



南京邮电大学  
Nanjing University of Posts and Telecommunications

# Java软件开发(混合式)

南京邮电大学 计算机学院  
Java课程组



# 第1章 Java语言概述



- 1.1 初识Java
- 1.2 安装和配置
- 1.3 编写Java程序
- 1.4 集成开发环节IntelliJ IDEA
- 1.5 项目概述

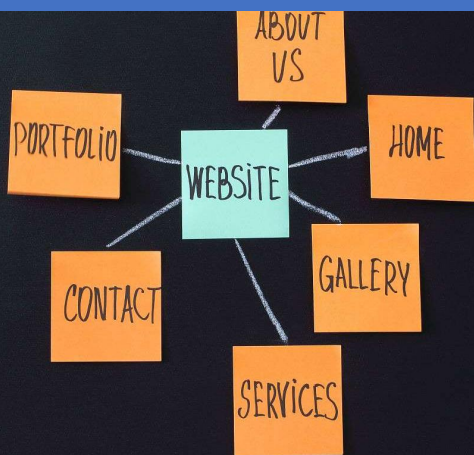


# 引言

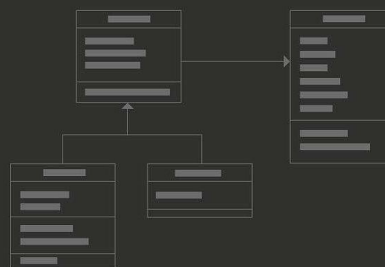


## Java的广泛使用

Java 语言是当今最受欢迎的编程语言之一，它被广泛应用于各种应用程序和平台。



## Class Diagram



## 介绍Java的内容

我们将探讨 Java 的基本概念、特点和优势，以及它是如何成为一种广泛使用的编程语言的。

## Java的跨平台特性

Java 语言具有跨平台的特性，能够使得程序在各种操作系统和硬件平台上运行。





# 1.1.1 Java的产生



- 1991年Sun Microsystems公司为发展消费类电子产品，成立了Green小组，该小组开发了一个名为 \*7的项目。这个项目的负责人是James Gosling，考虑到跨平台和安全性等问题，他们开发了基于C++语言的一种新的语言——Oak。该语言采用了很多C语言的语法，提高了安全性，并且是面向对象的语言。
- 该项目未获得商业成功。
- Star7演示视频：  
<https://www.bilibili.com/video/av54888072>



James  
Gosling



# 1.1.1 Java的产生



- 1994年，Sun公司发现他们的新型编程语言“Oak”比较适合编写Internet程序，于是他们对“Oak”进行改进和完善，并获得了巨大的成功。
- 1995年1月，“Oak”更名为Java。这个名字的产生，来自于印度尼西亚一个盛产咖啡的岛屿，中文名为爪哇，寓意是为世人送上一杯热气腾腾的咖啡。



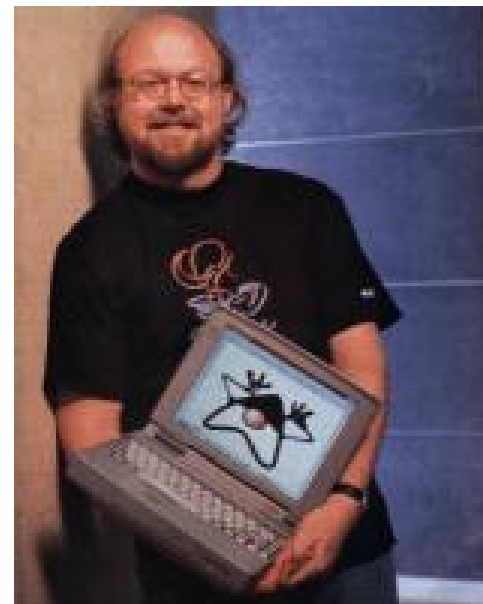


# 1.1.1 Java的产生



## James Gosling的个人荣誉：

- 2002年：被经济学人杂志授予发明奖；
- 2007年：被授予二级加拿大勋章；
- 2013年：当选为ACM Fellow；
- 2015年：被授予IEEE约翰·冯·诺依曼奖章；
- 2018年：因为设计并创造Java语言被收录到计算机历史博物馆荣誉墙。



James Gosling

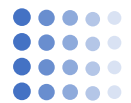


# 1.1.1 Java的产生



- **Java SE——Java标准版或Java标准平台（本课程对应内容）**
  - 提供了标准的JDK开发平台，利用该平台可以开发Java桌面应用程序和低端的服务器应用程序，也可以开发Java Applet。
- **Java EE——Java企业版或Java企业平台**
  - 可以构建企业级的服务应用，Java EE平台包含了Java SE平台，并增加了附加类库，以便支持目录管理、交易管理和企业级信息处理等功能。
- **Java ME——Java微型版或Java小型平台**
  - 是一种很小的Java运行环境，用于嵌入式的消费产品中。





