



主讲教师 张 智  
计算机学院软件工程系  
课程群: 421694618

# 1 Java概述

## 1.1 [Java语言简介](#)

## 1.2 [Java编程环境搭建](#)

## 1.3 [Hello World](#)

## 1.4 [Java运行机制](#)

## 1.5 [Java程序示例](#)

## 1.6 [打包Java程序](#)

【[附录](#)】Idea常用快捷键、多个main问题


## 1.1 Java语言简介

- Java是全球多年排名第一的编程语言。
- 常年霸占着三大市场：
  - 互联网和企业应用：体现Java的长期优势和市场地位
  - 大数据平台：Hadoop、Spark、Flink平台主要用Java开发
  - Android移动平台



James Gosling  
Java之父

## Java语言发展历史

- 1990年初,  Sun 公司基于家用电器编程需要, 设计了一种平台独立的语言Oak。
- 随着互联网潮流的兴起, Sun公司预见互联网应用前景, 于1995年5月, Oak经改造并正式更名为Java, 并在SunWorld大会上正式发布Java 1.0版本。Java语言第一次提出了"Write Once, Run Anywhere"口号。
- 1996年, JDK 1.0发布, 10个最主要的操作系统供应商声明将在其产品中嵌入Java技术。
- 同年9月, 已有大约8.3万个网页应用了Java技术来制作。
- Java伴随着互联网的迅猛发展而发展, 逐渐成为重要的网络编程语言。

JDK: Java Development Kit (Java开发包)

- 1998年12月，Java 2企业平台J2EE发布。
- 1999年6月，Java发布三个版本：J2SE、J2EE、J2ME
- 2005年6月，Java SE6.0发布，J2EE更名为Java EE，J2SE更名为Java SE，J2ME更名为Java ME。
- 2006年Sun公司将Java开源，此时JDK变为Open JDK。
- 2009年，ORACLE (甲骨文公司)以74亿美元现金收购Sun公司。Java商标从此正式归Oracle所有。
- 2011年7月，Oracle公司发布Java SE 1.7。
- 2014年3月，Oracle公司发布JDK8发布，这是一个里程碑的版本。
- 2017年，Oracle不允许开源组织使用Java名号。
- 2019年，Oracle发布JDK 8u211和8u212两个版本(属于JDK8系列)，并从这两个版本开始商用收费。
- 2022年，Oracle JDK18发布。

## Java的3个版本

### 1. 标准版：Java SE

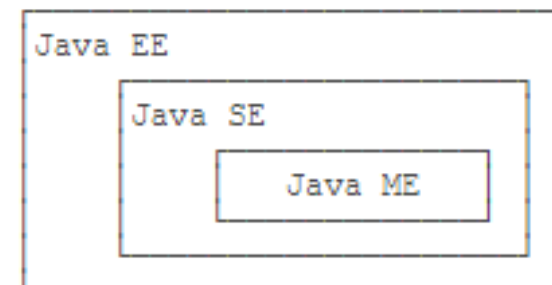
- 用于开发和部署在桌面、服务器等环境中的应用程序

### 2. 企业版：Java EE （ 俗称 J2EE ）

- 用于开发企业级Web应用程序

### 3. 微型版：Java ME → 已经被Android取代

- 用于移动设备和嵌入式设备程序开发



## Java语言特点

Write Once, Run Anywhere

- **平台无关性**：采用**虚拟机(JVM)**技术实现平台无关性，**是真正的跨平台可移植**。
- **面向对象**：具有良好的封装、继承、多态性，实现代码重用。
- **解释性**：采用预编译将源程序生成**.class字节码**，减轻运行时解释工作。
- **简单**：**不需要进行指针运算和存储器管理**，简化设计，减少出错可能性。
- **健壮安全**：**拒绝执行非法的内存访问**，超越权限的访问等。
- **多线程**：允许一个应用程序同时做**多个任务**。
- **分布式**：拥有网络协议对象库，可以象访问本地文件一样访问Internet上的对象。
- **动态性**：允许下载代码模块，当程序运行时也能动态升级。
- **高性能**：经过实际的综合评测得出结论，Java是高性能的。

[【返回】](#)

## 1.2 Java编程环境搭建

- 安装IntelliJ IDEA：推荐Ultimate英文版(不要汉化)
- 在线安装JDK：自动安装（推荐）
- 离线安装JDK：手工安装（了解）

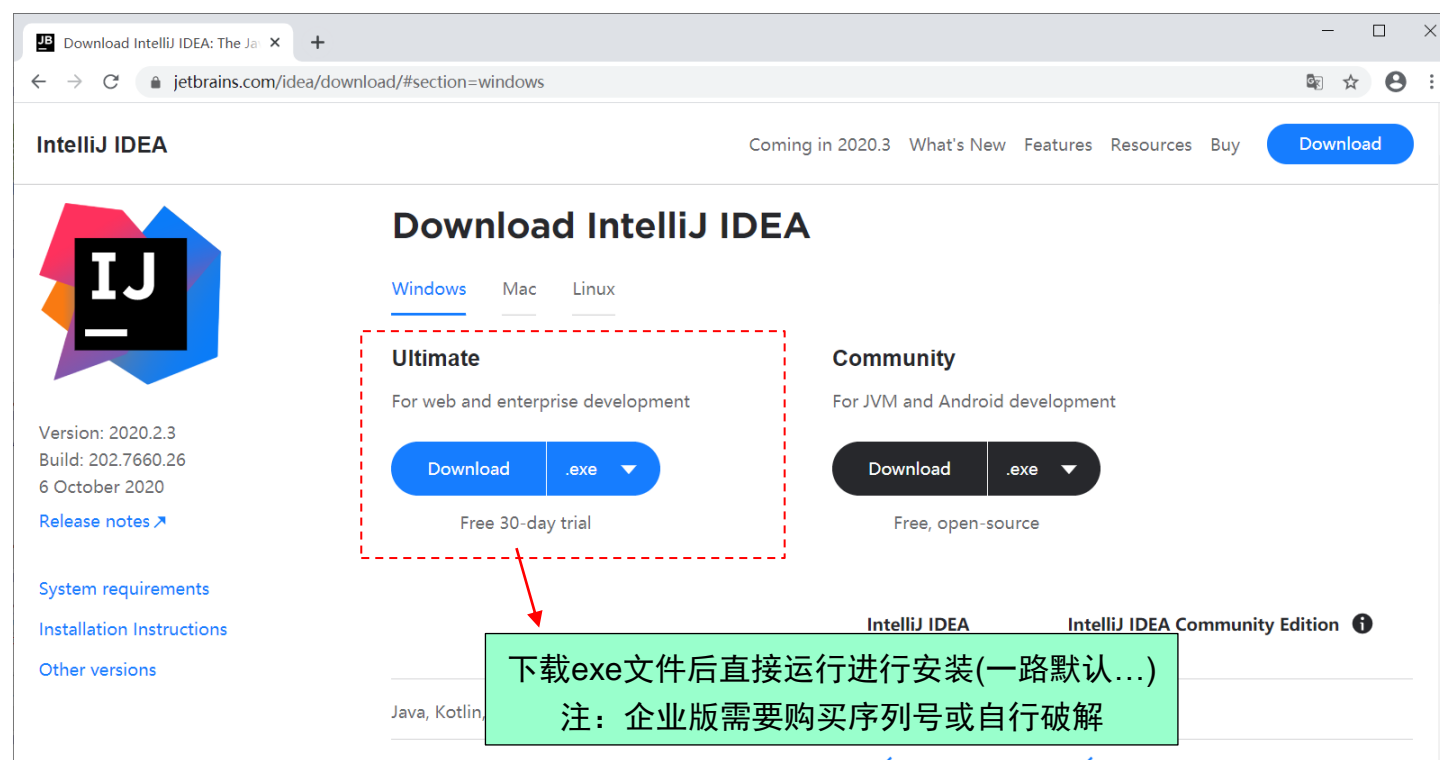
### 基本概念：

- IDE：Integrated Development Environment，集成了代码编写、编译、调试等功能的程序开发环境的应用程序，如IntelliJ Idea
- JDK：Java开发包 (Java Development Kit)，包含了Java运行环境(JRE)和开发工具(编译器，调试器，javadoc等)



## 1. 安装IntelliJ IDEA

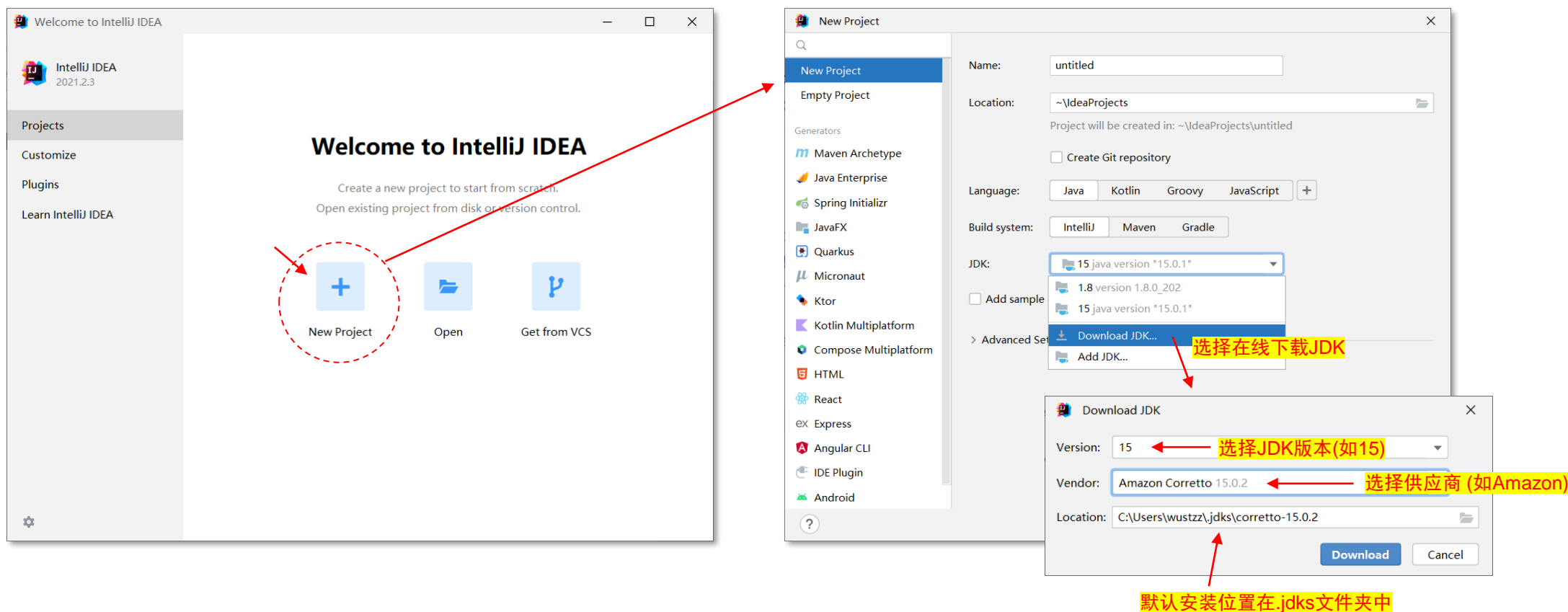
- 官网: <https://www.jetbrains.com/idea/download/#section=windows>



