

1.1 Java 开发环境配置

- Java 简介
- Java 的安装
- 第一个 Java 程序
- 安装开发环境

目录

1

Java 简介

2

Java 的安装

3

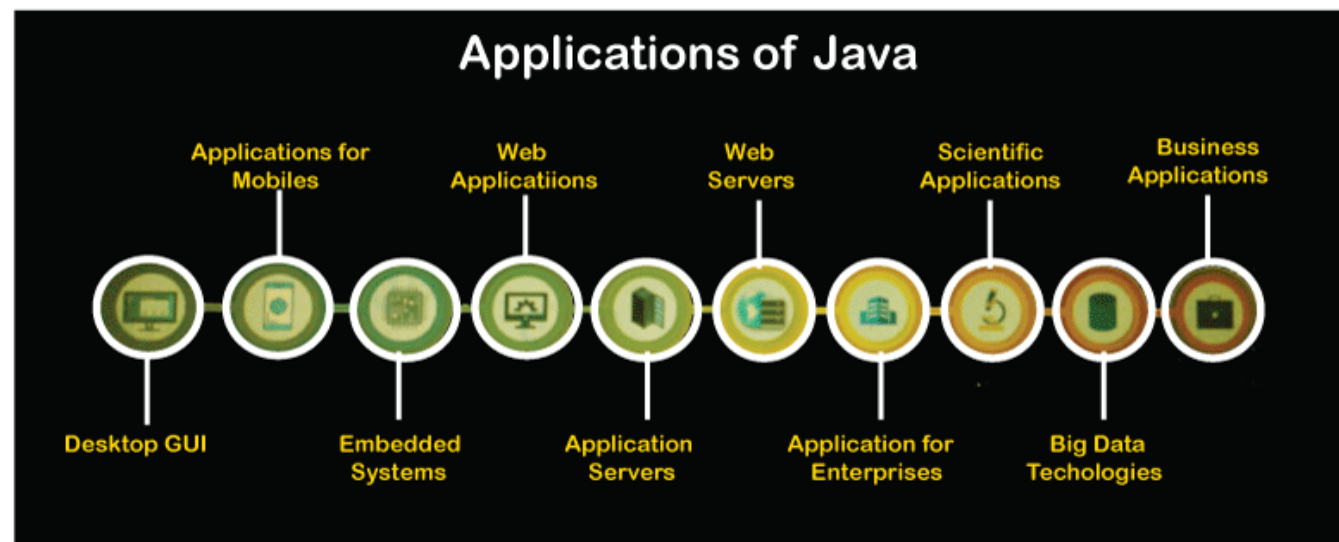
第一个 Java 程序

4

安装开发环境

Java 简介

- **Java** 是一种广泛使用的计算机编程语言，具有跨平台、面向对象[オブジェクト指向]、泛型编程等特性。
- Java 诞生的契机是发明者们觉得 C++ 语言使用起来较为复杂并容易出现问题[バグ]。因此 Java 与 C, C++ 在语法上很相似。
- Java 语言用途广泛：可用于编写服务器[サーバー]，Android 等移动平台应用，企业级业务，互联网应用，桌面应用等。



Java 主要语言特性

- 面向对象[オブジェクト指向]。
- 丰富的 API (Application Programming Interface) :
 - GUI (Graphical User Interface)。
 - 多线程[マルチスレッド]。
 - 网络通信[ネットワーク通信]。
- 自动内存管理:
 - 垃圾回收[ガベージコレクション]。
- 静态类型语言。
 - 编译[コンパイル]时决定变量类型。
- 不依存于平台 (WORA: Write Once Run Anywhere) 。

目录

1 Java 简介

2 Java 的安装

3 第一个 Java 程序

4 安装开发环境

第一步 下载开发工具

- **JDK (Java Development Kit)** 是 Java 开发工具。
- 我们使用 Oracle 官方的 JDK:
 <https://www.oracle.com/java/technologies/downloads/>
找到适合自己电脑的下载包进行下载。
- 本期课程将使用 JDK 17。
- 接下来的步骤会分成 Windows (👉 p7) 和 Mac (👉 p14) 两个版本进行。
- 已安装的同学可以跳过。