# 1.1.2 Java的特点



## 简单易用

Java 语言的设计使得开发者能够更加简单、更加高效地使用，同时降低了开发成本。

## 面向对象

Java 语言是一种面向对象的语言，提供简单的类机制以及动态的接口模型。它注重开发者的抽象思维和设计能力，同时提供了丰富的类库和API。

## 跨平台特性

Java 语言具有跨平台的特性，能够使得程序在各种操作系统和硬件平台上运行，而不需要重新编写代码。

## 动态特性

Java 语言具有动态特性，允许程序在运行时修改自身状态和行为，从而更加灵活和高效。



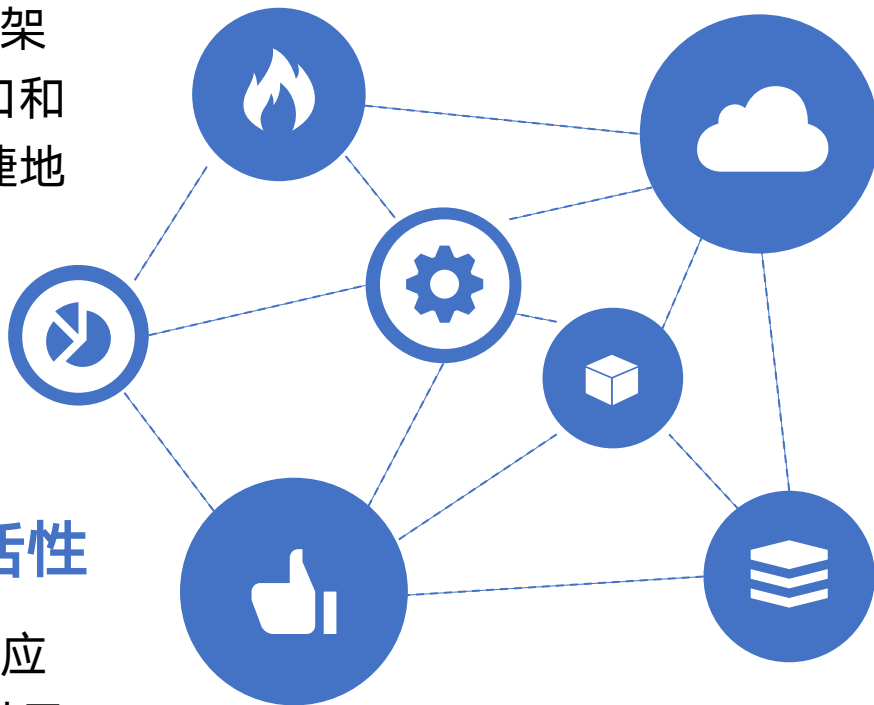


# 1.1.2 Java的特点



## 高度集成

Java 语言能够与各种平台和框架高度集成，同时提供统一的接口和规范，使得开发者能够更加便捷地开发应用程序。



## 灵活性

Java 语言具有灵活性，能够适应各种应用场景和需求，同时提供足够的灵活性和扩展性。

## 可维护性

Java 语言具有可维护性，能够提高代码的可读性和可维护性，同时降低开发成本和时间成本。

- Java语言提供的安全
- 编译器提供的安全
- 字节码校验
- 类装载

## 安全性

Java 语言具有安全性，能够提供强大的安全保障机制，同时保障用户数据和隐私的安全。



## 1.1.2 Java的特点

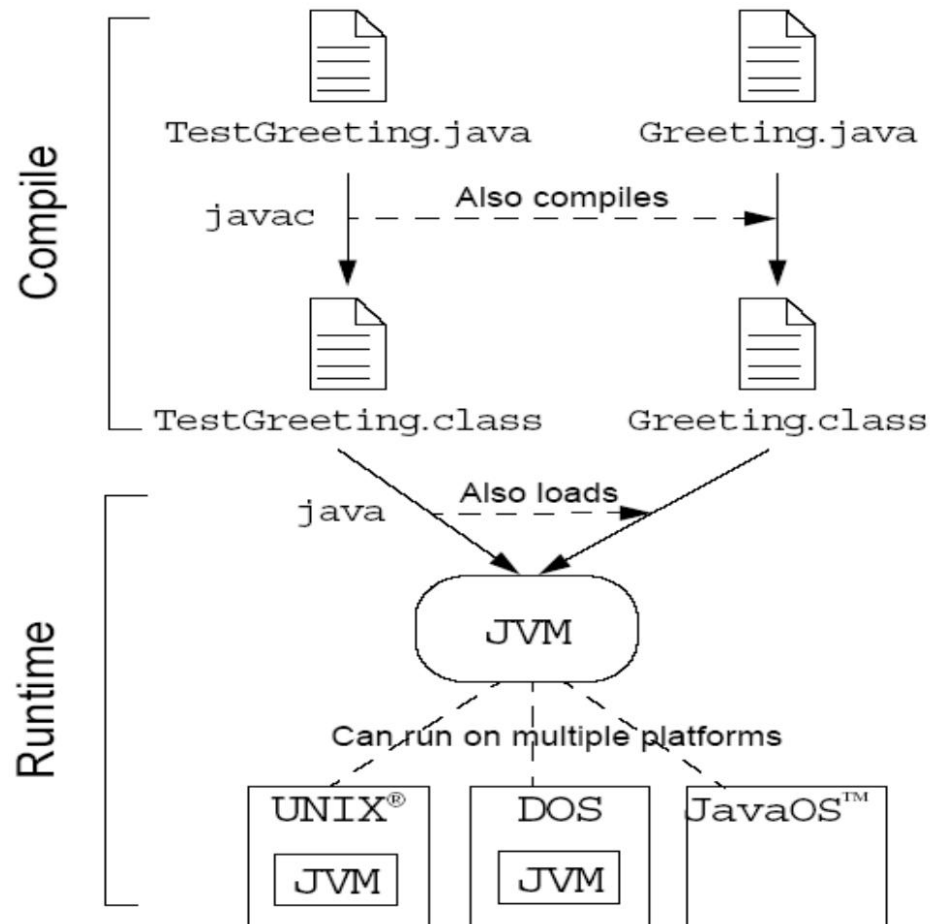


- 平台无关性：**Java语言和其他语言相比，最大的优势就是能够在所有的计算机上运行，这也是Java风靡全球的主要原因。

源代码：.java  
字节码：.class

先编译  
后解释执行

跨平台  
一处编译，多处运行





## 1.2.1 下载和安装JDK

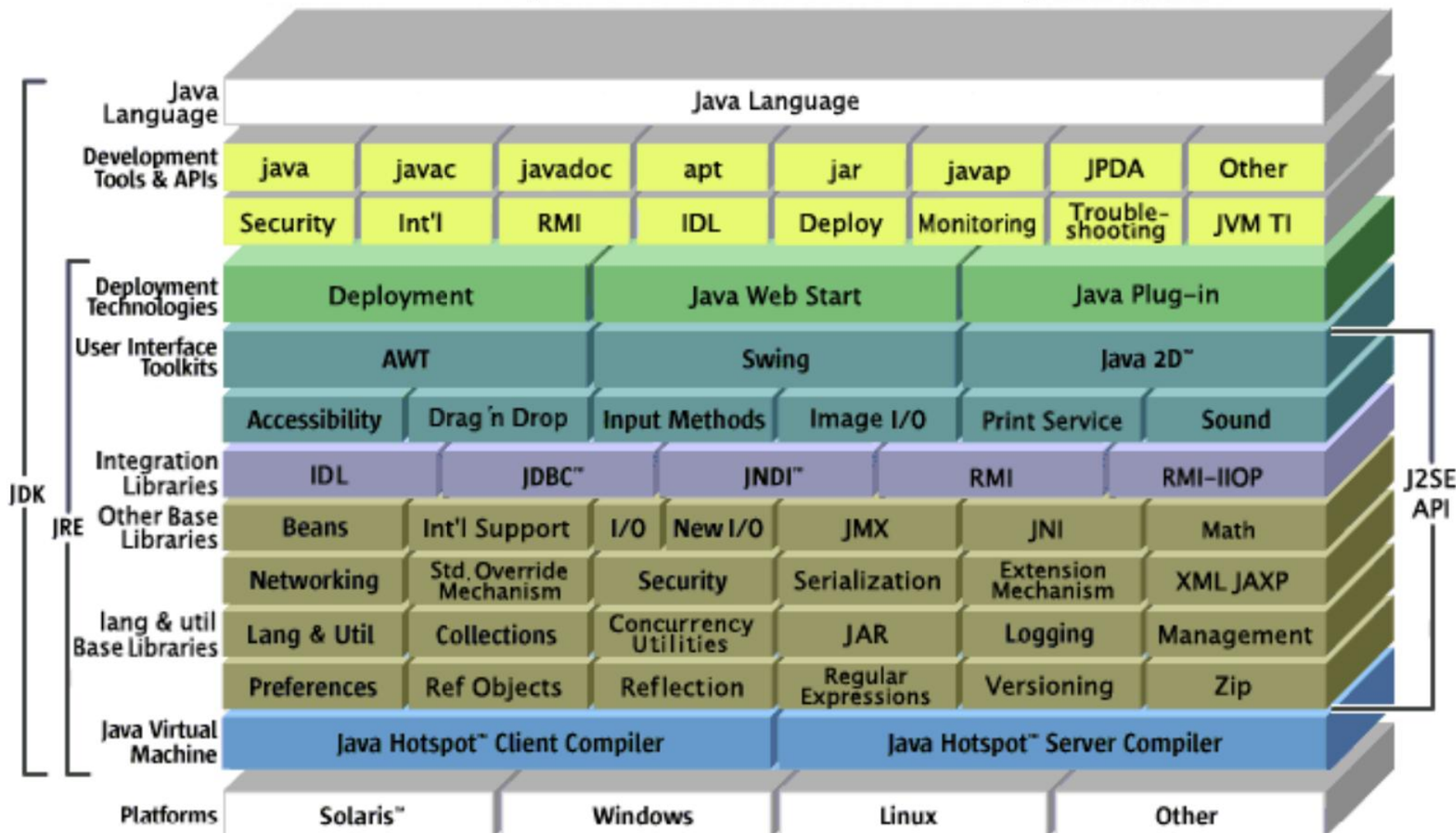
- **JDK: Java Developers Kits**，即Java开发工具包。JDK是整个Java开发的核心，是程序员使用Java语言编写Java程序所需的开发工具包，是提供给程序员使用的。包括：
  - JRE（Java的运行时环境）
  - Java工具（如：编译器javac，文档生成器javadoc等）
- **JRE: Java Runtime Environment**，即Java的运行时环境，是使用Java语言编写的程序运行所需要的软件环境，是提供给想运行Java程序的用户使用的。包括：
  - JVM（Java虚拟机）
  - Java SE标准类库
- **JVM: Java Virtual Machine**，即Java虚拟机，负责将**字节码**转换为**机器码**并执行，是跨平台能力的核心实现。



# 1.2.1 下载和安装JDK



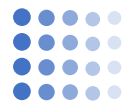
Java™ Platform, Standard Edition (Java SE)



JDK = JRE + Tools

JRE = JVM + API

需要下载JDK



# 1.2.1 下载和安装JDK



- ◆ 1996年 JDK 1.0。
- ◆ 1997年2月 JDK1.1。