下载exe文件后直接运行进行安装(一路默认...)  
注: 企业版需要购买序列号或自行破解

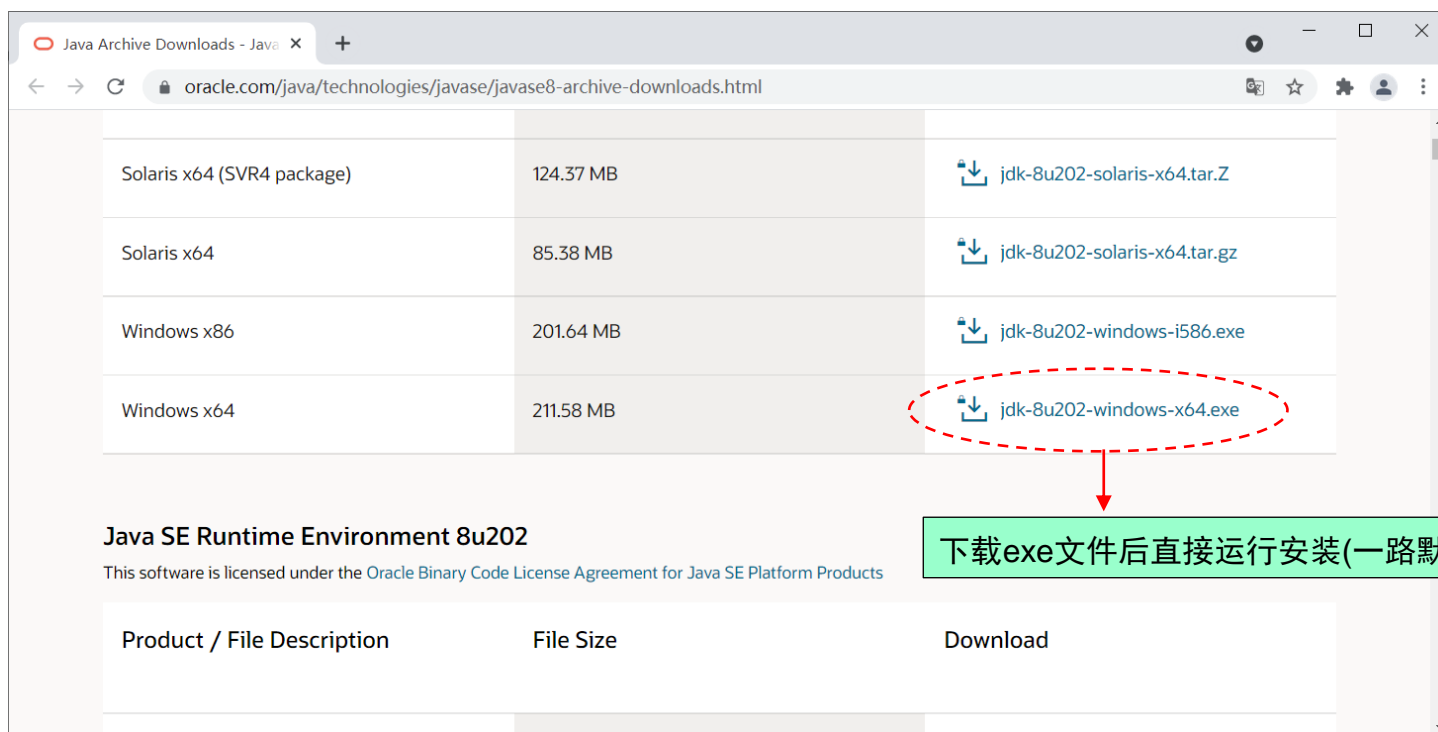
## 2. 在线安装JDK

■ 启动Idea → 点击 new project



### 3. 手工安装JDK (以JDK 1.8.0\_202为例)

- 官网下载: <https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html>

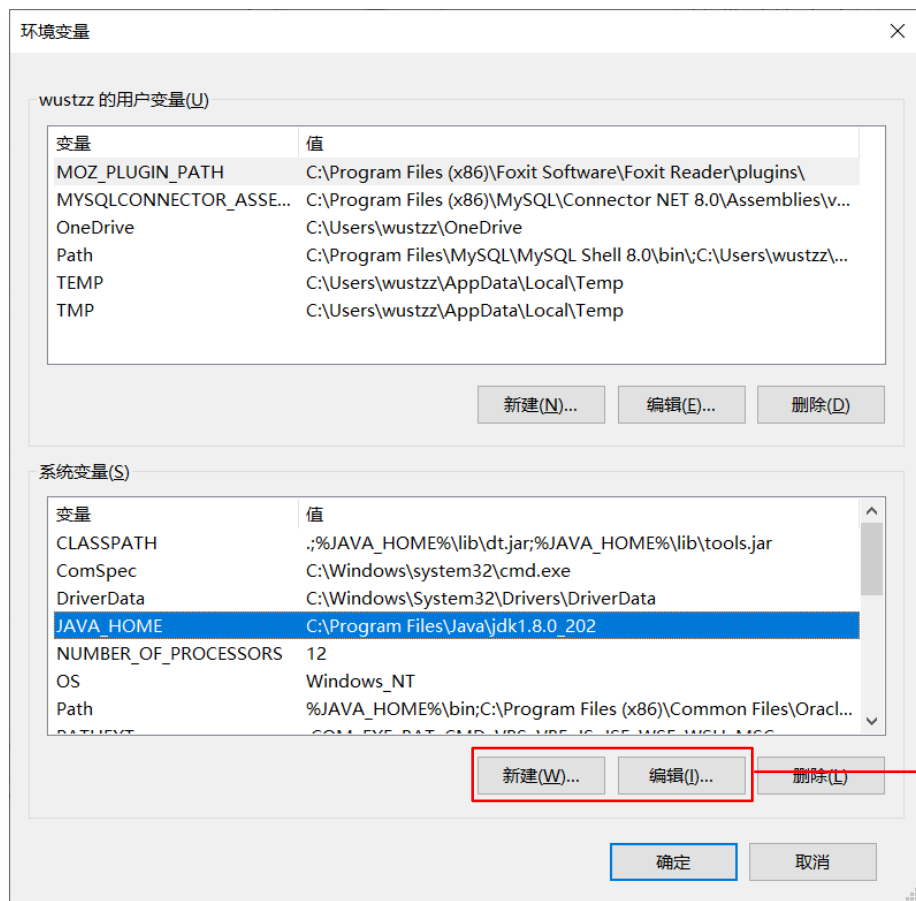


Product / File Description	File Size	Download
Solaris x64 (SVR4 package)	124.37 MB	<a href="#">jdk-8u202-solaris-x64.tar.Z</a>
Solaris x64	85.38 MB	<a href="#">jdk-8u202-solaris-x64.tar.gz</a>
Windows x86	201.64 MB	<a href="#">jdk-8u202-windows-i586.exe</a>
Windows x64	211.58 MB	<a href="#">jdk-8u202-windows-x64.exe</a>

**Java SE Runtime Environment 8u202**  
This software is licensed under the Oracle Binary Code License Agreement for Java SE Platform Products

下载exe文件后直接运行安装(一路默认...)

## 手工配置 Java 环境变量

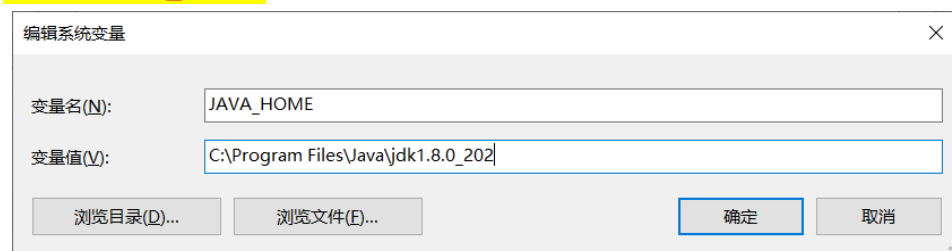


新建或编辑系统环境变量(参数见下页)

## Java8环境变量

- 新建：变量名：JAVA\_HOME 值： C:\Program Files\Java\jdk1.8.0\_202
- 新建：变量名：CLASSPATH 值： .;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar
- 编辑：变量名：PATH, 添加新值： %JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin

### ① 新建JAVA\_HOME



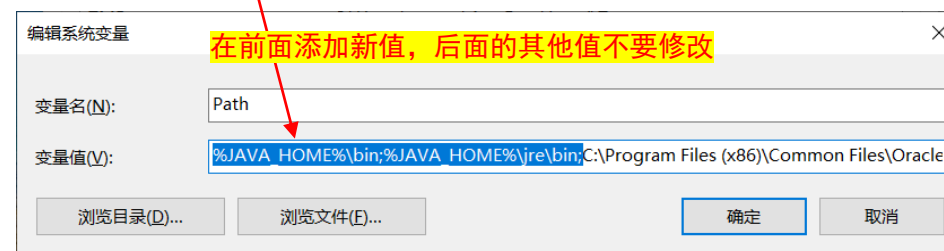
编辑系统变量

变量名(N): JAVA\_HOME

变量值(V): C:\Program Files\Java\jdk1.8.0\_202

浏览目录(D)... 浏览文件(F)... 确定 取消

### ③ 修改PATH



编辑系统变量

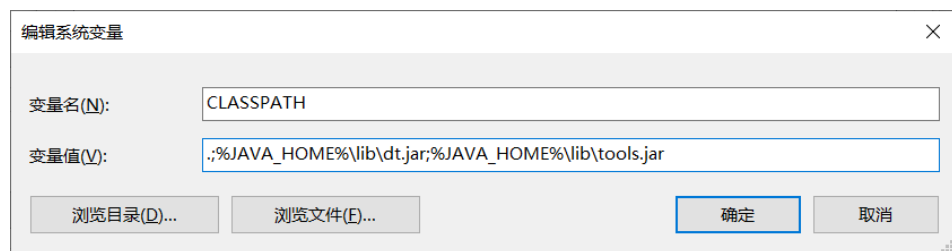
变量名(N): Path

变量值(V): %JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;C:\Program Files (x86)\Common Files\Oracle

浏览目录(D)... 浏览文件(F)... 确定 取消

在前面添加新值，后面的其他值不要修改

### ② 新建CLASSPATH



编辑系统变量

变量名(N): CLASSPATH

变量值(V): .;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar

浏览目录(D)... 浏览文件(F)... 确定 取消

## 检测Java环境是否配置成功

- 打开cmd窗口，分别输入命令行：`java -version` 和 `where java`

```
命令提示符
Microsoft Windows [版本 10.0.18363.1198]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\wustzz>java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)

C:\Users\wustzz>where java
C:\Program Files\Java\jdk1.8.0_202\bin\java.exe
C:\Program Files\Java\jdk1.8.0_202\jre\bin\java.exe
C:\Program Files (x86)\Common Files\Oracle\Java\javapath\java.exe

C:\Users\wustzz>
```