第二步 安装 JDK

- 双击下载好的 exe 文件。



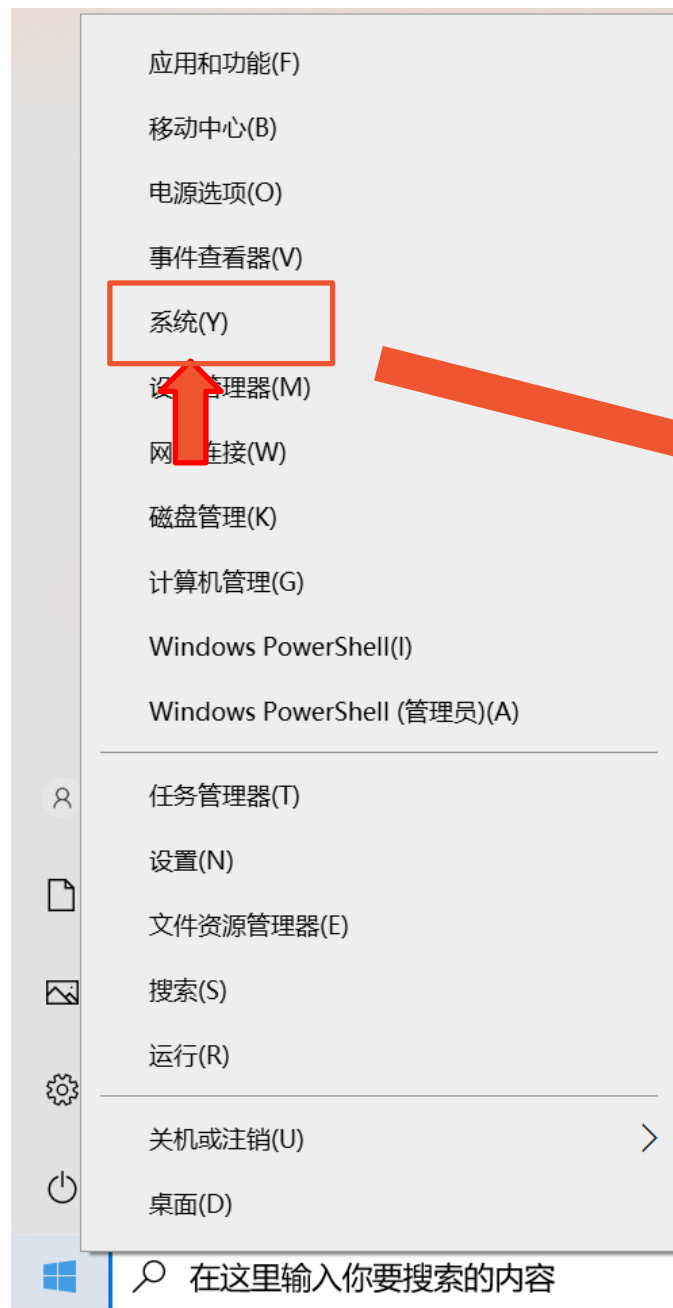
第二步 安装 JDK



记住该路径

第三步 设置环境变量

- 系统 → 高级系统设置 → 环境变量。



相关设置

BitLocker 设置

设备管理器

远程桌面

系统保护

高级系统设置

重命名这台电脑

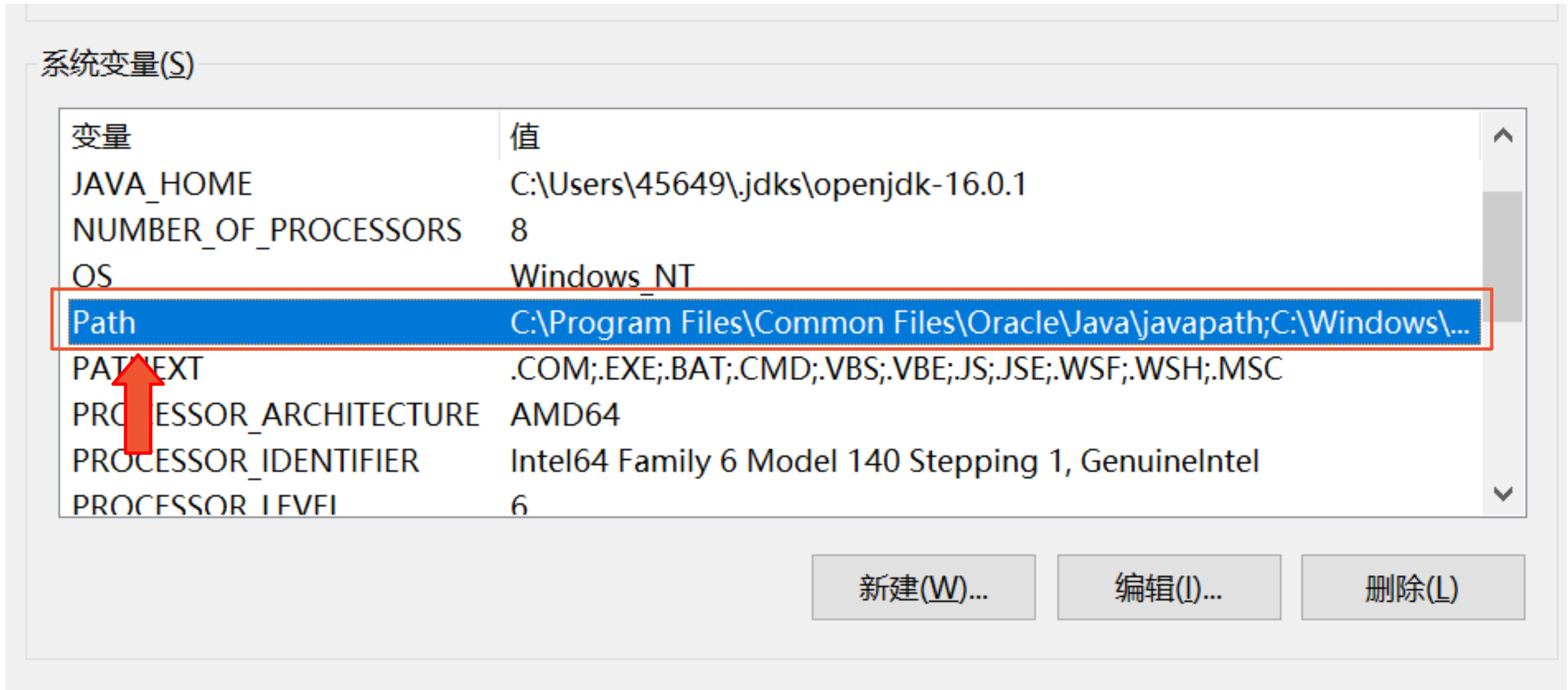
获取帮助

提供反馈

第三步 设置环境变量

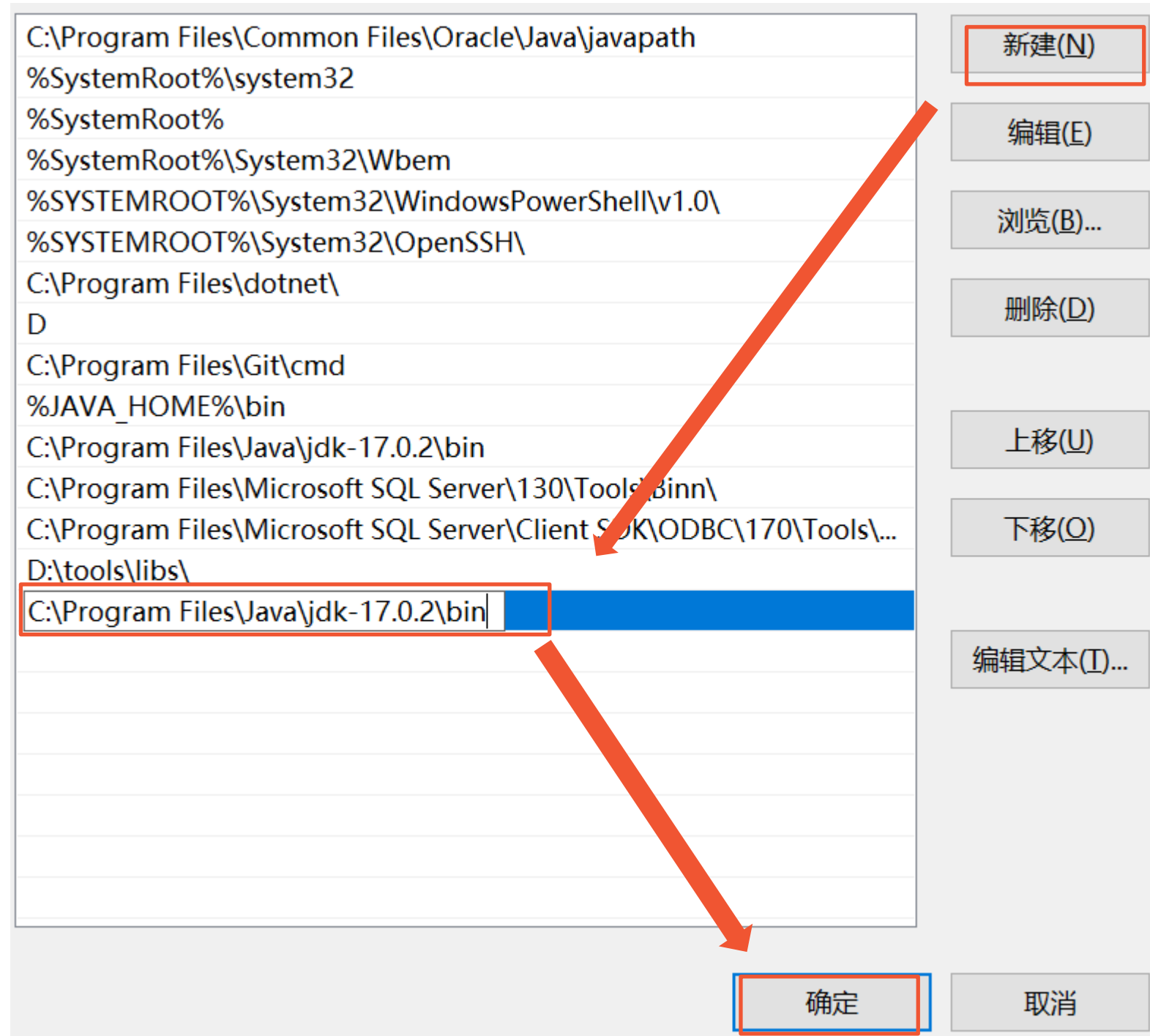


第三步 设置环境变量



第三步 设置环境变量

- 新建:
[Java 安装路径]\bin\



第四步 确认安装成功

- 打开 cmd，输入 `java -version`（注意空格），回车。

```
C:\Users\Your Name>java -version
```

- 如果环境配置成功，则会出现以下结果：

```
java version "11.0.1" 2018-10-16 LTS  
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)  
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)
```

- 这样，环境就配置成功了，可以开始编写 Java 程序了！

第二步 安装 JDK

- 双击下载好的 dmg 文件，双击里面的 pkg 文件。



第三步 确认安装成功

- Mac 会自动配置环境变量。
- 打开 terminal，输入 `java -version`（注意空格），回车。

```
(base) zmnnoMacBook-Pro:~ zmn$ java -version
```

- 出现类似这样的结果，则配置成功，可以开始编写 Java 程序了！

```
(base) zmnnoMacBook-Pro:~ zmn$ java -version
java version "14.0.1" 2020-04-14
Java(TM) SE Runtime Environment (build 14.0.1+7)
Java HotSpot(TM) 64-Bit Server VM (build 14.0.1+7, mixed mode, sharing)
```