- ◆ 1998年12月，JDK1.2(J2SE)。
- ◆ 2004年9月，JDK1.5→ Java SE 5.0
- ◆ 2009年4月，Oracle公司通过收购Sun公司获得Java版权。
- ◆ 2011年7月28日，甲骨文公司发布Java SE 7.0的正式版。
- ◆ 2014年3月18日，甲骨文公司发布Java 8.0的正式版。(长期)
- ◆ 2018年9月26日，甲骨文公司发布Java 11的正式版。(长期)
- ◆ 2019年9月17日，甲骨文公司发布Java 13的正式版。
- ◆ **2021年9月14日，甲骨文公司发布Java 17的正式版。(长期)**
- ◆ 2023年3月22日，甲骨文公司发布Java 20的正式版。
- ◆ **2023年9月19日，甲骨文公司发布Java 21的正式版。(长期)**
- ◆ 2025年3月18日，甲骨文公司发布Java 24的正式版。

- Java8 在 2019 年 1月之后停止更新。
- 从2017年9月发布Java 9开始，Oracle每六个月就会发布一个新版本的JDK，具体来说每年的3月和9月。
- 每三年会有一个“长期支持的版本”（Long Term Support release，简称LTS），该版本会提供为期三年的支持。
- 其他版本仅支持六个月，这就是说，Java 10会替代Java 9，而Java 11发布时又会替代Java 10。
- 目前最新的长期支持版本是2023年9月发布的Java 21，该版本将获得至少8年的支持。



## 1.2.1 下载和安装JDK



### 下载JDK

首先，访问Oracle官网，下载适合自己电脑配置的JDK安装包。



### 安装JDK

下载完成后，点击安装包进行安装。在安装过程中，请确保将JDK安装在默认路径下。



### 完成JDK安装

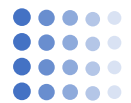
安装完成后，点击桌上的“此电脑”图标，进入文件系统。



### 确认JDK安装

在文件系统中，我们可以在“Program Files”目录下找到“Java”文件夹，表示JDK已成功安装。





# 1.2.1 下载和安装JDK



登录<https://www.oracle.com/cn/java/technologies/downloads/> 网站，  
下载适合自己计算机操作系统的JDK，按照安装向导进行安装。

JDK 24   JDK 21   GraalVM for JDK 24   GraalVM for JDK 21

### Java SE Development Kit 21.0.8 downloads

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions \(NFTC\)](#).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the [Oracle Open Network \(OTN\)](#) and production use beyond the [limited free grants](#) of the OTN license will [require a fee](#).