显示Java版本

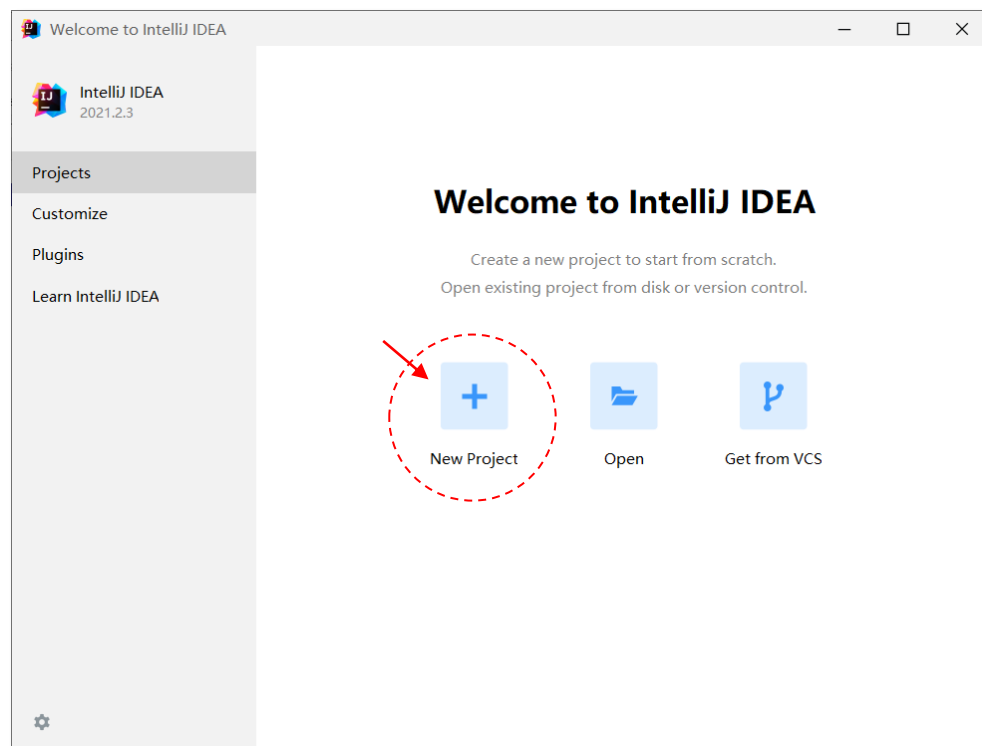
显示Java安装路径

能正确显示表明手工安装成功

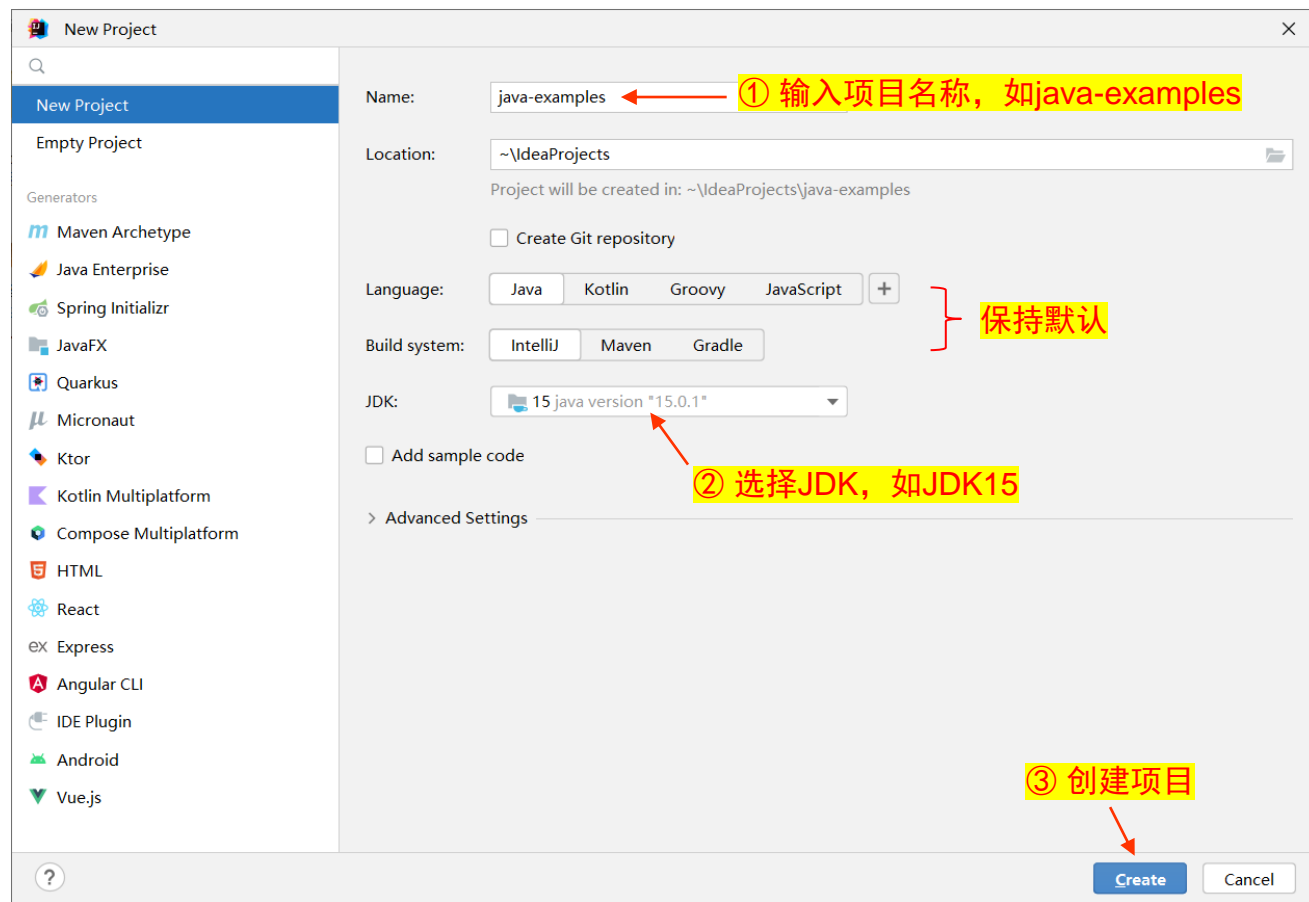
[【返回】](#)