Q & A

Question and answer

目录

1 Java 简介

2 Java 的安装

3 第一个 Java 程序

4 安装开发环境

Hello World 编写

- 使用文本编辑器创建一个文件，在里面输入以下代码：

```
1 public class Hello {  
2     public static void main(String[] args) {  
3         System.out.println("Hello world!");  
4     }  
5 }
```

- 注意**缩进**[インデント]的空格。
- 注意使用**英文输入法**。
- 保存为“Hello.java”（如使用 Mac 上的 TextEdit，则**必须**保存为纯文本格式。快捷键：Shift + Command + T）。

Hello World 运行

- 打开 cmd 或者 terminal, cd 到 Hello.java 所在的目录下。
- 输入 `javac Hello.java`, 回车。
- 输入 `java Hello`, 回车。

```
1.1 Java Environment Setup¥src>javac Hello.java  
1.1 Java Environment Setup¥src>java Hello
```

- 如果编写无误, 则屏幕上应显示 “Hello world!”。
- 恭喜你编写出了 Java 语言的 “Hello world” 程序!

Hello World 说明：类

- 每一份 Java 代码都属于某个“**类** (Class)” [\[クラス\]](#)。在示例中，我们定义了一个名为 Hello 的类。

```
public class Hello {
```

- Java 代码文件的名称必须与其包含的类名称相同。保存文件时，请使用类名进行保存，并在文件名的末尾添加“.java”。
- 类是面向对象编程的基本元素。我们会在之后的课程（[➡ § 2.1.1](#)）中对类作详细介绍。

Note



代码文件名须与类名一致。

Hello World 说明: main 方法

- “**main 方法**[メソッド]”是一切 Java 代码的起点。main 方法中的代码会在 Java 启动时被依序执行。
- main 方法的框架长得都一样:

```
public static void main(String[] args) {  
    希望被执行的代码  
}
```

- 我们会在之后的课程 (➡ § 1.3.3) 中详细讨论方法。现在, 只需复制粘贴 main 方法的框架至每一个程序即可。

Hello World 说明: `System.out.println`

- `System.out.println` 方法会在屏幕上打印（输出）一行文本：

```
System.out.println( "希望被输出的文本" );
```

- 文本须用双引号括起。
- 我们可以用这个方法输出我们想看见的结果，或是进行代码测试。

Hello World 说明：Java 的基本语法规则

- Java 语句由上至下依次执行。
- 每一个 Java 语句都由分号 “;” 结尾。换行并不能用于分隔语句！
- 每一个单词可以由任意个空白符号：空格 “ ”、tab “ ” 或换行符（回车）分隔。
- 花括号 “{}” 括起的语句属于一个代码块[ブロック]，如：同一个类 / 方法中的语句。“}” 后无需另加分号以分隔。

Java 注释

- 注释[コメント]是不会被当作代码编译或执行的文本。
- “//” 进行单行注释:

```
1 System.out.println("A"); // This will print "A".
```

- “/* */” 进行多行注释:

```
1 /* The next code will print "A" onto the screen.  
2    You can change "A" to any sentence you want. */  
3  
4 System.out.println("A");
```

练习时间

- 补全以下代码：

```
1 public class Hello {  
2     public static void main(String[] args) {  
3         // Write your code here.  
4     }  
5 }
```

- 以输出以下内容：

```
hello  
world
```

Tips

尝试 System.out.print 方法，看看它与 System.out.println 的区别。

Coffee ☕ Break

代码风格与可读性

一段同样功能的代码有很多不同的书写方式。在大部分现代编程语言中，诸如空格，缩进，换行等空白符号都可由程序员自由安排。因此，每一个开发人员都可以有不同的**代码风格**[プログラミング作法]和习惯。

另一方面，在合作开发的过程中，一个统一的书写风格是提高代码**可读性**[可読性]、提升交流效率的关键。我们鼓励大家学习借鉴主流的、权威的书写风格。

本教材使用的 Java 代码参考 Google 的开发风格：
 <https://google.github.io/styleguide/javaguide.html>。

Q & A

Question and answer


目录

- 1 Java 简介
- 2 Java 的安装
- 3 第一个 Java 程序
- 4 安装开发环境

集成开发环境

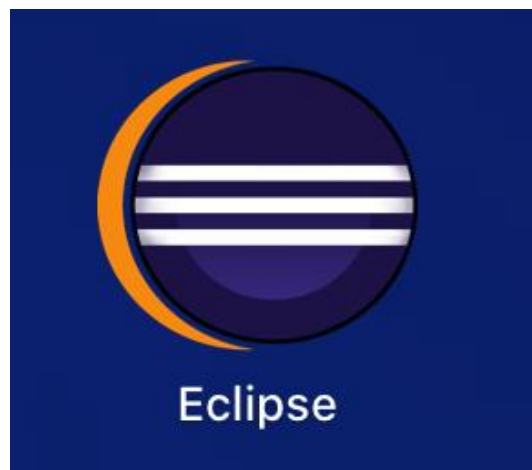
- **集成开发环境**[統合開発環境] (**I**ntegrated **D**evelopment **E**nvironment, **I**DE) 是辅助我们进行程序开发的软件。比起直接在文本编辑器中开发, IDE 提供了许多方便的功能:
 - 一键运行;
 - 代码错误自动检测;
 - 调试 (Debugging) ;
 - 代码自动生成;
 - 多种语言开发;
 -
- 常见的支持 Java 开发的 IDE 包括 **Eclipse**、VS Code、IntelliJ IDEA 等。