Linux   macOS   **Windows**

Product/file description	File size	Download
x64 Compressed Archive	186.05 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip</a> (sha256)
x64 Installer	164.42 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe</a> (sha256)
x64 MSI Installer	163.16 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi</a> (sha256)

jdk-21

此电脑 > OS (C:) > Program Files > Java > jdk-21 >

名称	修改日期	类型	大小
bin	2025/2/20 14:53	文件夹	
conf	2025/2/20 14:53	文件夹	
include	2025/2/20 14:53	文件夹	
jmods	2025/2/20 14:53	文件夹	
legal	2025/2/20 14:53	文件夹	
lib	2025/2/20 14:53	文件夹	
LICENSE	2025/2/20 14:53	文件	7 KB
README	2025/2/20 14:53	文件	1 KB
release	2025/2/20 14:53	文件	2 KB



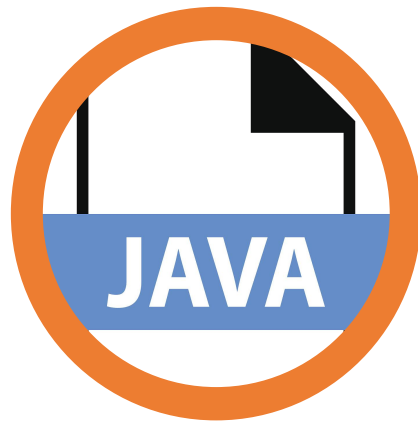


# 1.2.2 配置JDK



## 1. 配置环境变量

在计算机的“环境变量”中，添加**JAVA\_HOME**和**PATH**变量，用于存储JDK的安装路径和运行时路径。



## 2. 设置PATH变量

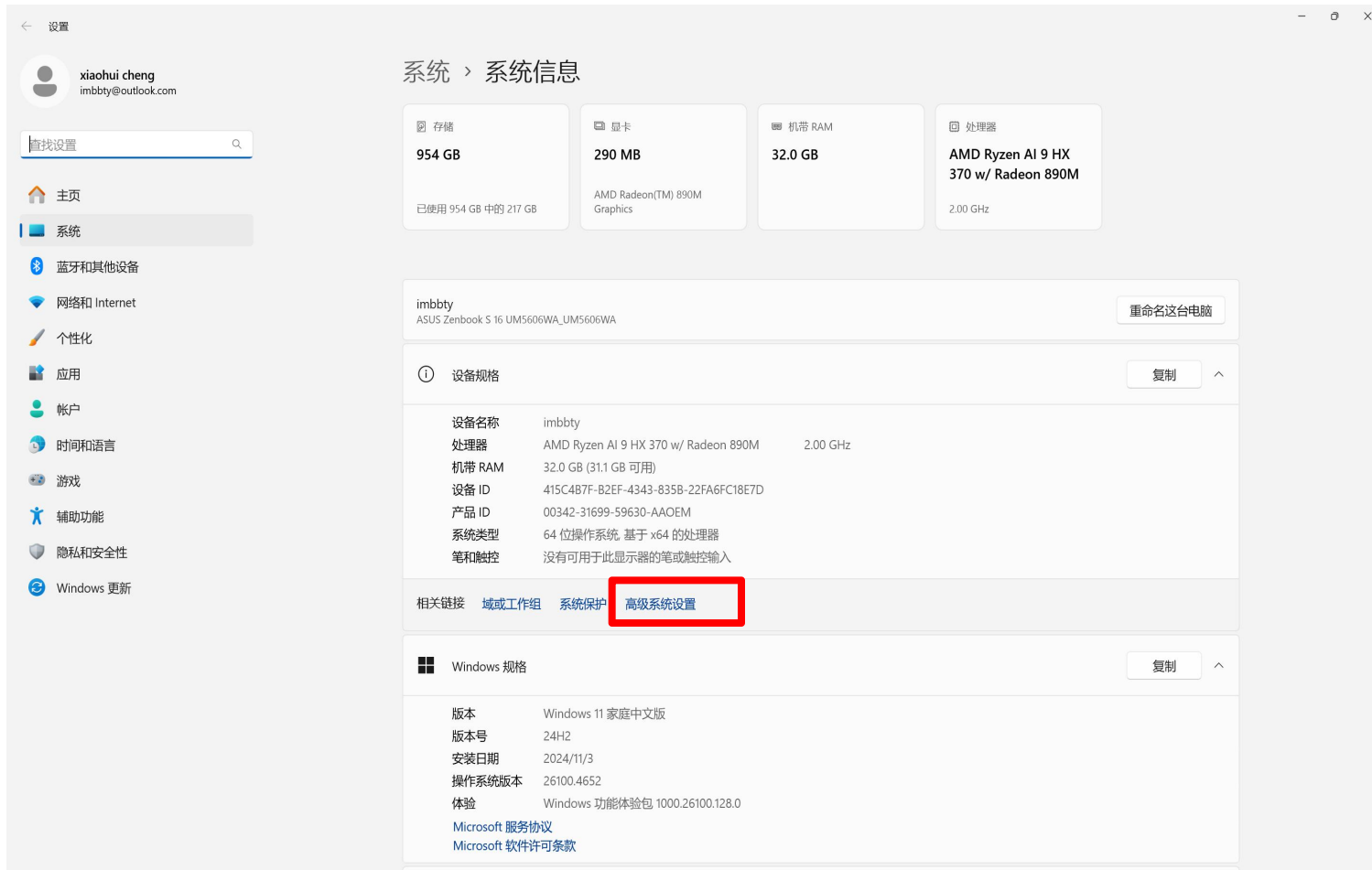
在PATH变量中添加“**%JAVA\_HOME%bin**”，确保系统能够找到并执行JDK中的可执行文件。