## 1.3 Hello World

■ 启动Idea → 点击 new project

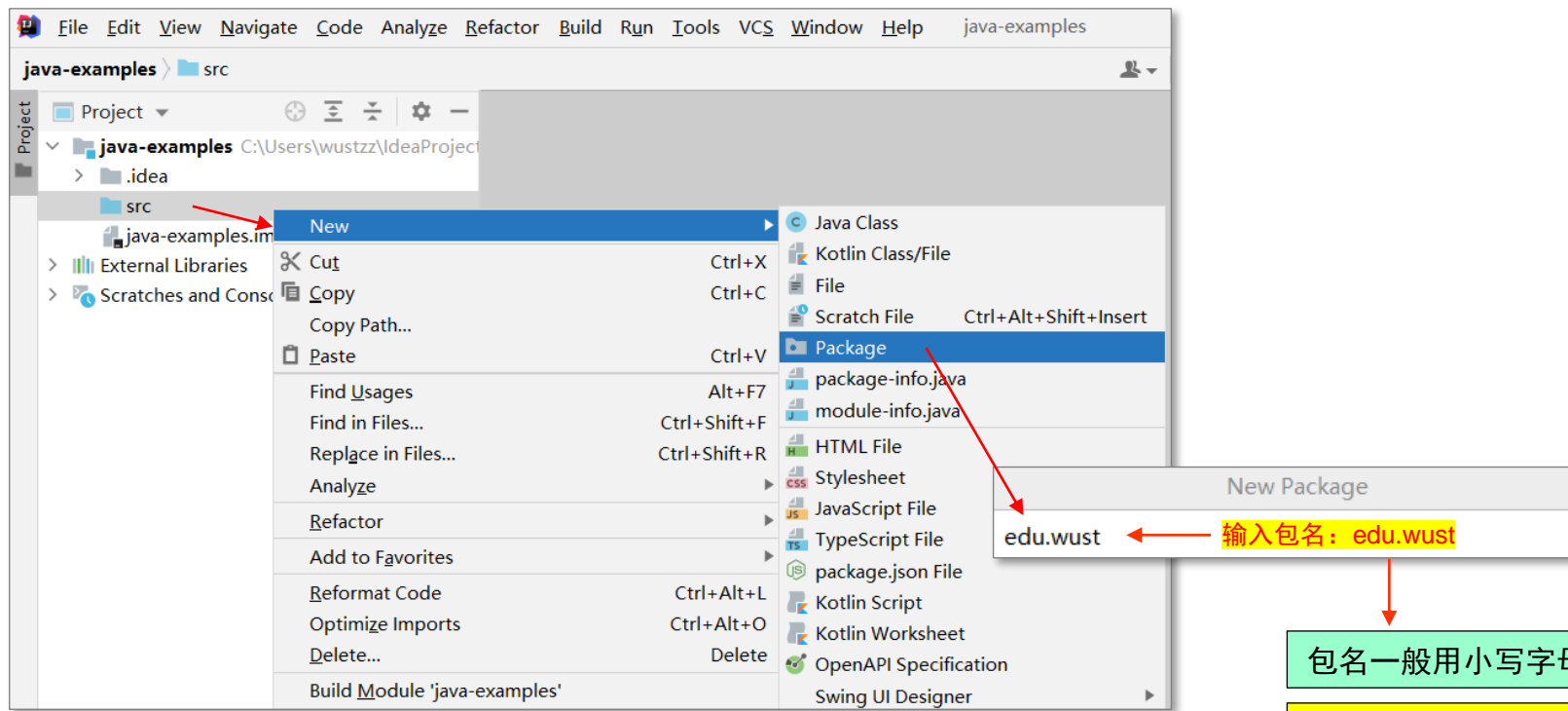


## ■ 选择New Project:





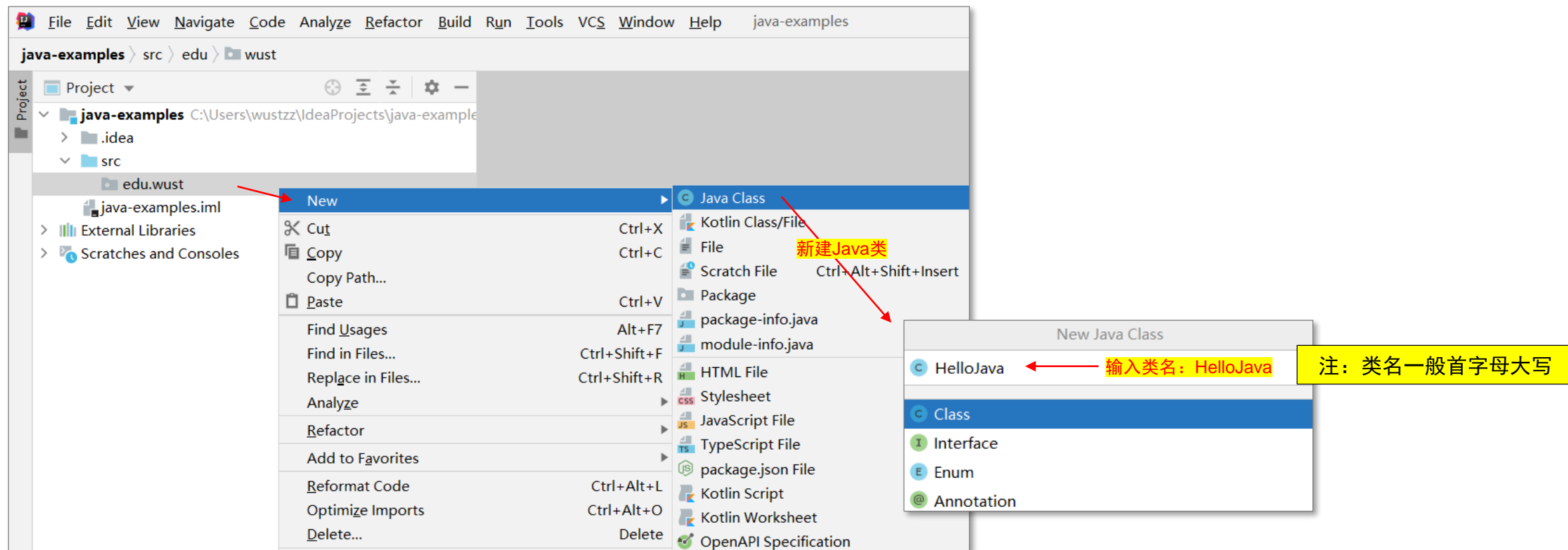
- 在 src 文件夹右键 → New → Package(包) → 新建包名为: edu.wust



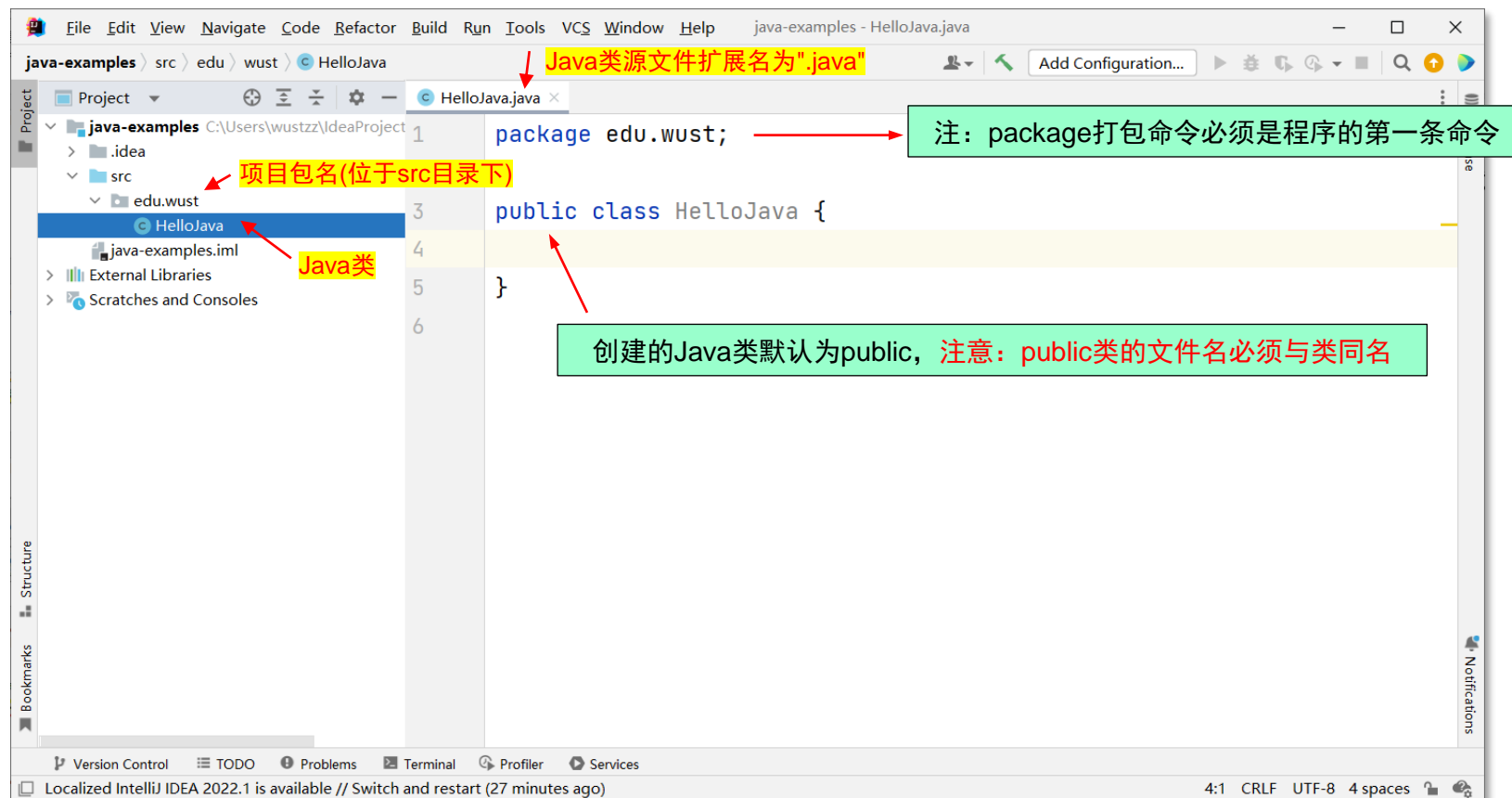
包名一般用小写字母命名

注: edu.wust实际是由"edu"和"wust"两个包组成, 包名用"."连接

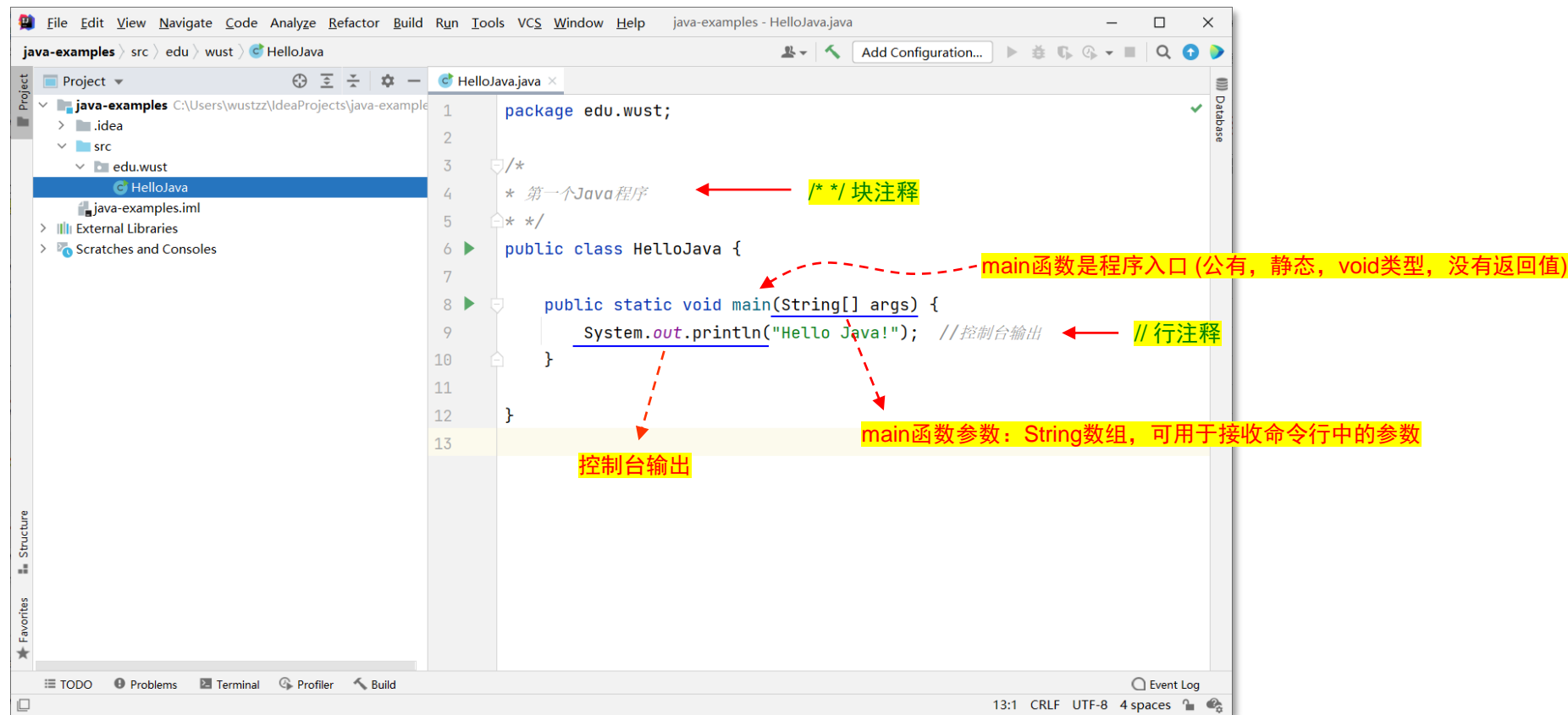
- 在 edu.wust 包右键→ New → Class(类) → 新建类名为：HelloJava



