

# Swing

@M了个J  
李明杰

<https://github.com/CoderMJLee>

<http://cnblogs.com/mjios>



实力IT教育 [www.520it.com](http://www.520it.com)

码拉松



# GUI (Graphical User Interface)

■ GUI (Graphical User Interface) : 图形用户界面

□ 指在计算机中采用图形方式显示的用户界面

■ 1970年代, 美国施乐公司的研究人员开发出了第一个图形用户界面

□ 这一设计使得计算机实现了字符界面向图形界面的转变

□ 后来, 微软、苹果等公司的操作系统陆续出现, 界面设计不断完善

□ 现在日常生活中使用的操作系统、应用程序都是基于 GUI 的



图形界面

```
C:\Users\MJ>ver
Microsoft Windows [版本 10.0.18362.535]

C:\Users\MJ>dir
驱动器 C 中的卷是 系统
卷的序列号是 322E-7DA7
```

字符界面

# Java 的 GUI 编程方案

■ Java 也可以开发 GUI 程序，常见的实现方案有 4 种

□ AWT (Abstract Window Toolkit)

✓ Java 官方最早推出的 GUI 编程开发包，界面风格是跟随操作系统的

□ SWT (Standard Widget Toolkit)

✓ 由 IBM 推出，著名的开发工具 Eclipse 就是用 SWT 开发的

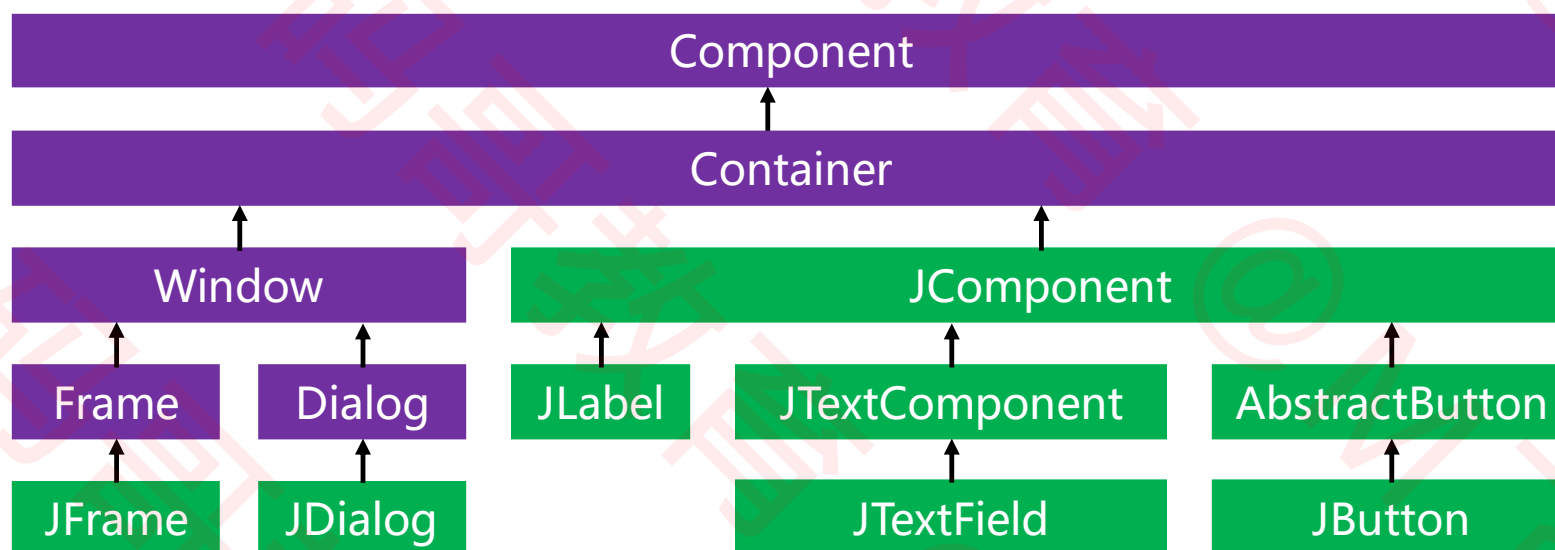
□ Swing

✓ 在 AWT 的基础上扩充了功能，灵活且强大，在不同操作系统中可以保持统一风格

□ JavaFX

✓ Java 官方推出的最新一代 GUI 编程开发包，参考资料要比 Swing 少一些

- 紫色的在 java.awt 包中，绿色的在 javax.swing 包中



# Swing 组件预览

JLabel\_文本标签

JButton\_按钮

JTextField\_文本输入框

.....

