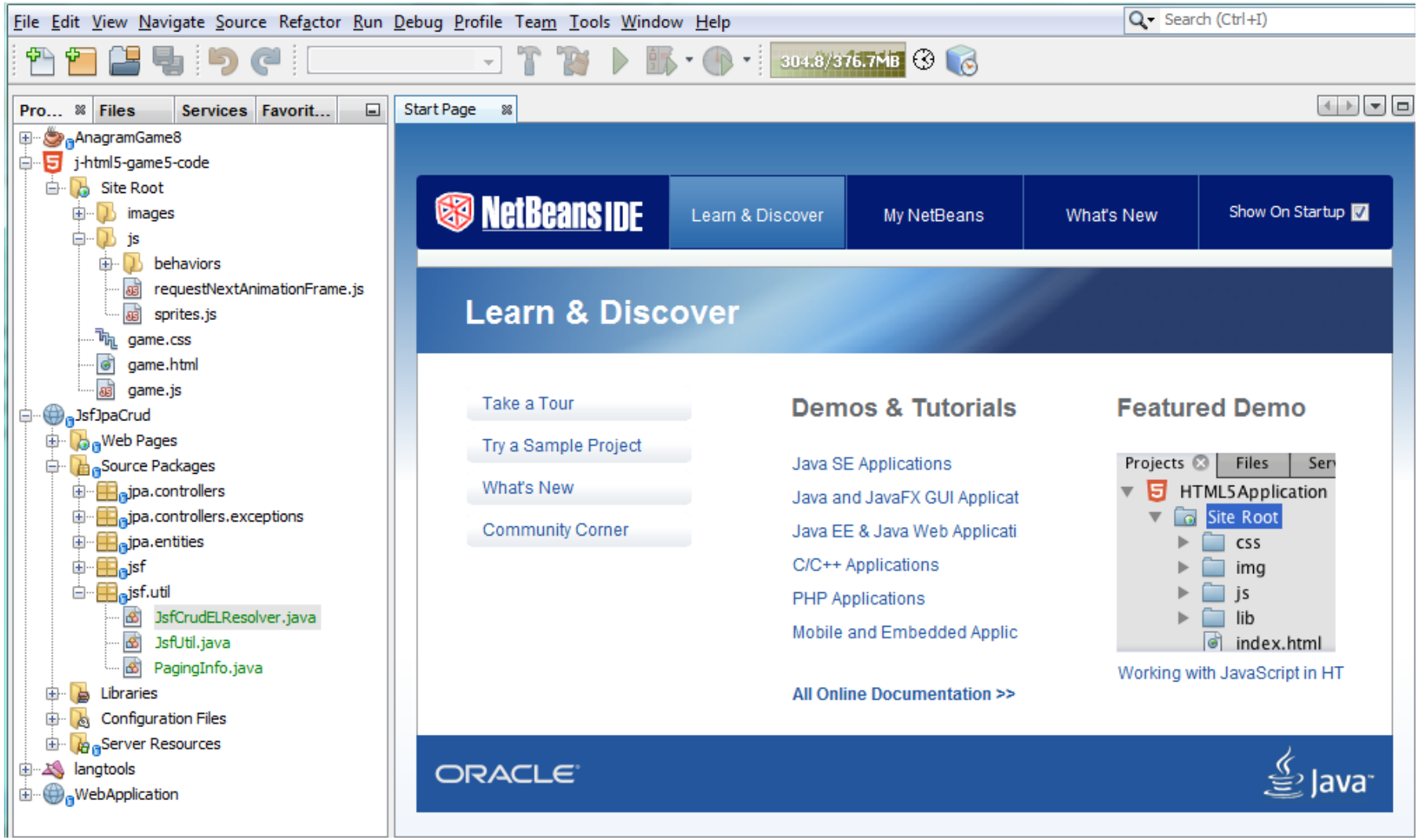
Homework: Build a Calculator using IntelliJ