下载 Eclipse

- 下载链接： <https://www.eclipse.org/downloads/>。
- 选择适合自己系统的进行下载。
- 已安装过的同学不必重复下载。

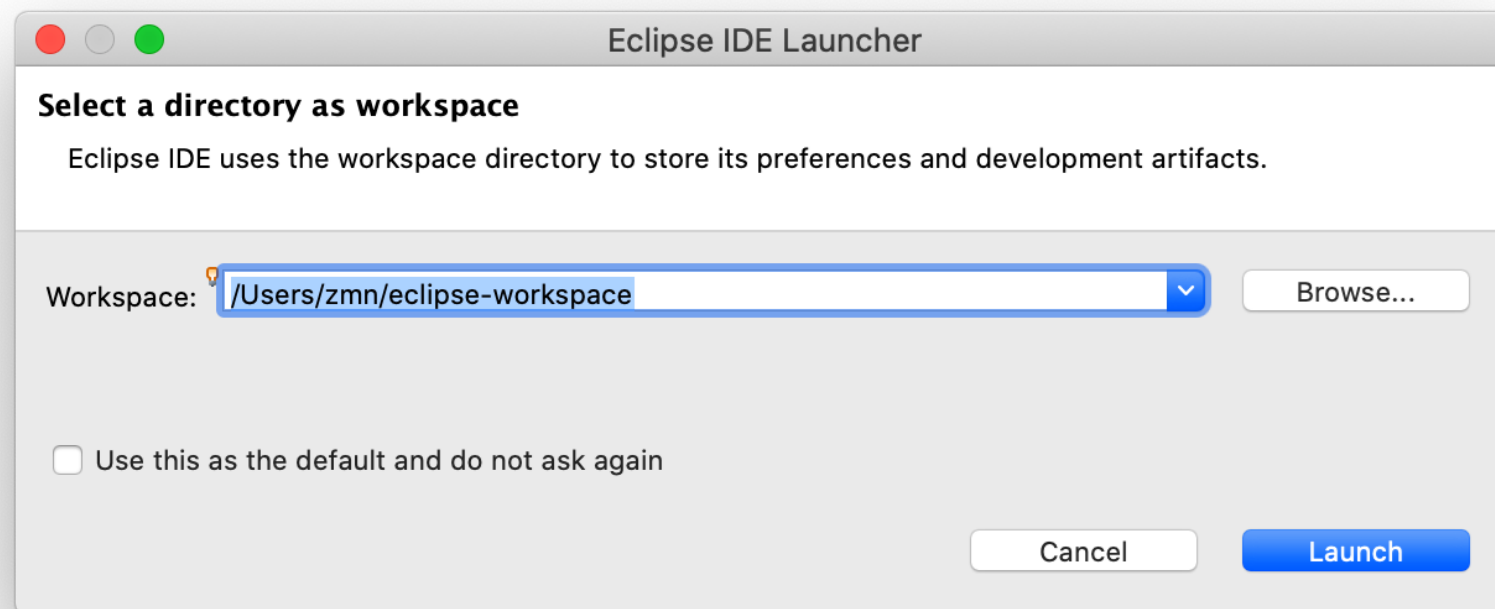
Eclipse 的安装

- 打开下载好的安装包 exe / dmg 进行安装。
- 安装好是这样的：

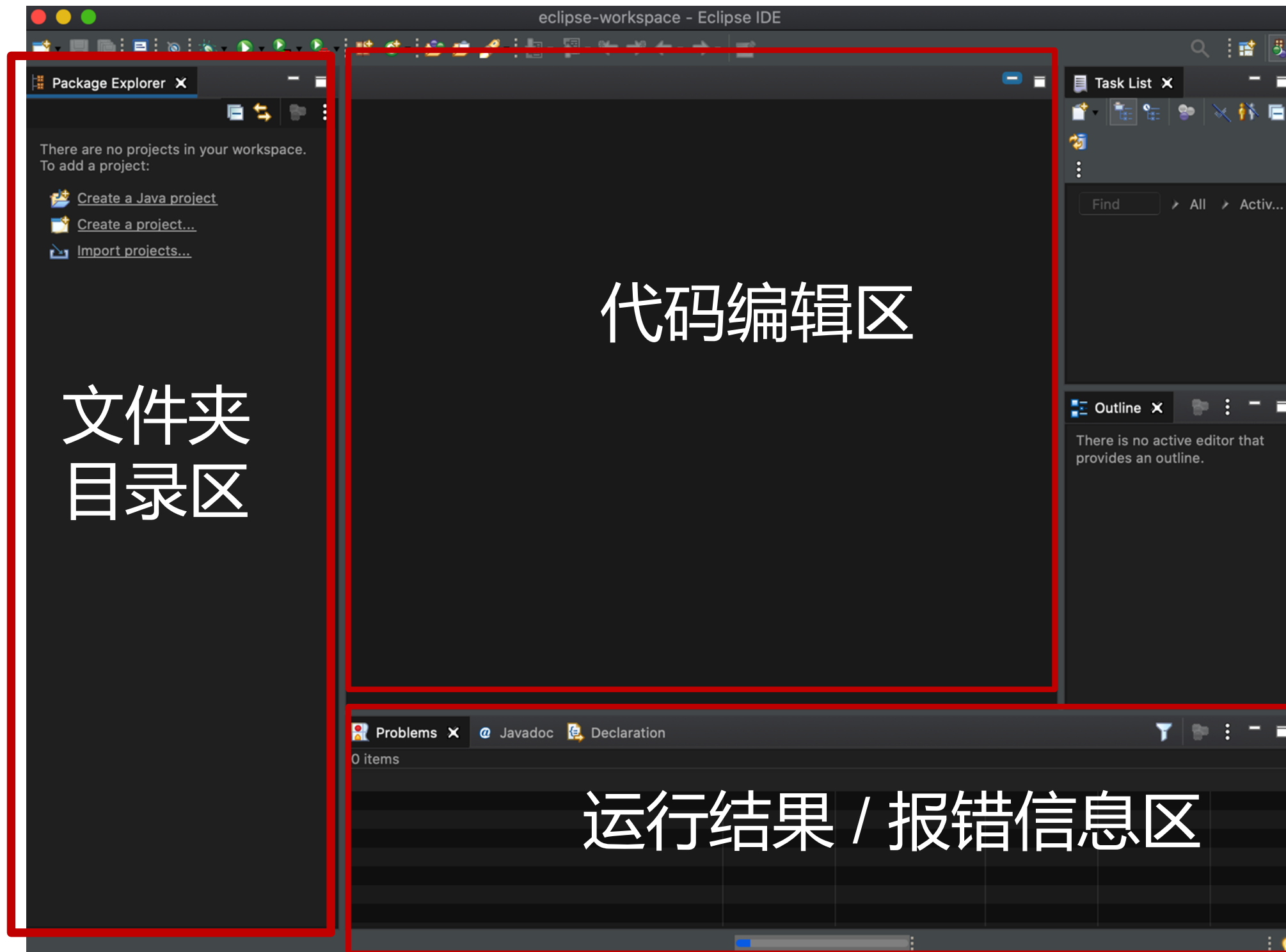


打开 Eclipse

- 双击打开 Eclipse，每次打开都会弹出一个让你选择 workspace 的窗口，用默认的地址就可以，点击 Launch。