JPasswordField

JTextArea\_多行文本输入框

☒ JCheckBox\_Java

☒ JCheckBox\_C++

☐ JRadioButton\_男

☒ JRadioButton\_女

JComboBox\_足球 ▼

JComboBox\_足球

JComboBox\_篮球

JComboBox\_台球

JList\_足球

JList\_篮球

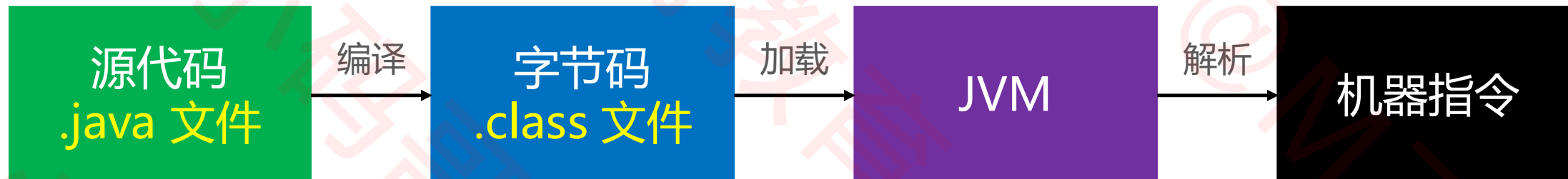
JList\_台球

JDialog\_对话框

网络繁忙，请稍后再试！

# Java 程序的运行过程

■ 思考：Eclipse 是如何将我们开发的 Java 程序运行起来的？



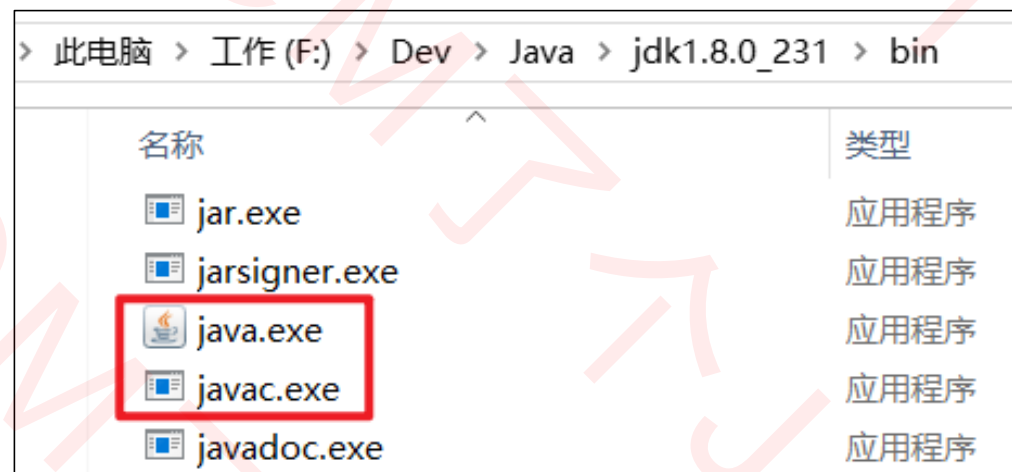
■ JDK 的 bin 目录中已经包含了运行 Java 程序的必备工具（Eclipse 就是调用了这些工具将 Java 程序运行起来的）