## 3. 验证JDK配置

在命令行中输入“**java -version**”，如果一切正常，将显示JDK的版本信息，表示JDK已成功配置。



# 1.2.2 配置JDK



第1步：  
此电脑，右击，选择“属性”

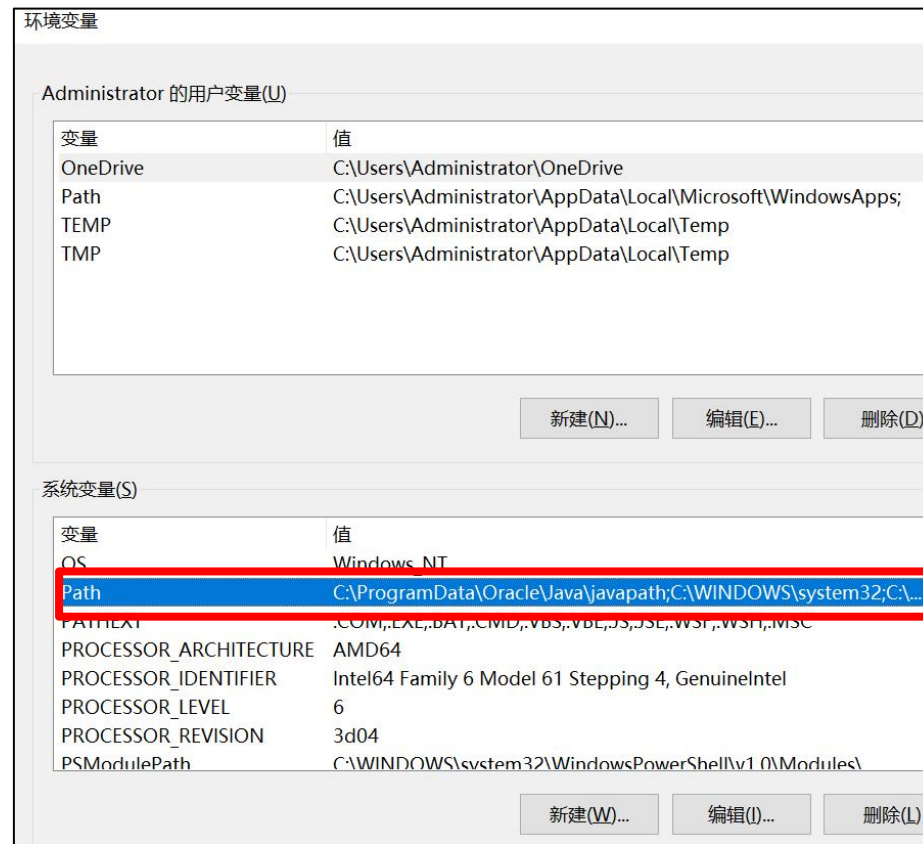
第2步：选择“高级系统设置”



# 1.2.2 配置JDK



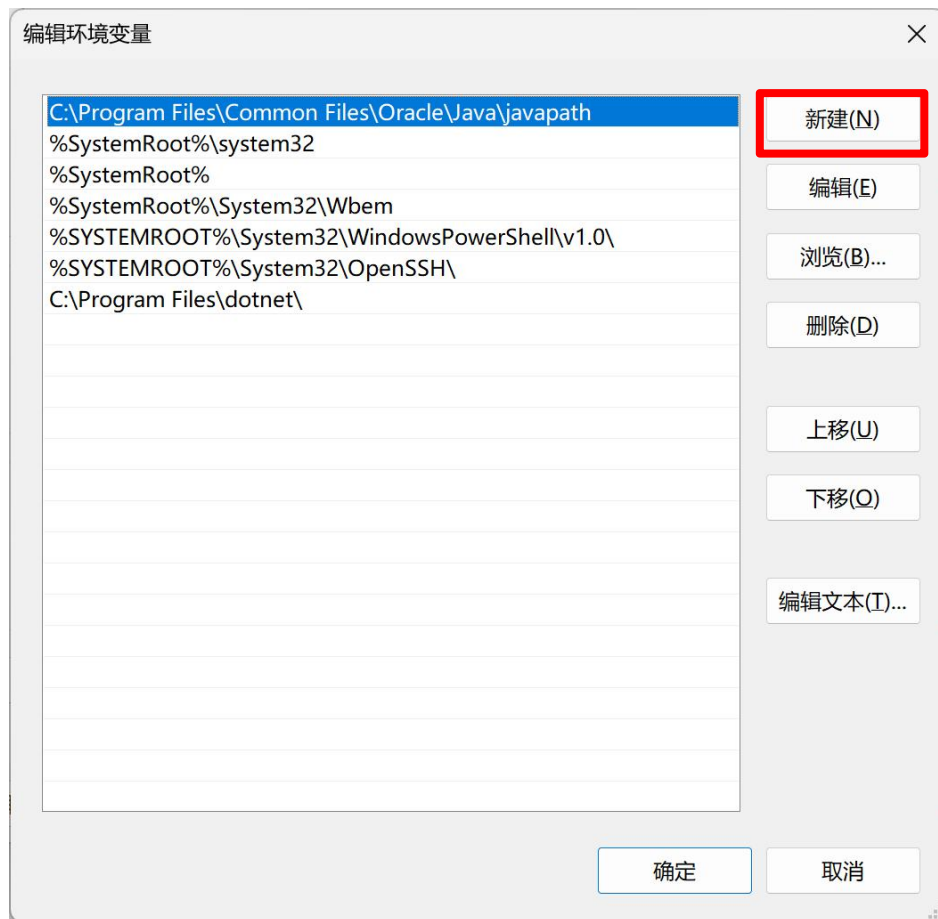
第3步：选择“环境变量”