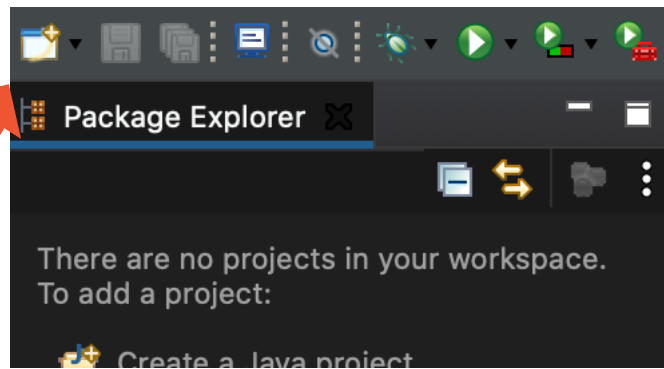


Eclipse 界面简介

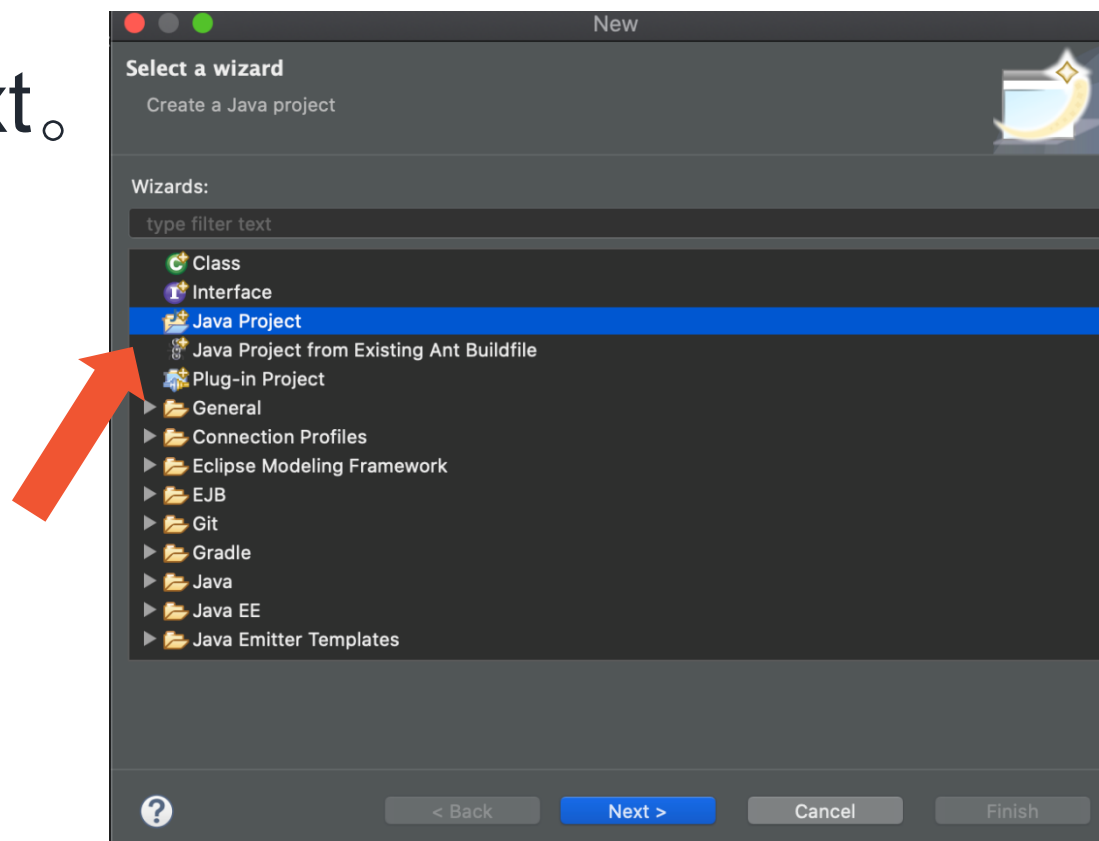


新建 Java 项目 1 / 2

- 点击左上角的新建图标。

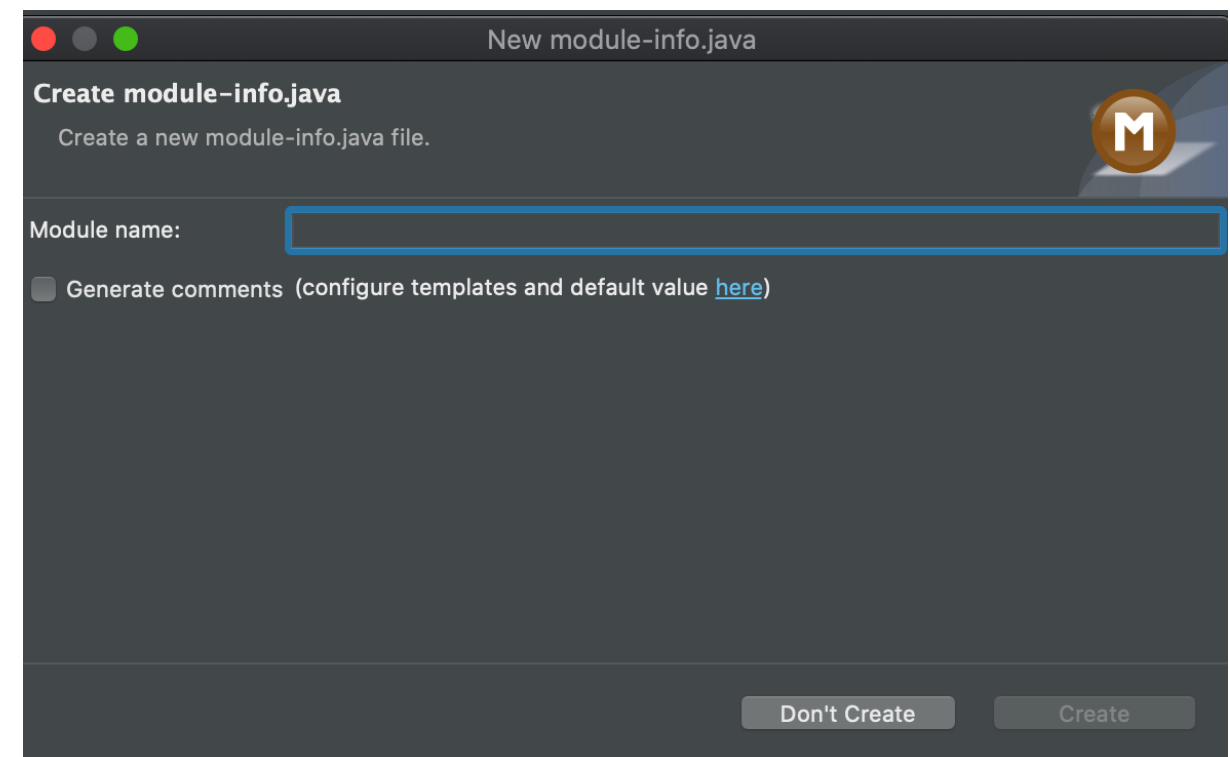
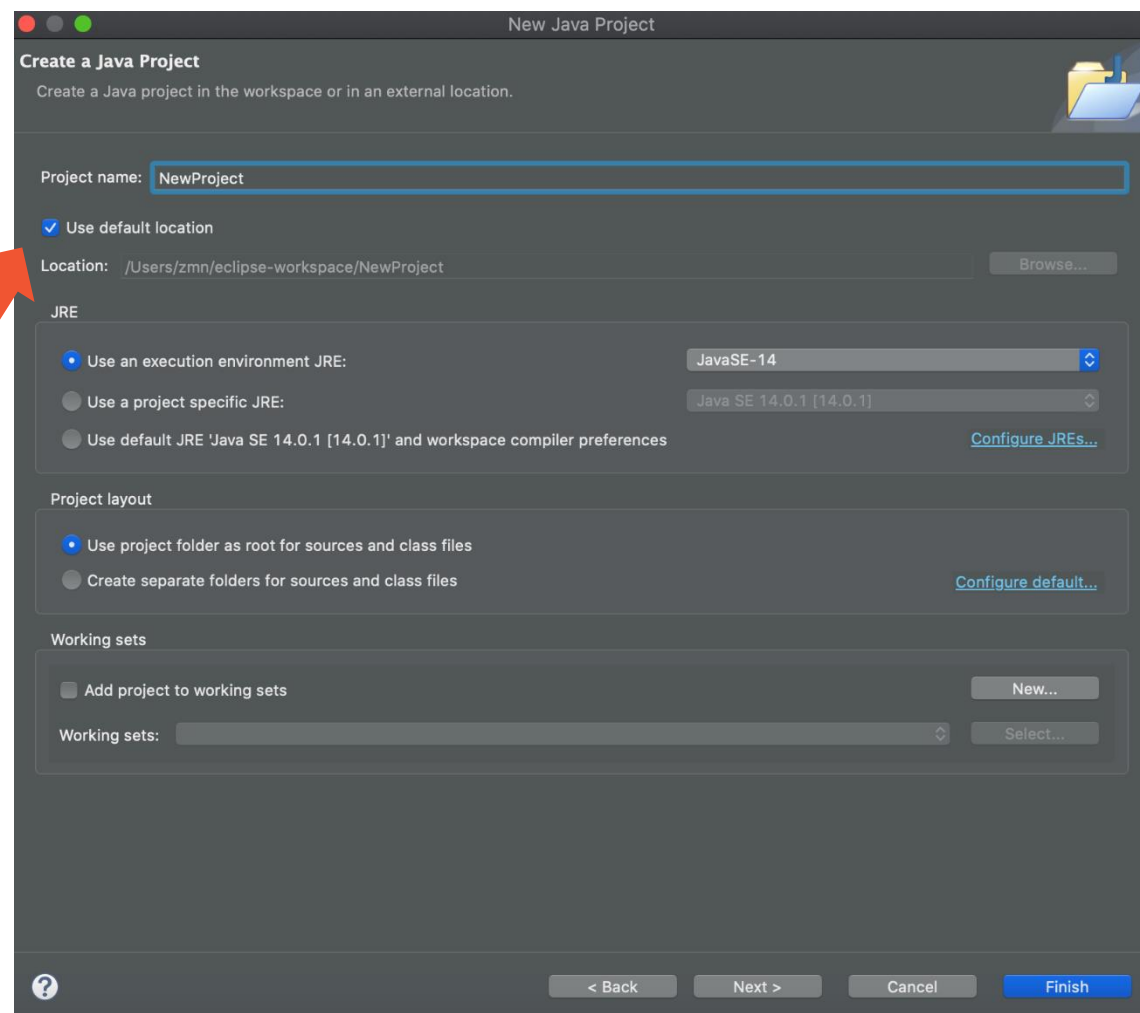


- 选择 Java Project → Next.