□ **javac xx.java**

✓ 编译 xx.java 文件为 xx.class 文件

□ **java xx**

✓ 启动 JVM，加载 xx.class 文件



# .class 文件

- 每一个类（也包括枚举）、接口编译完毕后都会生成一个对应的 .class 文件

类型	.class 文件名
顶级类型（类、接口）	类型
嵌套类型（类、接口）	外部类型 + \$ + 嵌套类型
局部类	外部类型 + \$ + 数字 + 局部类
匿名类	外部类型 + \$ + 数字

# main 方法的参数

```
public class Main {  
    public static void main(String[] args) {  
        try {  
            int n1 = Integer.parseInt(args[0]);  
            int n2 = Integer.parseInt(args[1]);  
            System.out.format("%d + %d = %d\n", n1, n2, n1 + n2);  
        } catch (Exception e) {  
            System.out.println("Please pass two ints.");  
        }  
    }  
}
```

## ■ java Main

□ Please pass two ints.

## ■ java Main 11 22

□ 11 + 22 = 33

## ■ main 方法的限制

□ 必须是公共、静态 (`public`、`static`)

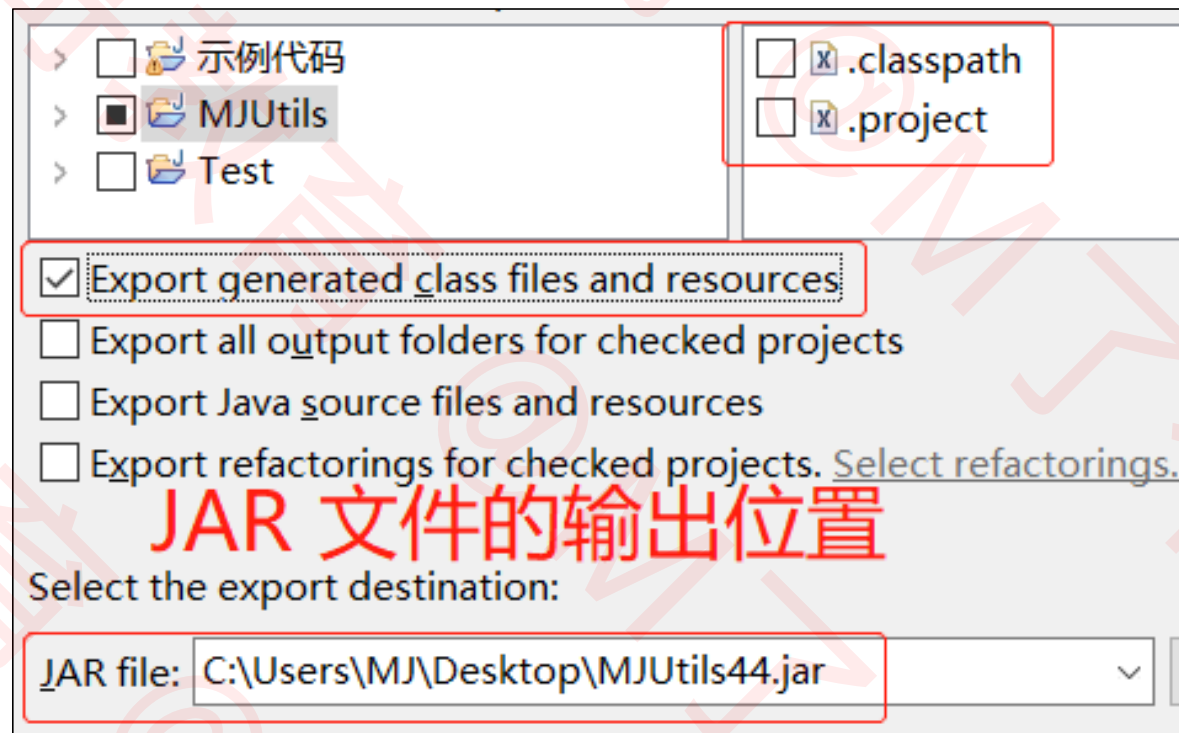
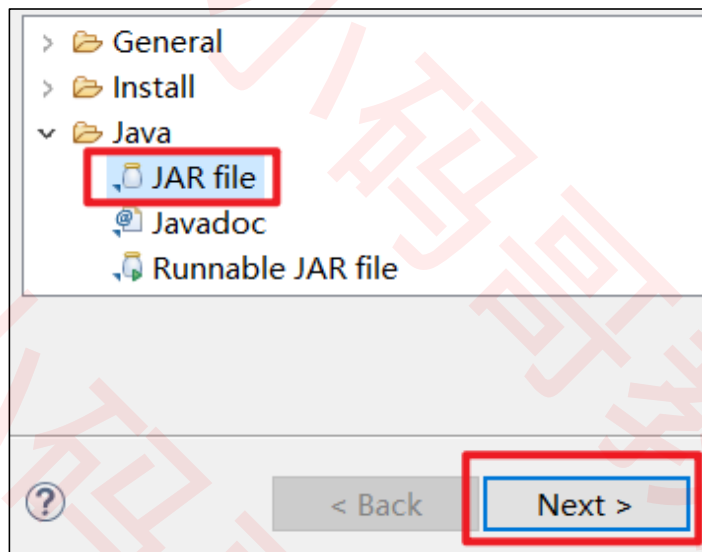
□ 不能有返回值 (`void`)