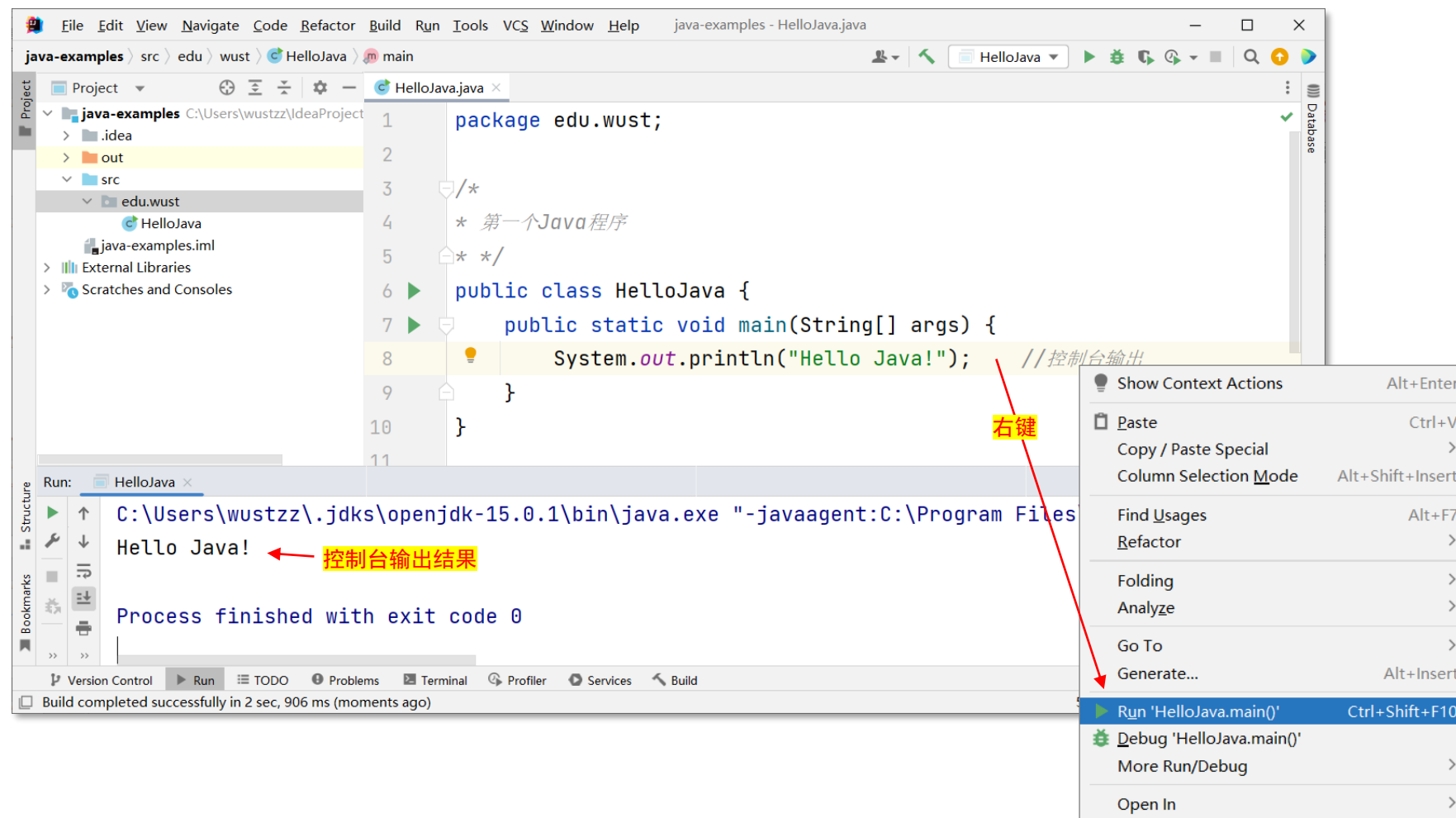
# HelloJava类



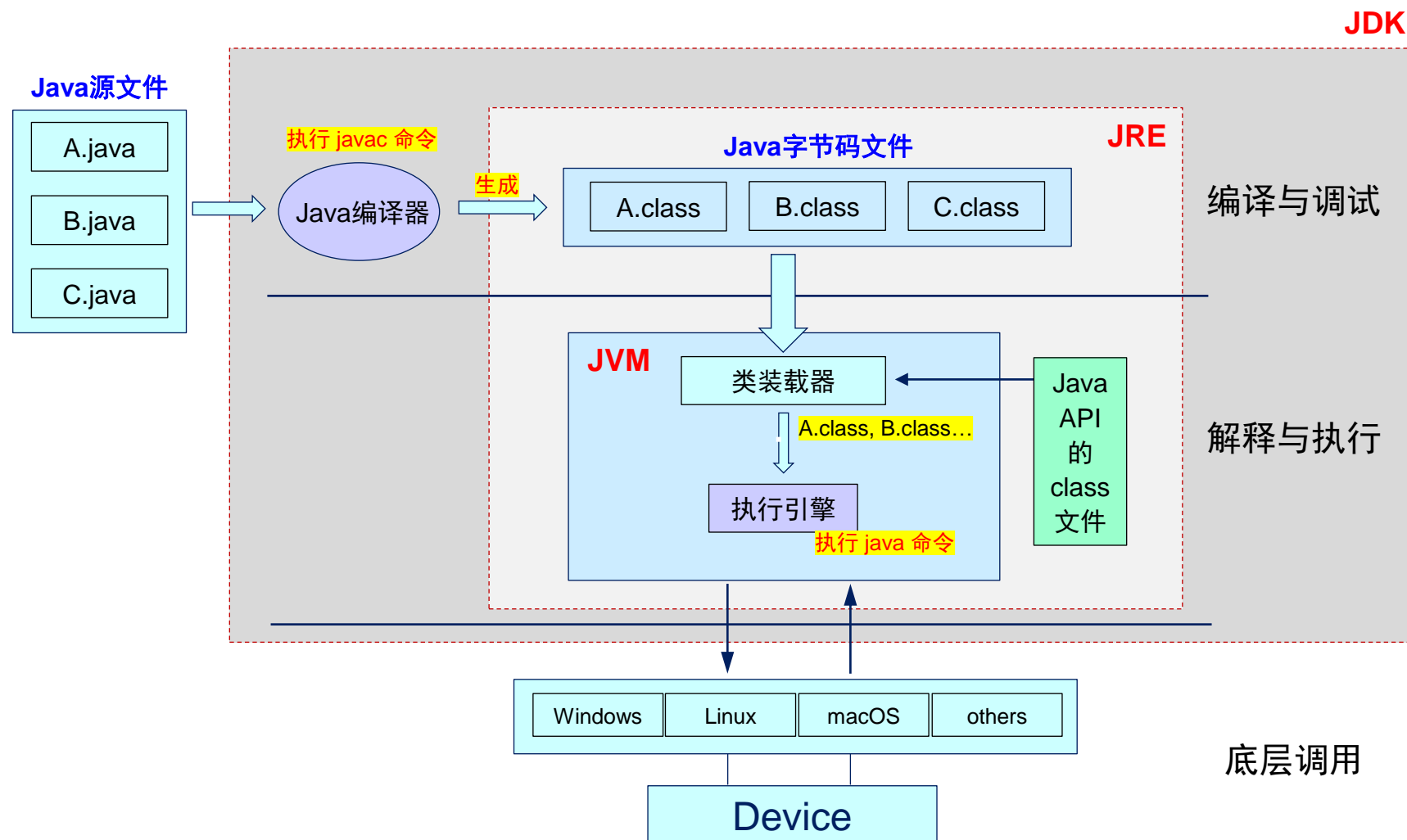
## 在类中添加main入口函数：






## 运行程序 -- 初次运行时

[【返回】](#)

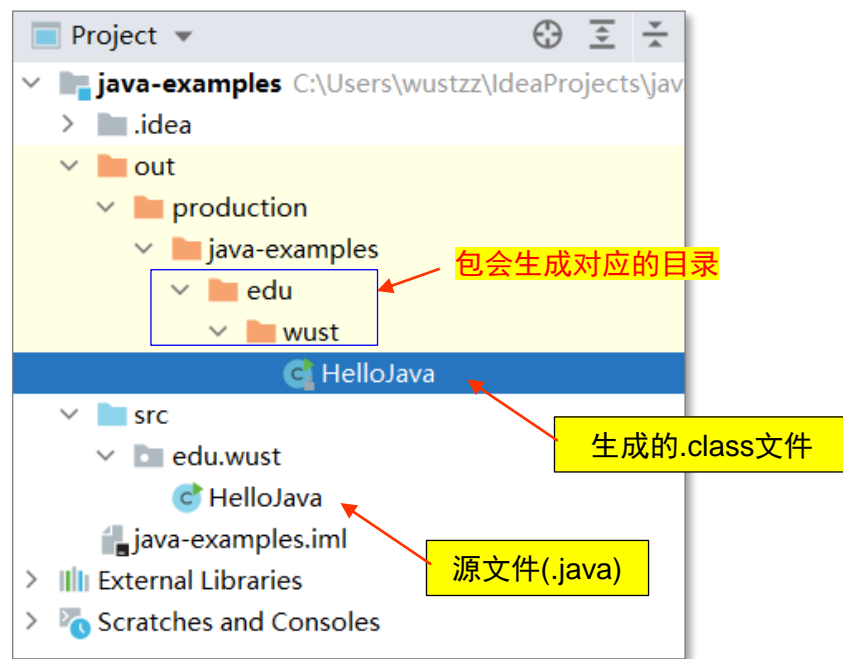
## 1.4 Java运行机制



## (1) Java程序的编译与运行

- 编译程序: `javac HelloJava.java`  编译时要带扩展名, 编译成功会生成同名.class文件  称为字节码文件
- 运行程序: `java HelloJava`  运行字节码文件, 文件名区分大小写, 不加".class"

## 生成的.class文件

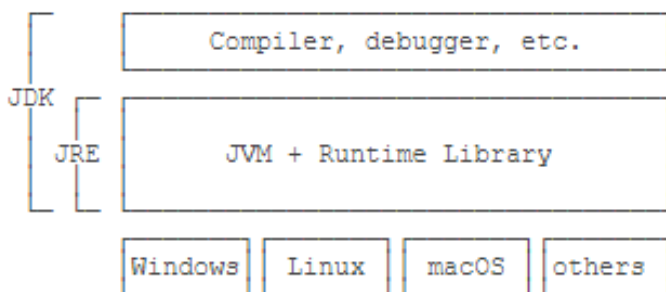




## (2) JVM、JRE、JDK概念

### ■ JVM: Java虚拟机 (Java Virtual Machine)

- JVM是一个**虚拟的计算机**，它有一套完整的体系架构，包括处理器、堆栈、寄存器等
- JVM可以理解成一个以**字节码(class文件)**为机器指令的CPU  
JVM的指令序列
- JVM屏蔽了底层运行平台的差别，实现了"Write Once, Run Anywhere"



## ■ JRE: Java运行时 (Java Runtime Environment)

- JRE为Java提供了运行环境，其中重要的一环就是通过JVM将字节码解释成可执行的机器码
- JRE由JVM, Java运行时类库等组成

## ■ JDK: Java开发包 (Java Development ToolKit)

- JDK包含了Java运行环境(JRE)和开发工具(编译器, 调试器, javadoc等)
- 开发人员依靠JDK来开发和运行Java程序

[【返回】](#)

## 1.5 Java程序示例

- [控制台输入输出程序](#)
- [多个类的程序](#)
- [Java程序调试](#)

**[【返回】](#)**

## 1. 控制台输入输出程序 -- 输入一个数据

HelloJava.java

```
package edu.wust;  
  
import java.util.Scanner; //导入Scanner类  
  
public class HelloJava {  
  
    private final static double VERSION = 1.0; //定义一个类成员 (静态常量)  
  
    public static void main(String[] args) {  
  
        System.out.println( "版本号=" + VERSION ); // "+"为字符串连接, 能自动将后面的数据转换为字符串  
  
        System.out.print("请输入圆的半径=");  
        Scanner sc = new Scanner( System.in ); //创建Scanner对象 (读取键盘缓冲区)  
        double r = sc.nextDouble(); //从键盘缓冲区读取一个double数据, 回车结束  
  
        System.out.println("计算圆面积=" + String.format("%.2f", Math.PI*r*r) );  
    }  
}
```

运行结果

```
版本号=1.0  
请输入圆的半径=10  
计算圆面积=314.16
```

String.format用于格式化数据

String、Math类均来自java.lang包, 可直接使用, 无需导入

## 读取数据的常用方法

- 读取数值：nextInt() 、 nextByte()、 nextLong()、 nextFloat()、 nextDouble()
- 读取boolean数据：nextBoolean() 输入true或false
- 读取一行作为字符串(以回车结束)：nextLine()