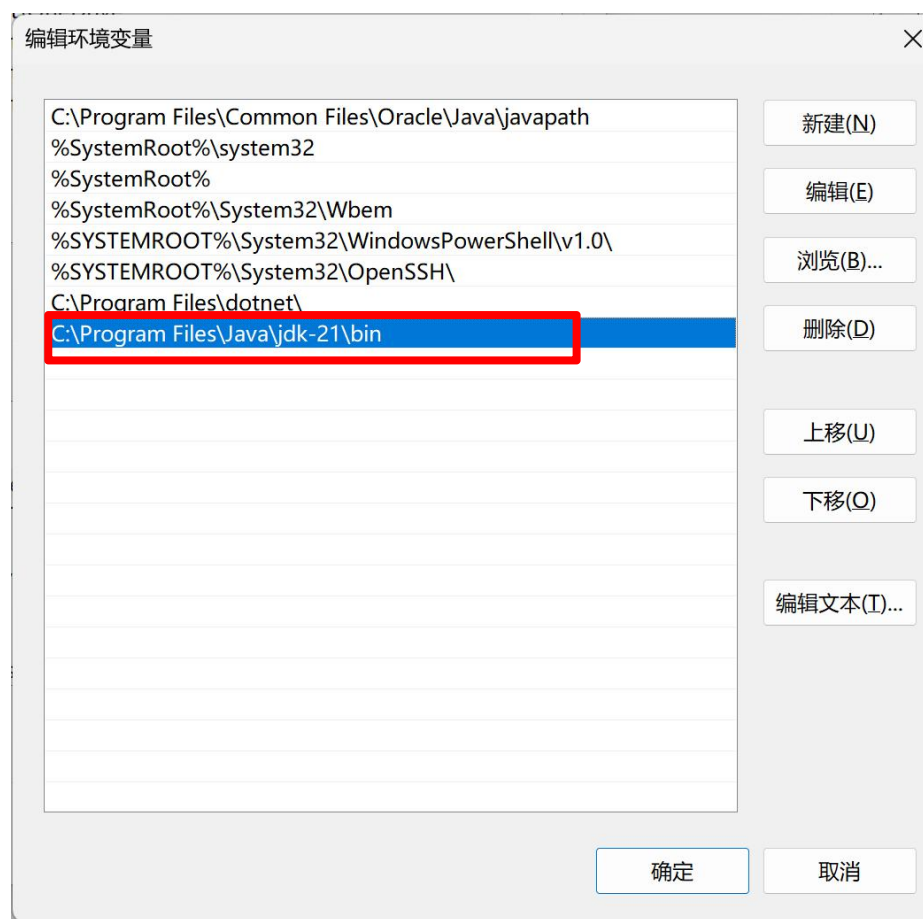
第4步：双击“Path”



## 1.2.2 配置JDK



第5步：点击“新建”



第6步：输入JDK的安装路径



## 1.2.2 配置JDK



```
命令提示符
Microsoft Windows [版本 10.0.26100.3194]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\imbtt>java -version
java version "21.0.6" 2025-01-21 LTS
Java(TM) SE Runtime Environment (build 21.0.6+8-LTS-188)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.6+8-LTS-188, mixed mode, sharing)
```

打开命令行窗口后，输入“java -version”，则显示JDK的版本信息，则环境变量配置成功。





# 无法找到或无法正确执行Java程序



## 原因

这通常是因为没有正确配置环境变量或者环境变量被修改过。

## 解决方案

检查环境变量是否正确配置，确保JAVA\_HOME和PATH变量包含正确的JDK路径。

```
public function imgdeploy()
{
    $PR = Product::select('id', 'image', 'mirror')->whereIn('cat_id',
    $bar = new ProgressBar($this->output, $PR->count());
    $bar->setFormat('debug')->setBarCharacter('<comment>=</comment>');

    $prod_path = public_path( path: 'imgs/products/');
    $products = $PR->get();
    foreach ($products as $product) {
        if(file_exists( filename: $prod_path . 'min/' . $product->image))
            copy( source: $prod_path . 'min/' . $product->image, dest: $
        } else {
            $file_min = file_get_contents( filename: config( key: 'app.url
            file_put_contents( filename: public_path( path: 'imgs/products
        }
        if(file_exists( filename: $prod_path . 'big/' . $product->image))
            copy( source: $prod_path . 'big/' . $product->image, dest: $
        } else {
            $file_min = file_get_contents( filename: config( key: 'app.url
            file_put_contents( filename: public_path( path: 'imgs/products
        }
        foreach($this->model->params->unique('name') as $par) {
            $var = $par->variants->where('id', $par->pivot->variant_id)-
            $names->push(['sort' => $var->sort, 'name' => $var->($this->
        }
        foreach($this->params->unique('name') as $par) {
            $var = $par->variants->where('id', $par->pivot->variant_id)-
            $names->push(['sort' => $var->sort, 'name' => $var->($this->
        }
    }
    try{
        if($variant_id = $this->model->categories->first()->pivot->
        $variant = $this->model->categories->first()->variants->
        $name = $variant->name;
        $gender = $variant->gender;
    } else {
        $variant = $this->model->categories->first();
        $name = $variant->name_sgl;
        $gender = $variant->gender;
    }
    return (object)['name' => $name, 'gender' => $gender]
```



# 无法编译Java程序



## 原因

这可能是由于没有正确的编译工具或者编译工具与JDK不兼容。

## 解决方案

检查是否已安装正确的编译工具，如gcc或clang，并确保它们与JDK兼容。





# 程序运行时出现异常



```

"color: #5e9ca0; text-align: justify;">
style="background-color: #808000;">
n style="color: #ff0000;">Lorem&nbsp;  </span>ipsum</span></p>
le="color: #2e6c80; padding-left: 30px;">
n style="color: #ff00ff;">consectetuer elit, sed diam:</sp
s autem</li>
eum iriur</li>
olor in hendrerit in vul</li>

te velit esse molestie conse</p>
le="color: #2e6c80;">
style="color: #ffff00;">Duis autem vel eum iriure dolor:<
class="editorDemoTable">

an style="background-color: #99cc00;">Nonsectetuer cing</s
>

modo consequat</td>
piscing elit</td>
rong>euismod</strong></td>

>

n>&nbsp;  </strong><strong>Scons adipiscing elit, se</st
```

## 原因

这可能是由于JDK版本与操作系统不兼容或者存在其他运行时错误。

## 解决方案

检查JDK版本与操作系统兼容性，并确保程序在正确的环境中运行。



## 1.3.1 第一个Java程序



在计算机的文件系统中，我们可以通过创建、编辑和保存Java程序来开始编写Java程序。



**编写Java代码：** 打开文本编辑器等，编写程序，将Java代码保存到计算机的文件系统中，文件名必须与**类名**一致，扩展名必须为“.java”。



**编译Java代码：** 在命令行中，进入保存Java代码的目录，并执行命令进行编译。



**运行Java代码：** 在命令行中，进入保存Java代码的目录，并执行命令来运行Java程序：



**程序输出结果：** 运行成功后，控制台将输出“Hello, Java!”这个字符串，表示Java程序已成功运行。

## 【例1-1】编写第一个Java程序。(源代码:HelloJava.java)

```
public class HelloJava {  
    public static void main(String[] args){  
        System.out.println("Hello Java!");  
    }  
}
```

运行结果：

Hello Java!