□ 有且只有 1 个参数，参数类型只能是 `String[]` 或者 `String...`



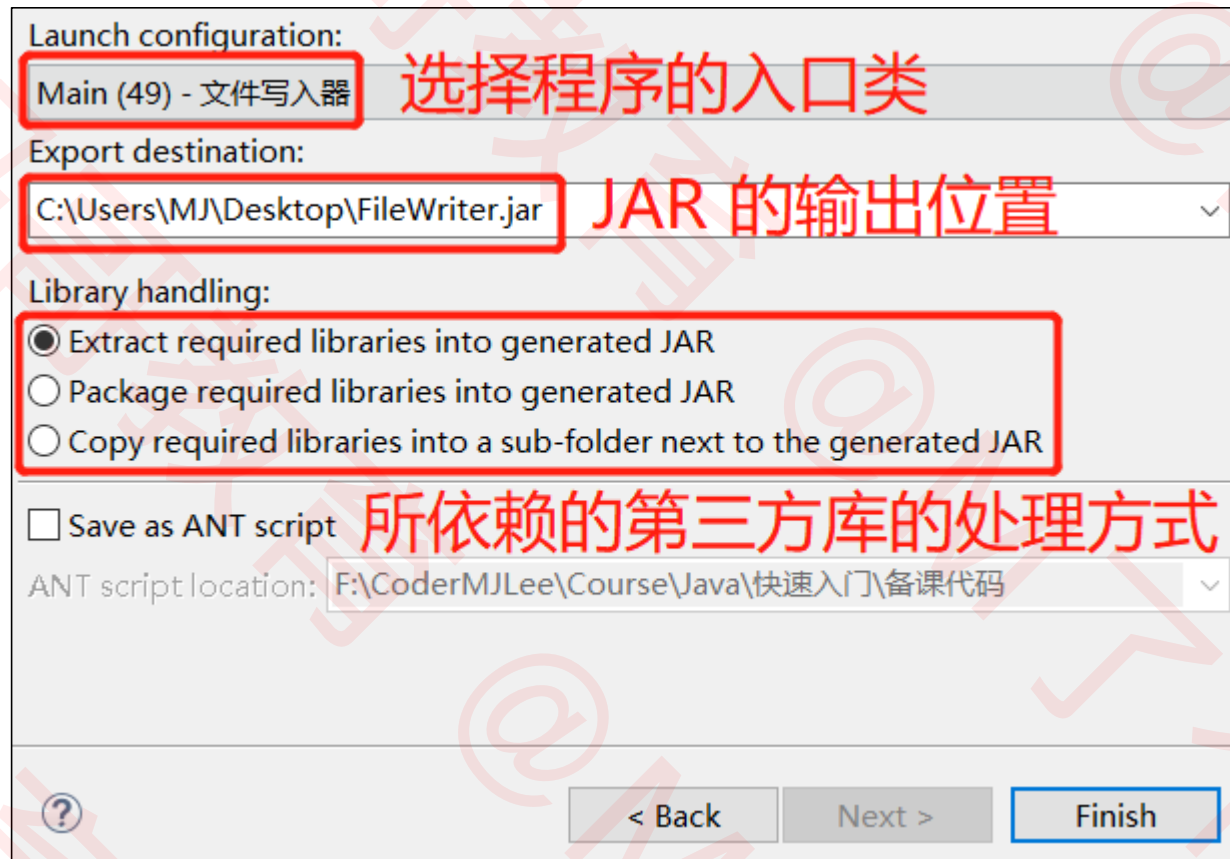
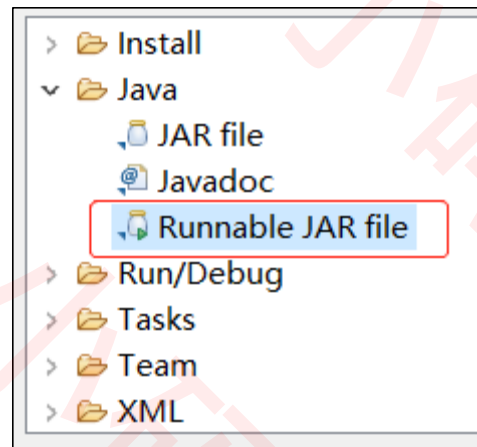
- 如果想将自己写的 Java 代码提供给别人使用，常见做法有
  - 将相关的 .java 文件直接分享出去
  - 将相关的 .class 文件打包成一个 .jar 文件后分享出去
- JAR (美 [dʒɑːr])，全称是 Java Archive (美 [ˈɑːrkaɪv])，使用 ZIP 文件格式来打包
- Eclipse 自带了导出 JAR 包的功能：右击项目 ⇒ Export ⇒ Java ⇒ JAR File
  - 本质上是调用了 JDK bin 目录中的 jar.exe
- 可以对 JAR 进行数字签名，防止代码被篡改