## 一次输入多个数据

```
public class HelloJava {  
    public static void main(String[] args) {  
        System.out.println("请输入若干数，#结束输入");  
  
        double sum = 0;  
  
        int count = 0;  
  
        Scanner sc = new Scanner( System.in );  
  
        while ( sc.hasNextDouble() ) { ← hasNextDouble(): 判断是否还有下一个double数据  
            sum += sc.nextDouble(); ← 推荐用回车分割数据(空格也行)  
            count++;  
        }  
  
        System.out.println( String.format( "%d个数的平均值=%.2f", count, sum/count ) );  
    }  
}
```

请输入若干数，#结束输入

1.2

3.4

5

#

3个数的平均值=3.20

[【返回】](#)

## 2. 多个类的程序

涉及访问权限问题

- Java程序都是以类来组织，类名前可加public也可不加 (推荐加public)
- 一个源文件中只能含有一个标记为public的类，加了public的类其源文件名必须跟这个类的名字相同
- 一个源文件也可包含多个类 (不推荐，建议一个类单独保存为一个文件)，每个类编译完成后会生成各自的字节码文件 (.class)
- 每个Java类都可有一个public static void main(String[] args)作为程序入口 (有且只有一个)

# 示例1：多个类写在一个源文件 (不建议)

```
package edu.wust;
```

```
class A {
```

```
    private int value;    // 私有成员变量
```

```
    public A() {          // 默认构造函数
```

```
    }
```

```
    public A(int value) { // 自定义构造函数
```

```
        this.value = value;
```

```
    }
```

```
    public void foo() {    // 公有方法
```

```
        System.out.println("value=" + value);
```

```
    }
```

@Override注解：表示重写父类方法

```
@Override
```

toString()方法是继承Object类的方法，Object是Java根对象

```
public String toString() { //重写toString方法
```

```
    return "[value=" + value + "];"
```

```
    }
```

```
}
```

一个源文件只能有一个public类

```
public class HelloJava {
```

```
    public static void main(String[] args) {
```

```
        A a1 = new A();    // 实例化
```

```
        a1.foo();          // a1.value=0
```

```
        A a2 = new A(100); // 实例化
```

```
        System.out.println(a2); // 输出对象时会自动调用该对象的toString()方法
```

```
    }
```

```
}
```

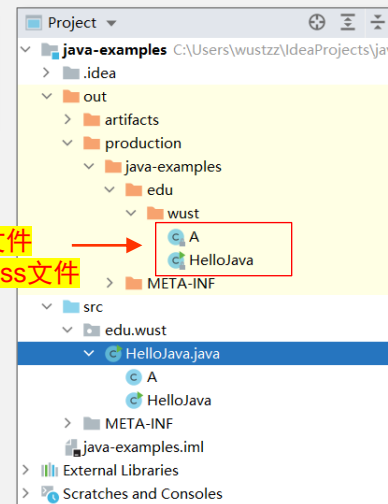
源文件名必须跟public类同名

HelloJava.java

运行结果

```
value=0
[value=100]
```

虽然只有一个源文件  
但每个类都会生成.class文件



思考：给A类加public会怎样？



## 示例2：一个类一个源文件（推荐）

A.java

```
package edu.wust;
```

```
public class A {
```

```
    private int value;    // 私有成员变量
```

```
    public A() {          // 默认构造函数
```

```
}
```

```
    public A(int value) { // 自定义构造函数
```

```
        this.value = value;
```

```
}
```

```
    public void foo() {    // 公有方法
```

```
        System.out.println("value=" + value);
```

```
}
```

```
@Override
```

```
    public String toString() { // 重写toString方法
```

```
        return "[value=" + value + "];"
```

```
}
```

```
}
```

HelloJava.java

```
package edu.wust;
```

```
public class HelloJava {
```

```
    public static void main(String[] args) {
```

```
        A a1 = new A();    // 实例化
```

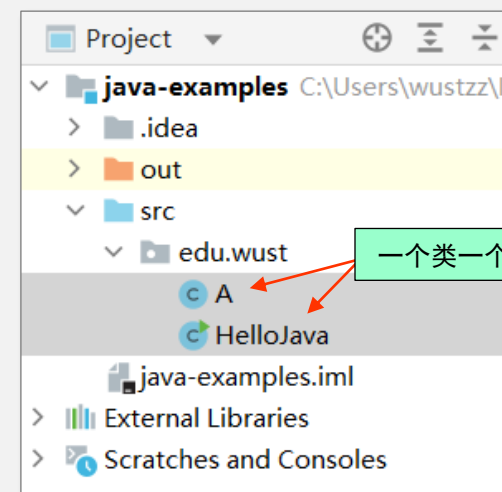
```
        a1.foo();          // a1.value=0
```

```
        A a2 = new A(100); // 实例化
```

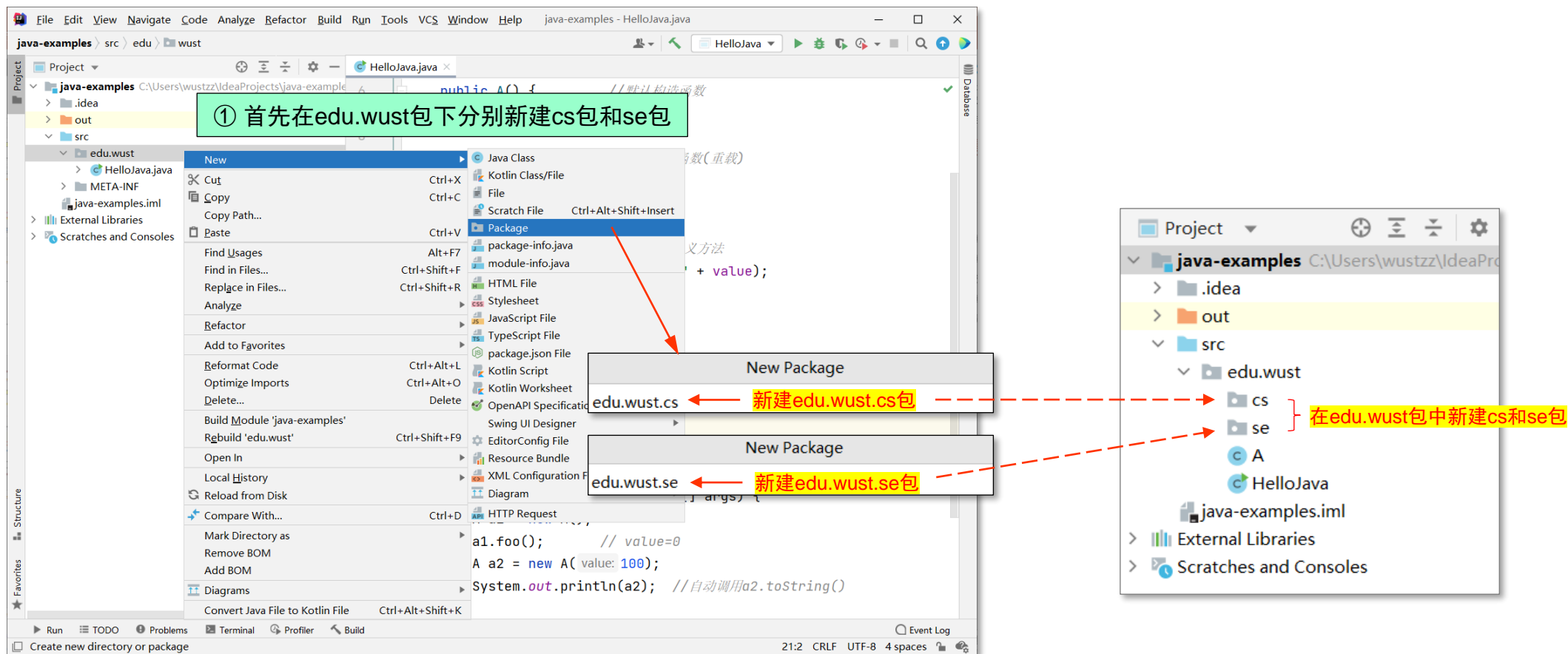
```
        System.out.println(a2); // 输出对象时会自动调用该对象的toString()方法
```