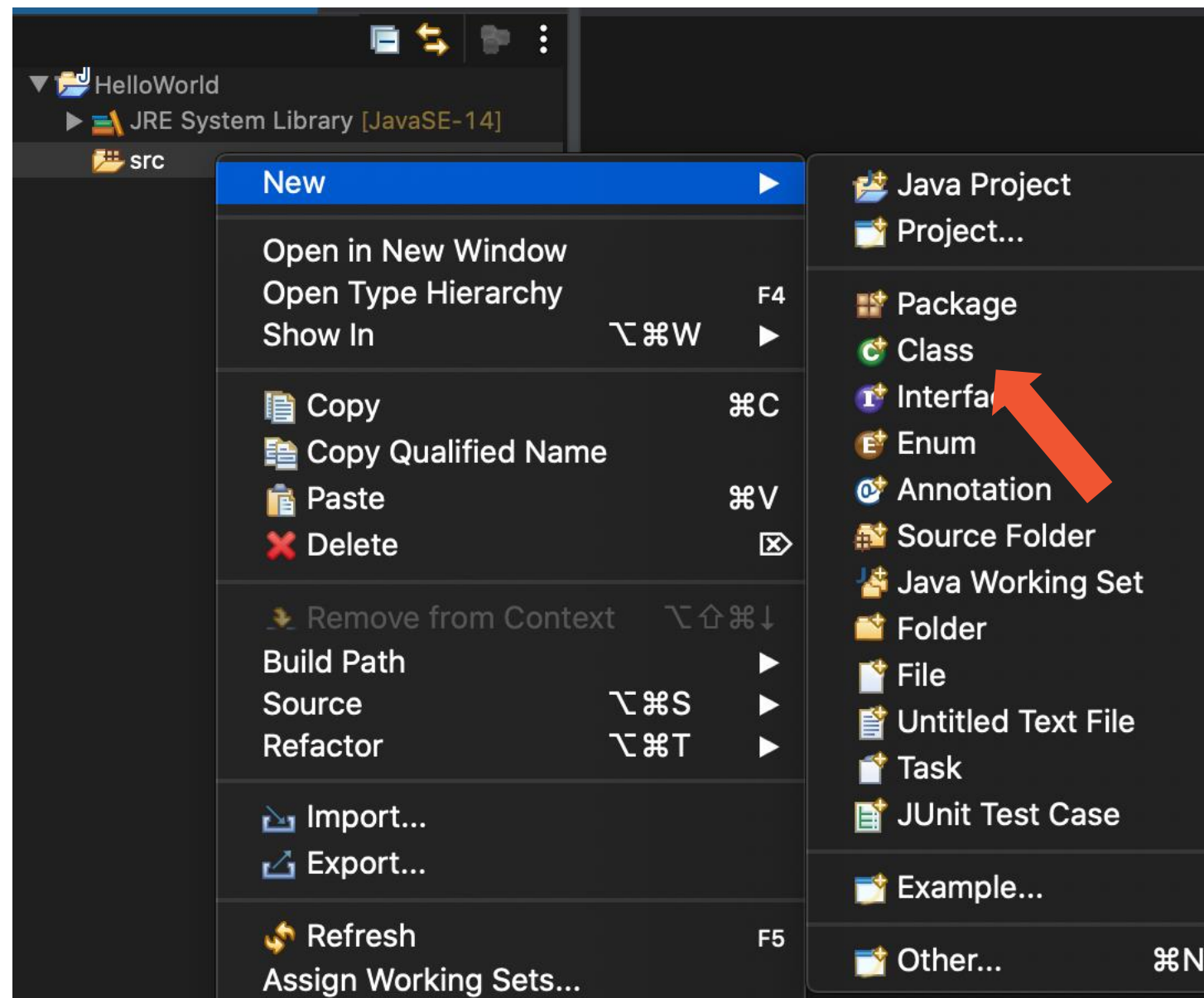
新建 Java 项目 2 / 2

- 填写 Project name。
- 其他的默认即可，点击 Finish。
- 会弹出一个询问你是否创建 module 的窗口，这里选择 Don't Create。



新建 Java 类 1 / 2

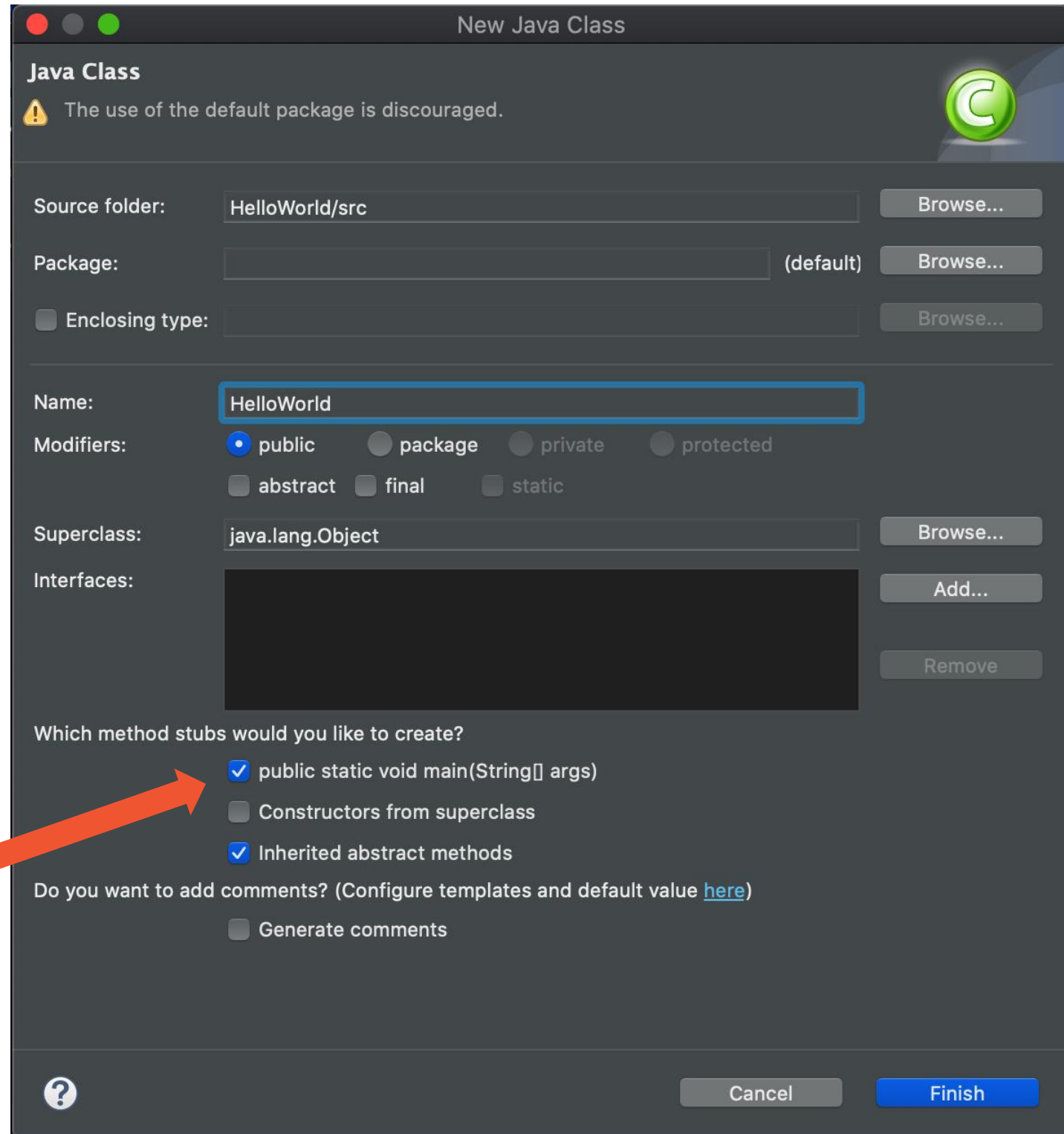
- 创建好以后，右键其中的 src 文件夹，New → Class。



新建 Java 类 2 / 2


Tips


打上这个钩，可以自动生成 main 方法哦！




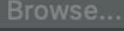
New Java Class

Java Class

 The use of the default package is discouraged.