# Eclipse 导出 JAR



# Runnable JAR

■ 可运行 JAR (Runnable JAR) : 包含了程序入口 main 方法的 JAR



# Runnable JAR – 所依赖的第三方库的处理方式

- Extract required libraries into generated JAR
  - 将第三方库的内容解压后放入 JAR 中
  
- Package required libraries into generated JAR
  - 第三方库不解压，直接放入 JAR 中
  
- Copy required libraries into a sub-folder next to the generated JAR
  - 将第三方库放入跟 JAR 同目录的文件夹中（文件夹名默认是：JAR文件名\_lib）

# 启动 Runnable JAR 的常见方式

## ■ 命令行

□ `java -jar JAR文件路径`

## ■ 编写 bat 脚本文件，双击执行 bat 文件（适用于 Windows）

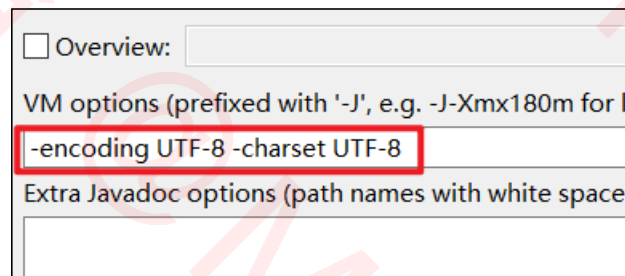
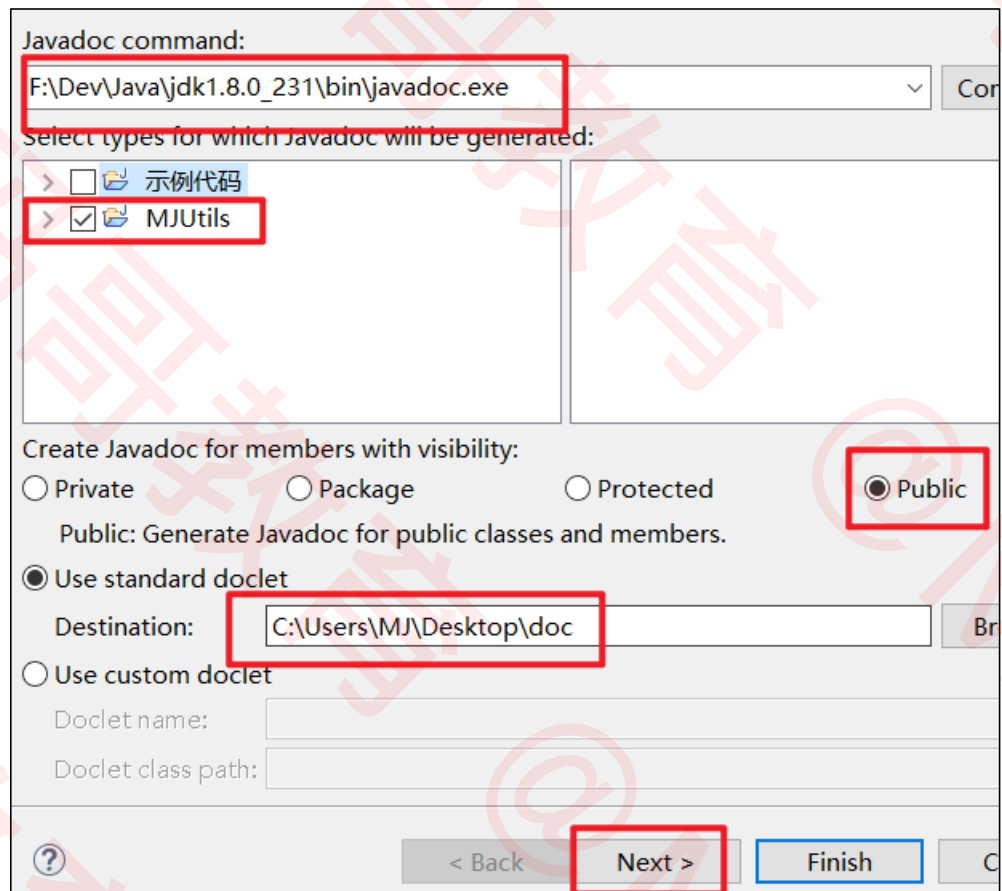
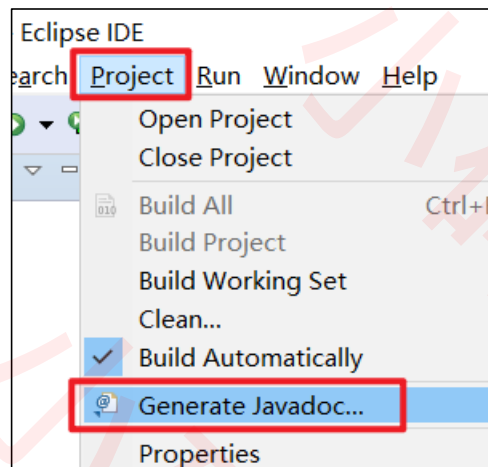
□ bat 文件的内容大概如下所示

```
java -jar JAR文件路径  
pause
```

## ■ 利用工具将 JAR 转为 exe 文件，双击执行 exe 文件（适用于 Windows，依然需要 JVM 的支持）

□ 比如 exe4j：<https://exe4j.apponic.com/download/>

# 生成 Java 文档



■ 本质上是调用了 JDK bin 目录中的 javadoc.exe

# 双括号初始化 (Double Brace Initialization)

```
List<Integer> list = new ArrayList<Integer>() {{  
    add(11);  
    add(22);  
}};  
// [11, 22]  
System.out.println(list);
```

```
Map<String, Integer> map = new HashMap<String, Integer>() {{  
    put("jack", 18);  
    put("rose", 20);  
}};  
// {rose=20, jack=18}  
System.out.println(map);
```

```
public class Person {  
    public void run() {}  
    public void eat() {}  
}
```

```
Person p = new Person() {{  
    run();  
    eat();  
}};
```

# 双括号初始化 (Double Brace Initialization)

- 优点

- 代码简洁

- 缺点

- 本质是多生成了一个匿名类

- 可能会影响 equals 方法的正常使用

- 建议

- 慎重使用