例1-1的功能是输出一行信息：Hello Java!。下面说明程序的基本结构。

### (1) 类定义

Java程序由类构成，本例中是HelloJava类。用关键字**class**来声明类，**public**指明该类是一个公共类，一个Java文件中可以定义多个类，但最多**只能有一个公共类**。

### (2) main()方法

例1-1定义了一个main()方法。main()方法是应用程序的入口，必须用**public**、**static**和**void**指明，**public**表示所有的类都可以使用这一方法，**static**则说明该方法是一个静态方法，可以通过类名直接调用，**void**指明本方法不返回任何值。Java解释器在没有生成任何对象时，以main()方法做为入口来执行程序。

在main()方法中，String args[]是传递给main()方法的参数，这个参数是String类型数组，名称为args。

### (3) 程序内容

main()方法中只包括一条语句，用来实现字符串的输出。  
System.out.println("Hello Java!");

```
public class HelloJava {  
    public static void main(String[] args){  
        System.out.println("Hello Java!");  
    }  
}
```

# 良好的编程习惯

- 在编写程序时，一行最好只写一条语句，类体的方法的大括号最好也独占一行，并有明显的缩进。
- 标点符号全部为英文输入法。
- 合理使用注释语句。
- 命名尽量是有意义的。
- Java是区分大小写。



## 1.3.2 编译和运行Java程序



### 编译Java程序

当我们要将Java程序从文本形式转换为可执行文件时，就需要使用到编译器。

### 执行Java程序

在命令行中，需要进入到保存Java文件的目录，并执行以下命令来运行Java程序。

```
C:\Users\imbtt>d:
D:\>cd JavaCode
D:\JavaCode>javac HelloJava.java
D:\JavaCode>java HelloJava
Hello Java!
D:\JavaCode>
```

### 传递参数

如果Java程序需要接受命令行参数，可以在运行程序时，将参数传递给Java程序。

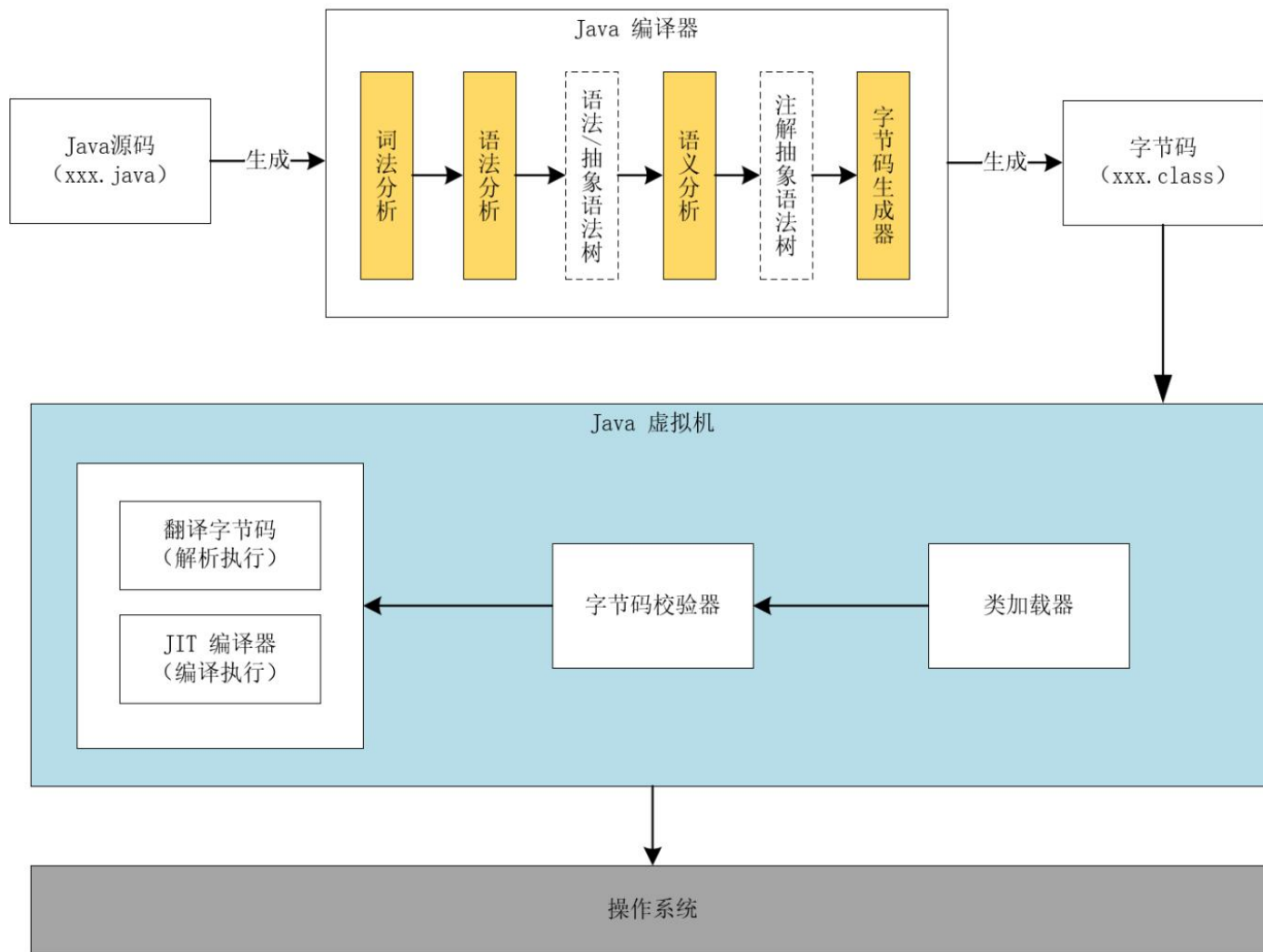
### 显示输出结果

在命令行中，可以查看Java程序的输出结果，包括执行成功或失败的信息等。

```
D:\JavaCode>javac HelloJava.java
D:\JavaCode>java HelloJava HelloWorld
HelloWorld
D:\JavaCode>java HelloJava InputParameter
InputParameter
```