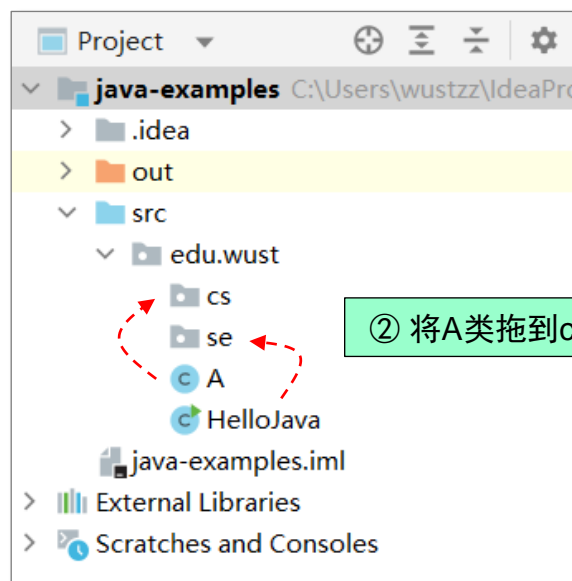
```
}
```

```
}
```

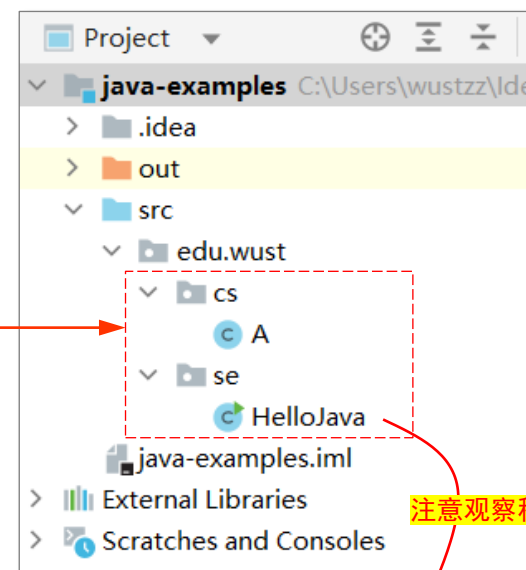


## 示例3: Java类在不同包中





② 将A类拖到cs包中，将HelloJava类拖到se包中

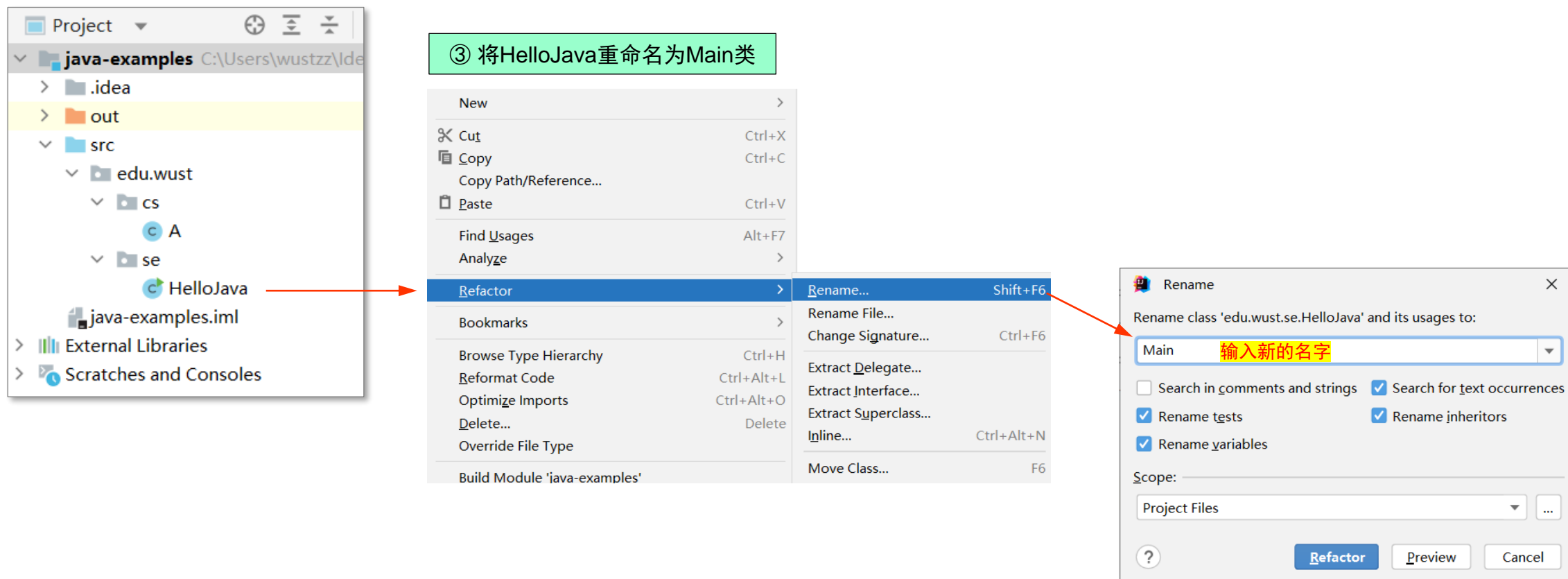


注意观察程序的变化（以HelloJava为例）

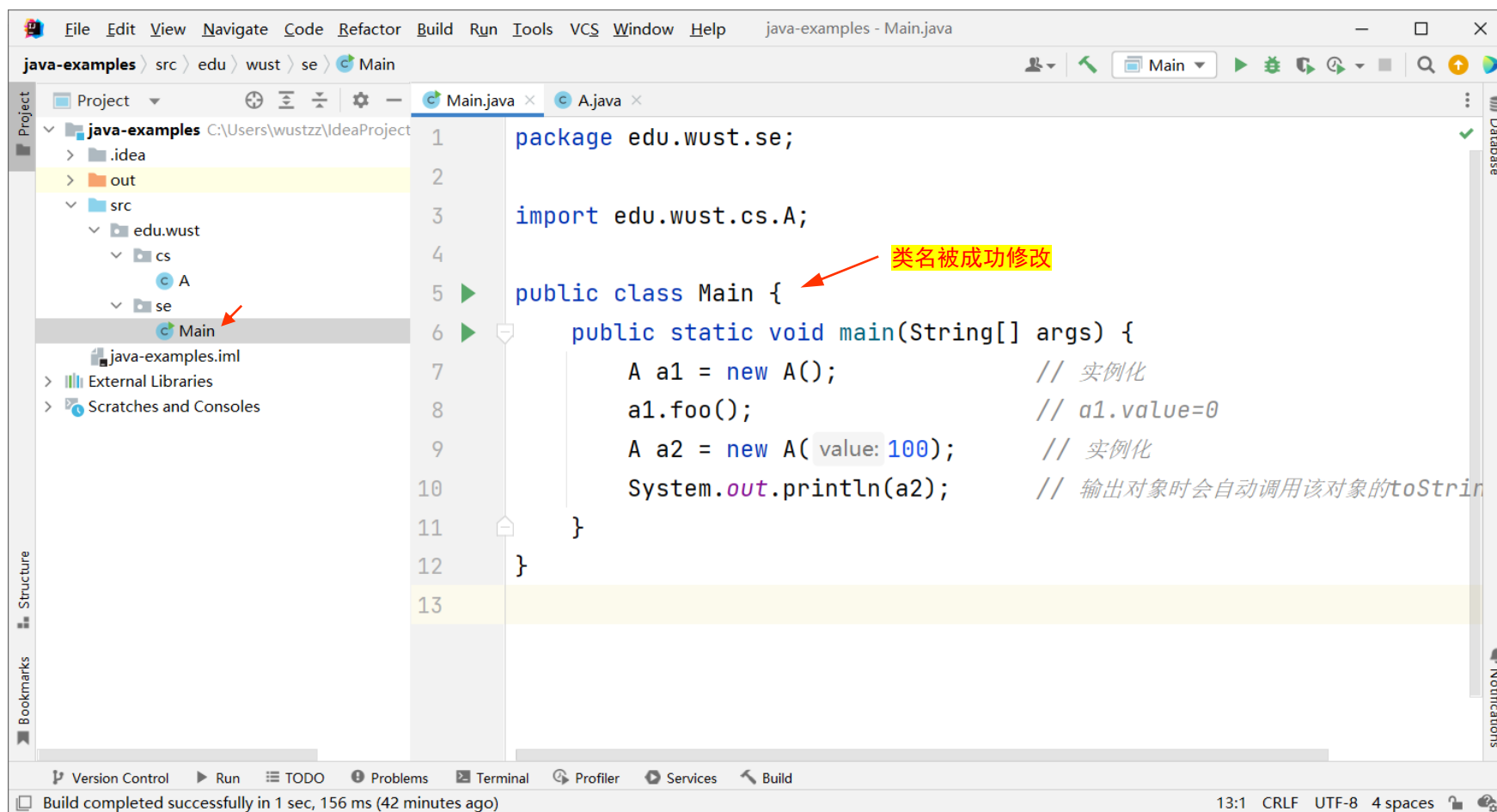
变化1: 打包命令有变化

变化2: 增加了导入A类的命令

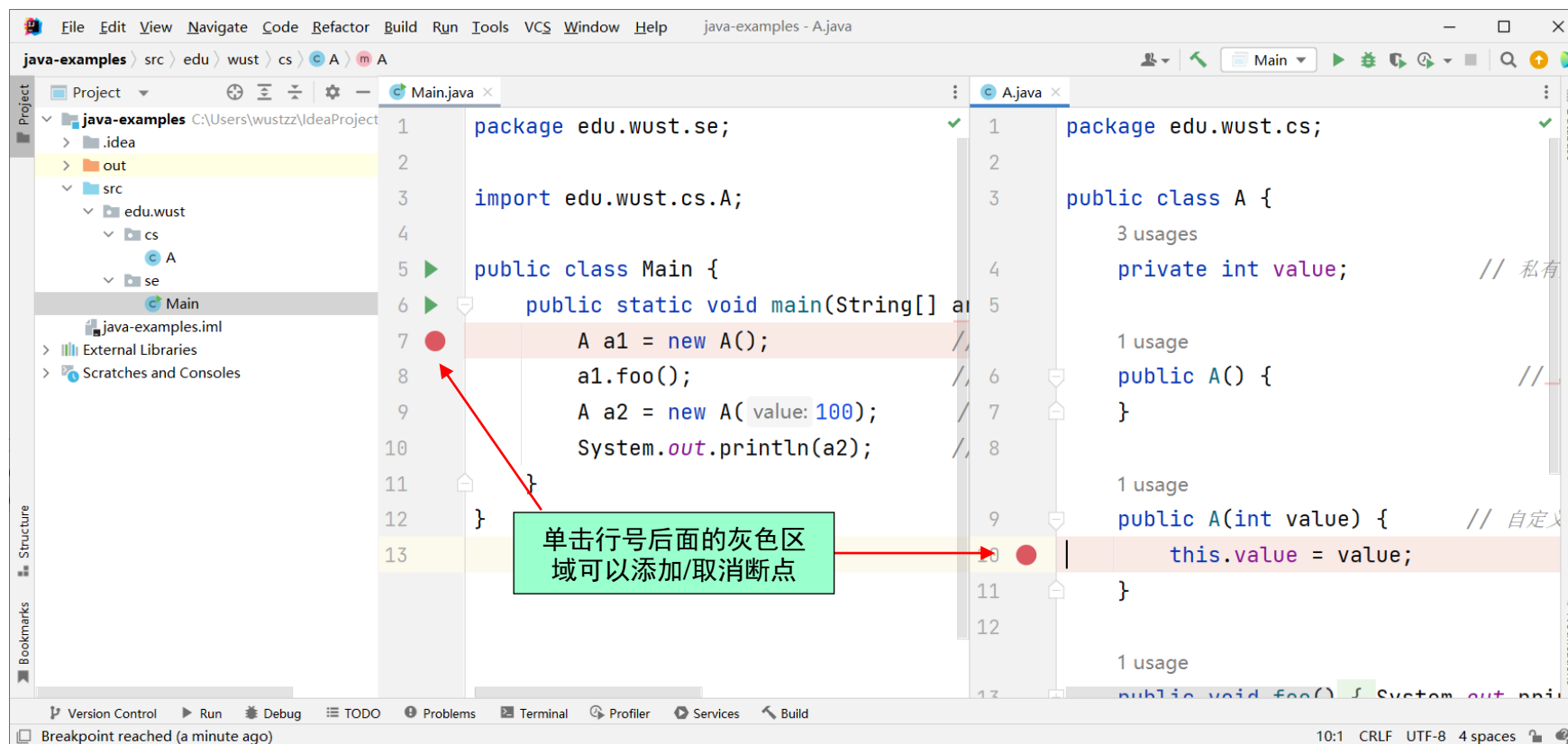
```
1 package edu.wust.se;
2
3 import edu.wust.cs.A;
4
5 public class HelloJava {
6     public static void main(String[] args) {
7         A a1 = new A();           // 实例化
8         a1.foo();                 // a1.value=0
9         A a2 = new A( value: 100); // 实例化
10        System.out.println(a2);   // 输出对象时会自动调用该对象的toString()方法
11    }
12 }
```