Source folder: HelloWorld/src 

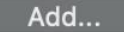

Package: (default) 

☐ Enclosing type: 

Name: HelloWorld

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static




Superclass: java.lang.Object 

Interfaces:  

Which method stubs would you like to create?

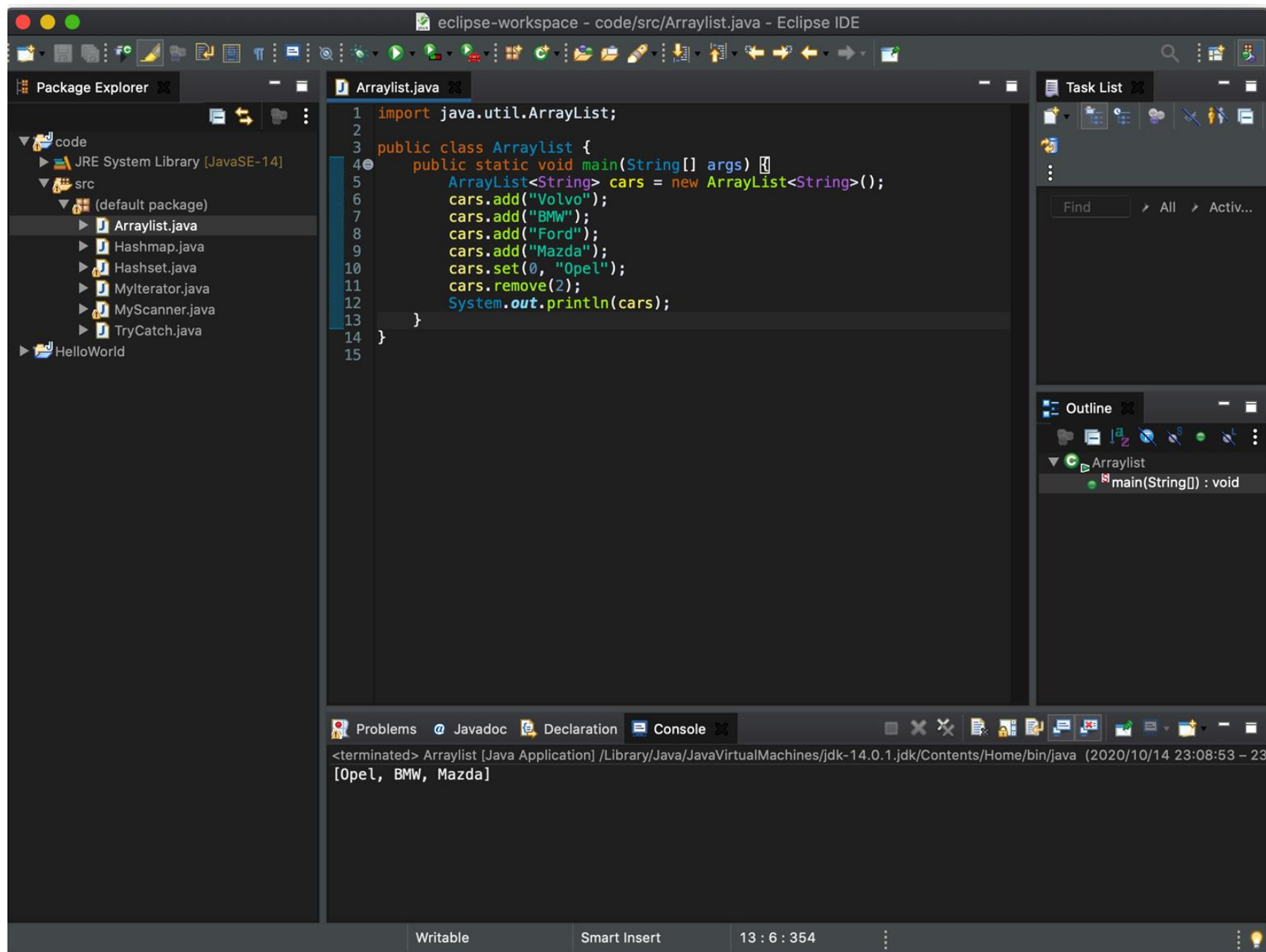
☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