- http://www.aiotlab.org/teaching/oop/homework/oop_hw2.pdf



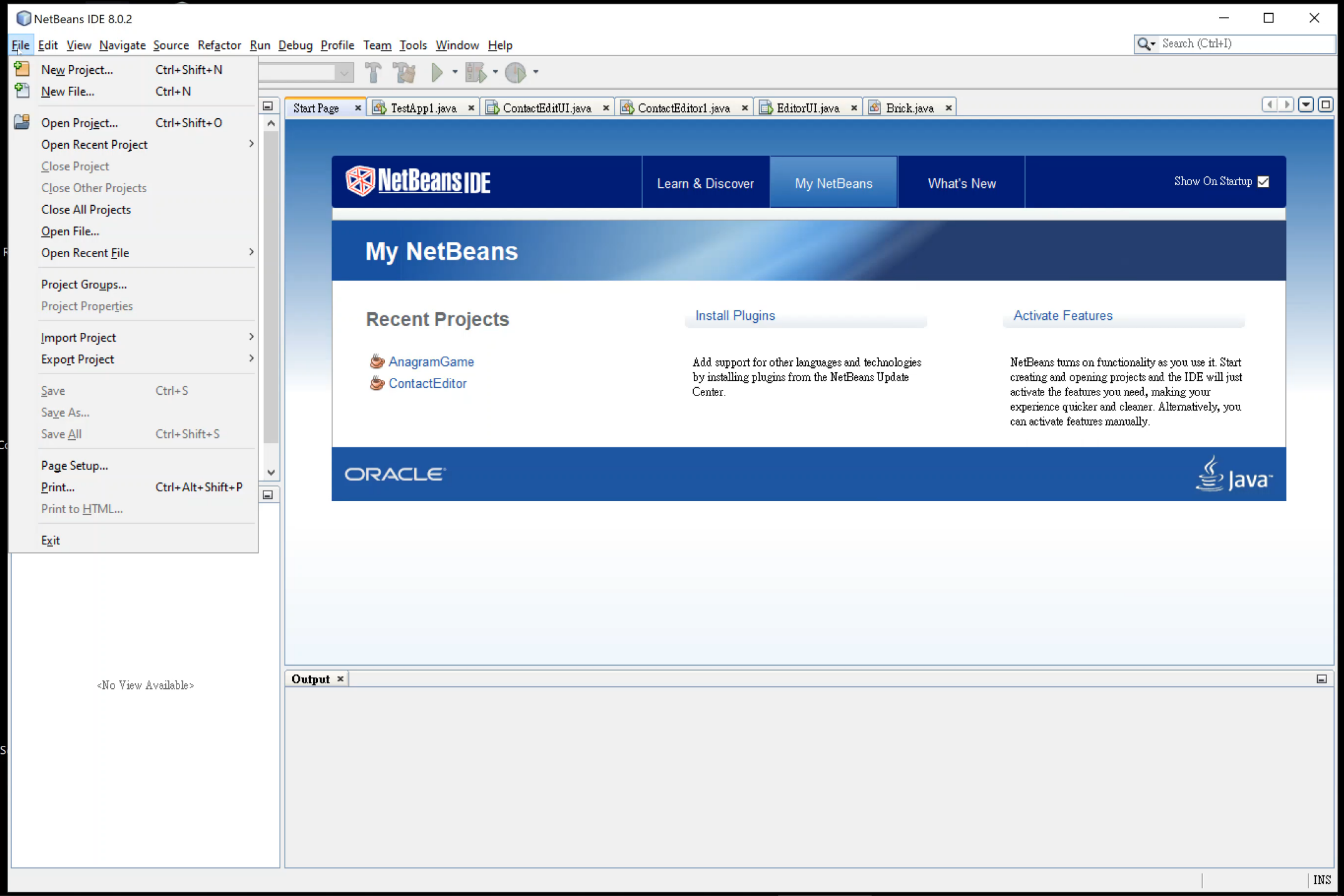
NetBeans

- https://netbeans.org/features/ide/



Install NetBeans

- Download Java SDK 1.8 from Oracle website
- Unzip and add Java SDK 1.8 to system path
- Download NetBeans 8.0
- Unzip and run it!


NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
New Project... Ctrl+Shift+N
New File... Ctrl+N
Open Project... Ctrl+Shift+O
Open Recent Project
Close Project
Close Other Projects
Close All Projects
Open File...
Open Recent File
Project Groups...
Project Properties
Import Project
Export Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Page Setup...
Print... Ctrl+Alt+Shift+P
Print to HTML...
Exit
Start Page
TestApp1.java
ContactEditUI.java
ContactEditor1.java
EditorUI.java
Brick.java
NetBeans IDE
Learn & Discover
My NetBeans
What's New
Show On Startup
My NetBeans
Recent Projects
AnagramGame
ContactEditor
Install Plugins
Add support for other languages and technologies by installing plugins from the NetBeans Update Center.
Activate Features
NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner. Alternatively, you can activate features manually.
ORACLE
Java
<No View Available>
Output
INS

Reference

- https://www.javatpoint.com/java-swing
- https://examples.javacodegeeks.com/desktop-java/ide/intellij-gui-designer-example/
- https://netbeans.org/kb/docs/java/quickstart-gui.html