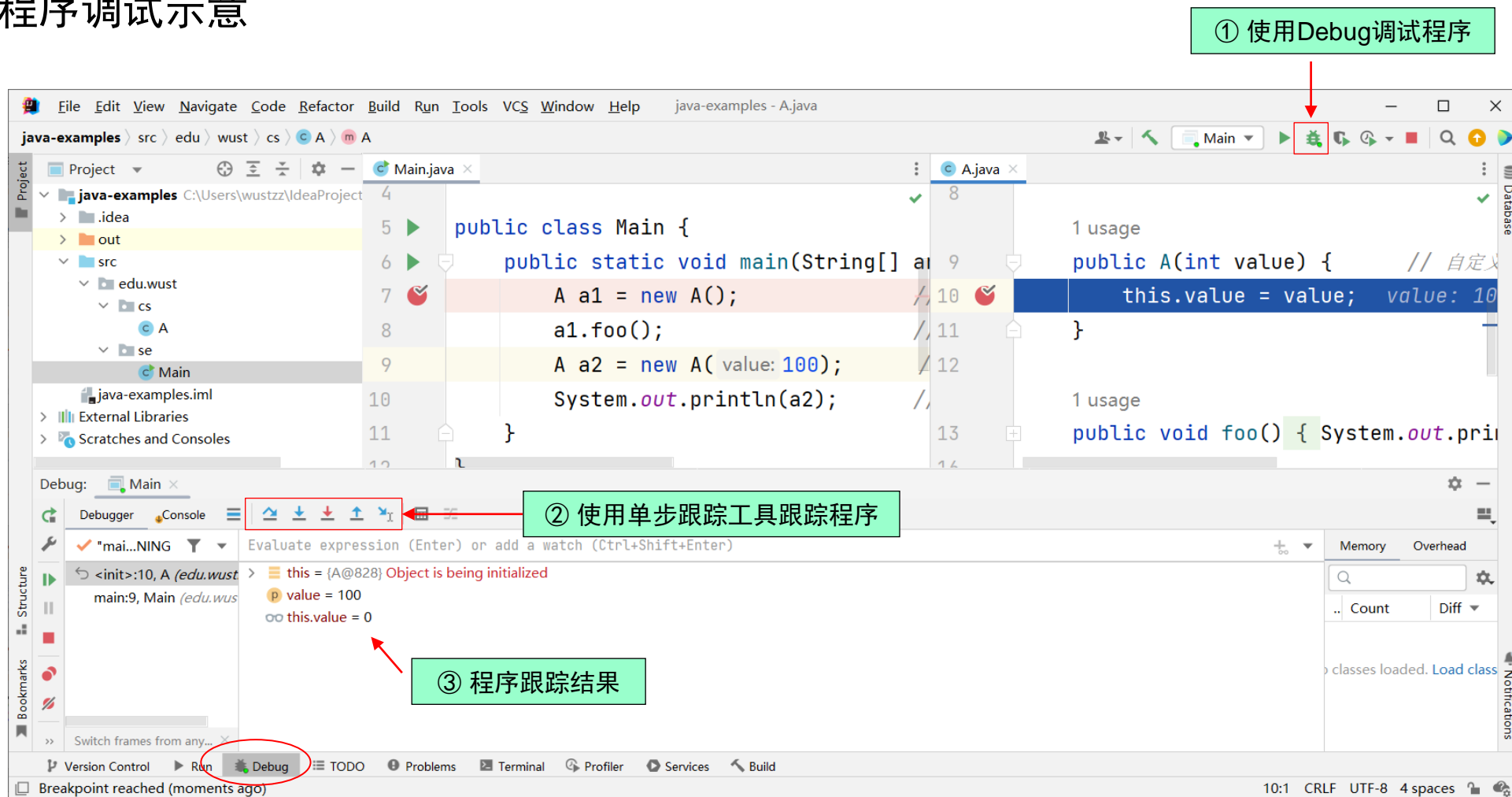
## 最终结果：

[【返回】](#)

### 3. Java程序调试



## 程序调试示意

[【返回】](#)

## 1.5 打包Java程序

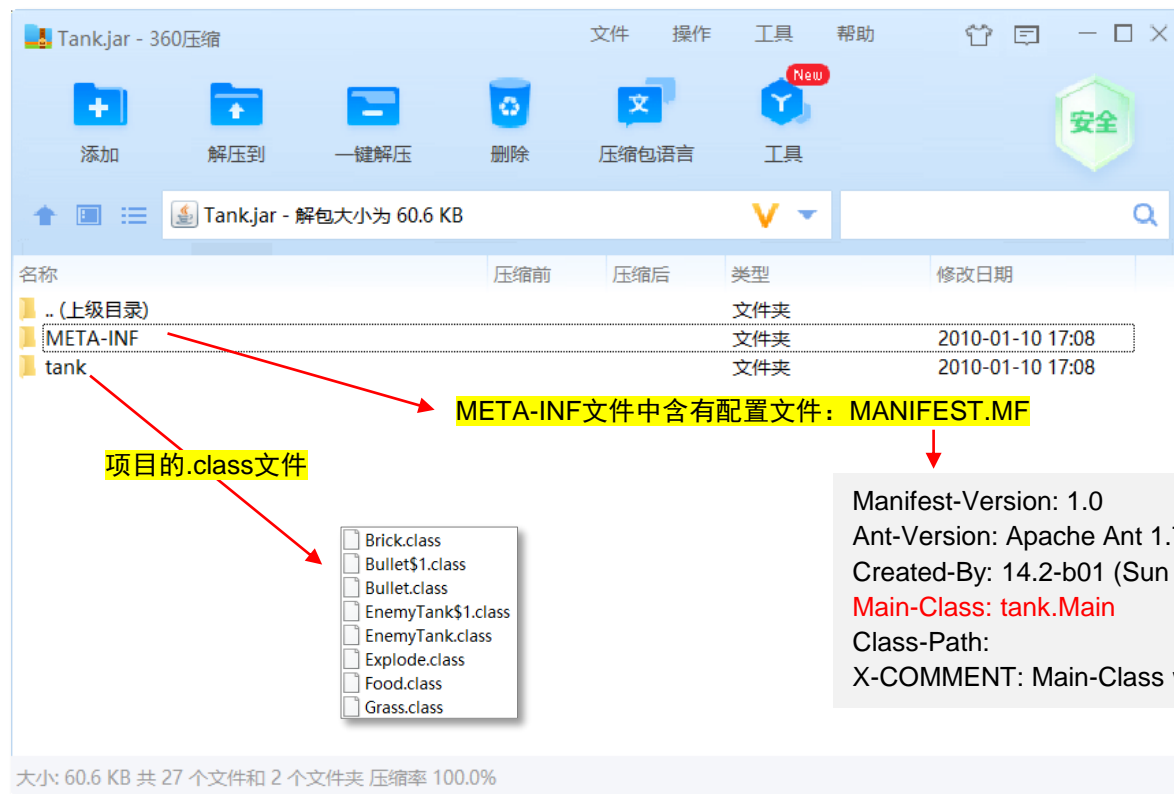
- 在开发过程中，开发人员知道：哪个class是主程序，使用哪个class可以直接运行程序
- 但对普通用户来说：只希望双击启动程序运行得到结果
- 打包：Java可以使用 jar (java archive) 文件来发布项目 ← 注：不是exe文件
  - jar文件是一种按Java格式压缩的类包，是Java文件封装的最小单元
  - 对于含有main()入口函数的jar包，可以双击运行 【[坦克大战jar示例](#)】
  - 也可使用命令行运行，例如：java -jar Tank.jar
  - 通常依赖包都以jar形式发布



依赖包：指第三方提供的程序开发包

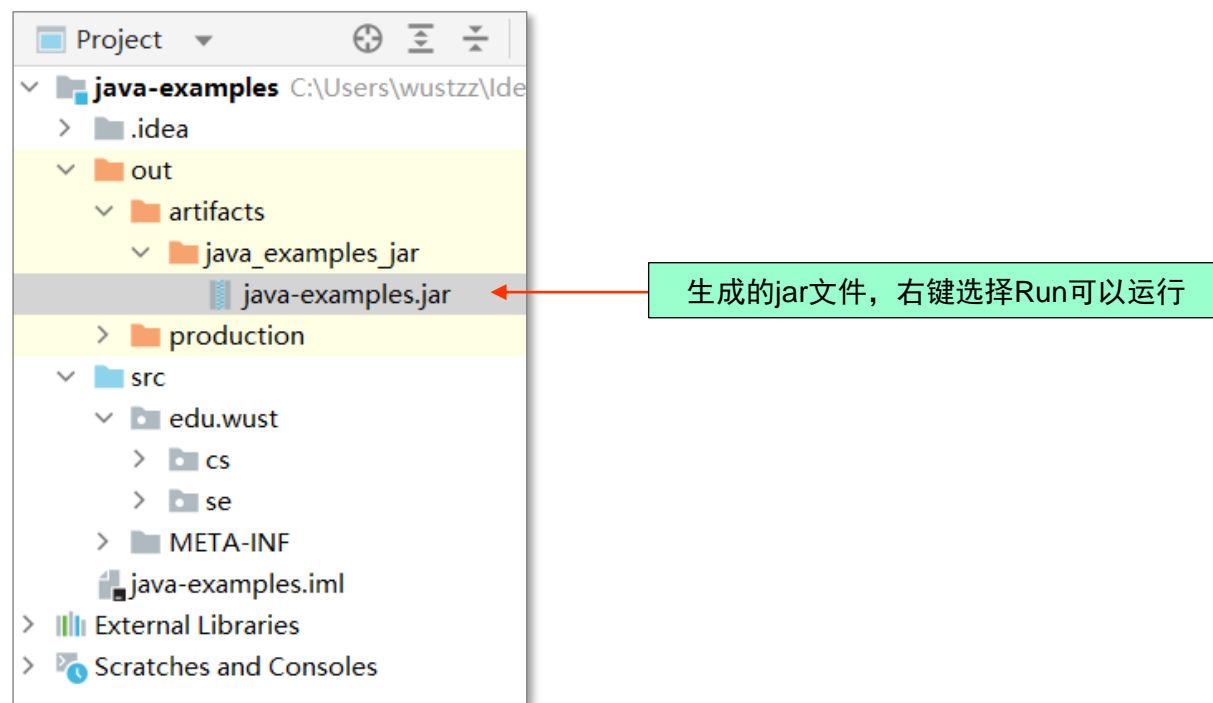


## jar文件示例：使用压缩工具打开jar



## 如何打包

- 打包过程详见：<https://www.cnblogs.com/july7/p/11473751.html>



【[返回](#)】

## 附录1：Idea常用快捷键

- 代码对齐：Ctrl+Alt+L
- Ctrl+/ 或 Ctrl+Shift+/：代码注释
- Ctrl+Z：撤销
- Ctrl+单击：查看源码定义
- 代码提示：Alt+/
- Ctrl+D：复制行
- Ctrl+X：剪切行
- Ctrl+F：查找
- Ctrl+R：替换
- Ctrl+H：显示类结构图（类的继承层次）

## 附录2：多个main问题

- 每一个java类都可有一个`public static void main(String[] args)`作为程序入口（有且只有一个）
- 类中可以有其他main方法，比如：`public static void main()`，`void main(int a)`
- 在运行时会自动寻找main入口函数，而其他的main函数会被当成普通函数来处理

## 多个main示例

```
class Test {  
    public static void main(String[] args) {  
        System.out.println("main3");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        System.out.println("main1");  
        main();  
        new Test().main(null);  
    }  
    public static void main() {  
        System.out.println("main2");  
    }  
}
```

■ 选择Main为主类时运行结果：

main1  
main2  
main3

■ 选择Test为主类时运行结果为：

main3

【完】