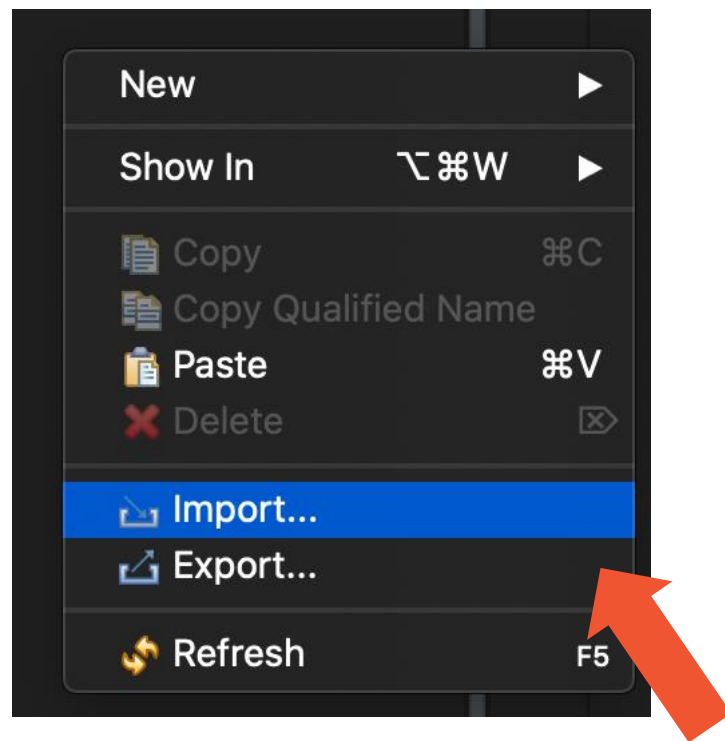
运行代码

- 点击上方绿色标志 ，即可运行当前代码。



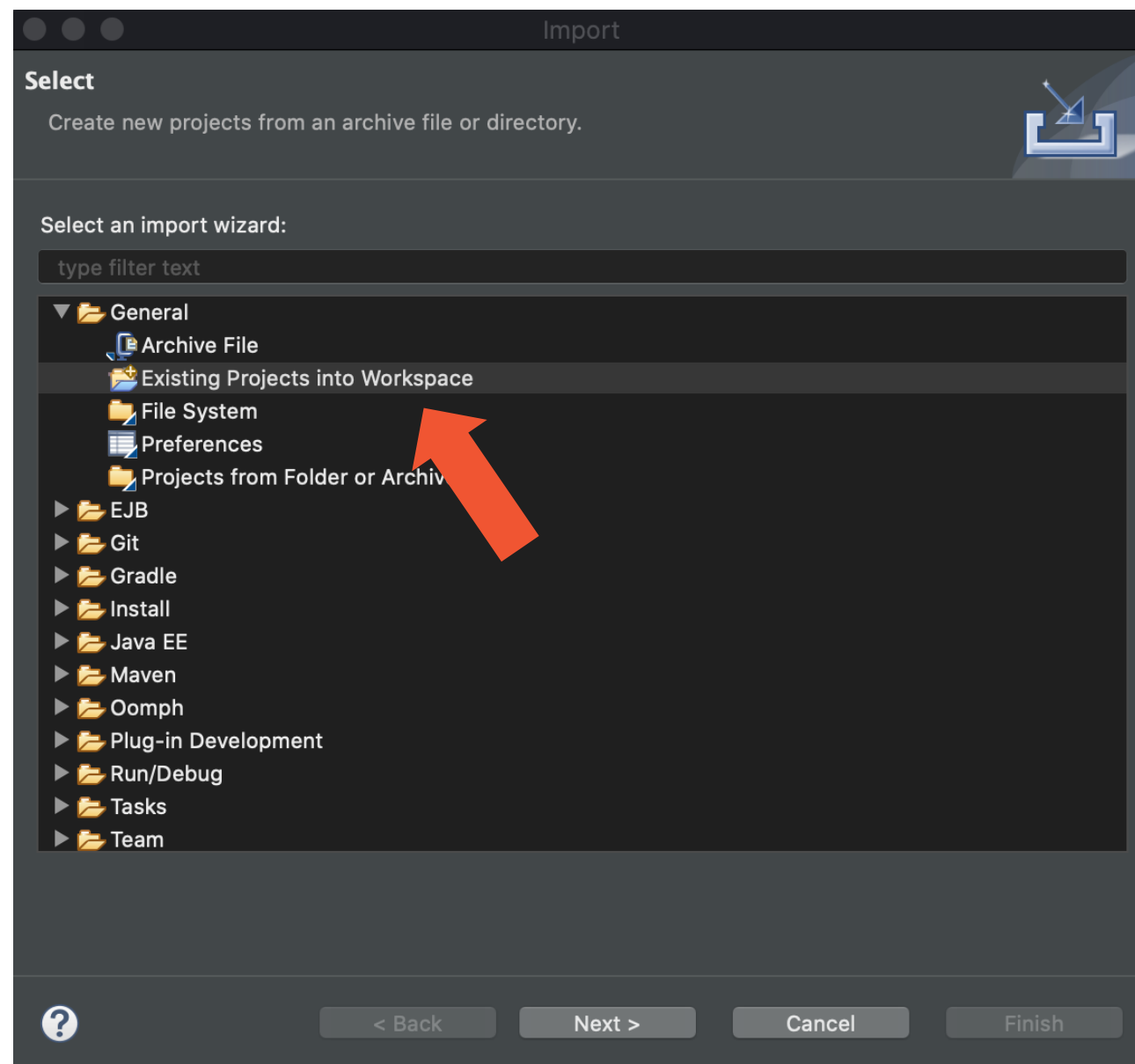
导入外部项目 1 / 3

- 接下来有些代码会作为项目发给大家，所以编写之前需要先导入。
- 右键文件夹区域，选择 Import。



导入外部项目 2 / 3

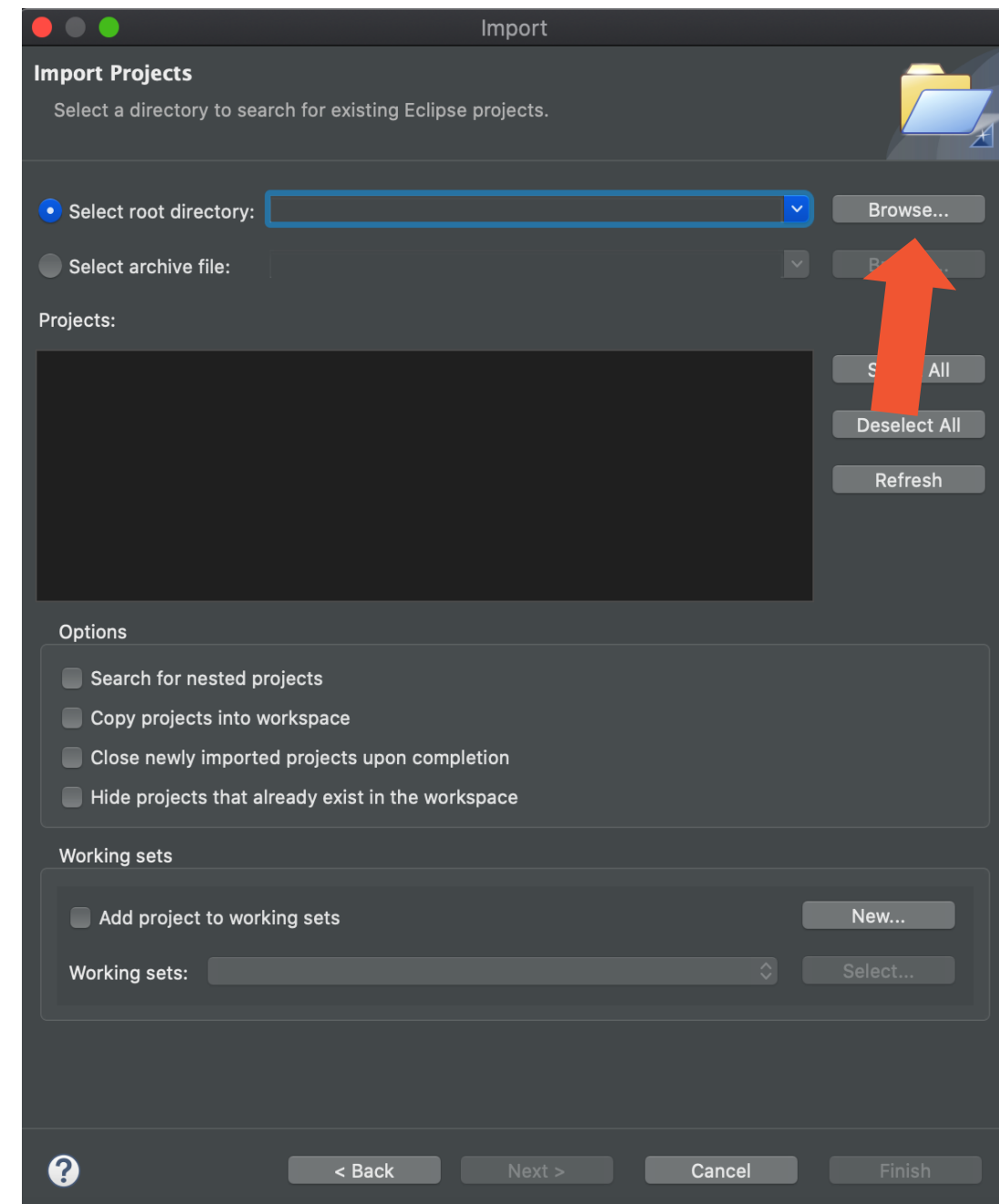
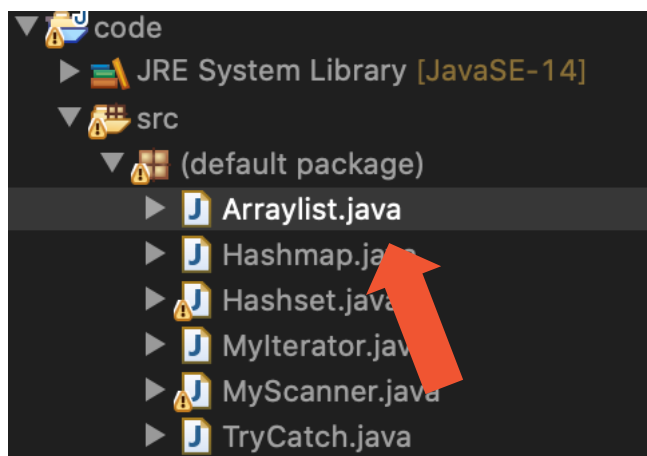
- 选择General → Existing Projects into Workspace → Next。



导入外部项目 3 / 3

- 点击 **Browse**，找到本节课课件代码文件夹，点击 **open**，**Finish**。

- 导入后，双击 **Java** 文件查看代码。



Q & A

Question and answer

THANK YOU!