## 1.3.2 编译和运行Java程序







# 编写Java程序注意事项



## 注释规范

在编写Java程序时，要使用特定的注释规范，以帮助其他开发人员更好地理解代码。

## 类型安全

在编写Java程序时，要确保变量和方法的类型安全，以避免潜在的错误和异常情况。

## 命名规则

在编写Java程序时，要遵循特定的命名规则，以方便代码的可读性和维护性。

## 缩进和格式

在编写Java程序时，要使用适当的缩进和格式，以使代码结构清晰、易于阅读。

## 性能优化

在编写Java程序时，可以考虑进行性能优化，以提高程序的运行效率和响应速度。



# 1.4.1 下载和安装IntelliJ IDEA



## 下载IntelliJ IDEA

访问IntelliJ IDEA的官方网站，点击“下载”按钮，选择适合自己电脑的版本进行下载。

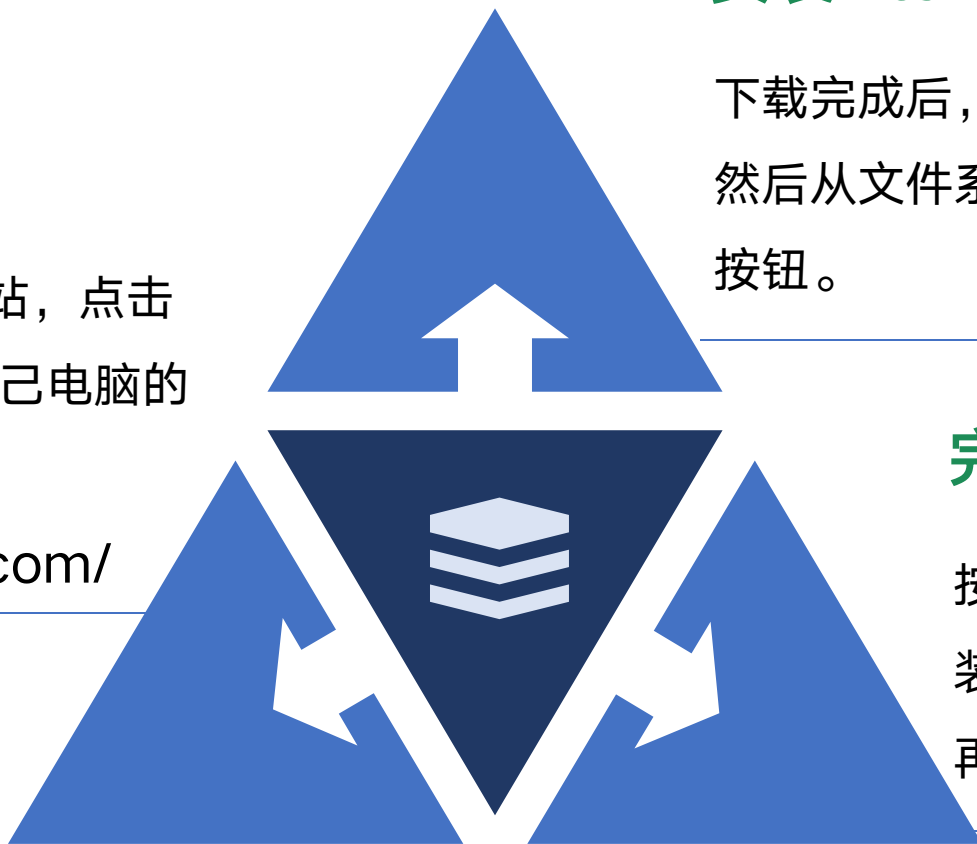
<https://www.jetbrains.com/>

## 安装IntelliJ IDEA

下载完成后，将文件保存在电脑的特定位置，然后从文件系统进入该文件，点击“安装”按钮。

## 完成IntelliJ IDEA安装

按照提示步骤，将IntelliJ IDEA安装到电脑中，并确保在完成安装后，再次启动计算机。





# 1.4.1 下载和安装IntelliJ IDEA



Screenshot of the JetBrains website (https://www.jetbrains.com) showing the Developer Tools section. The URL bar is highlighted with a red box. The 'Developer Tools' menu item in the navigation bar is also highlighted with a red box. The 'IntelliJ IDEA' link in the 'JETBRAINS IDEs' list is highlighted with a red box.

**JETBRAINS** AI **Developer Tools** Team Tools Education Solutions Support Store

**JETBRAINS IDEs**

- All IDEs
- CLion
- DataGrip
- DataSpell
- Fleet
- GoLand
- IntelliJ IDEA**
- PhpStorm
- PyCharm
- Rider
- RubyMine
- RustRover
- WebStorm

**PLUGINS & SERVICES**

- All Plugins
- AI in IDEs
- IDE Themes
- Big Data Tools
- Code With Me
- RiderFlow
- Scala
- Toolbox App
- Grazie
- Junie

**.NET & VISUAL STUDIO**

- Rider
- ReSharper
- ReSharper C++
- dotCover
- dotMemory
- dotPeek
- dotTrace
- .NET Tools Plugins

**LANGUAGES & FRAMEWORKS**

- Kotlin
- Ktor
- Exposed
- MPS
- Compose
- Multiplatform

**DataGrip**  
Many databases, one tool →

**Junie by JetBrains**  
Your smart coding agent for JetBrains IDEs →

Not sure which tool is best for you?  
Whichever technologies you use, there's a JetBrains tool to match

[Find your tool](#)



# 1.4.1 下载和安装IntelliJ IDEA



We're committed to giving back to our wonderful community, which is why IntelliJ IDEA Community Edition is completely free to use

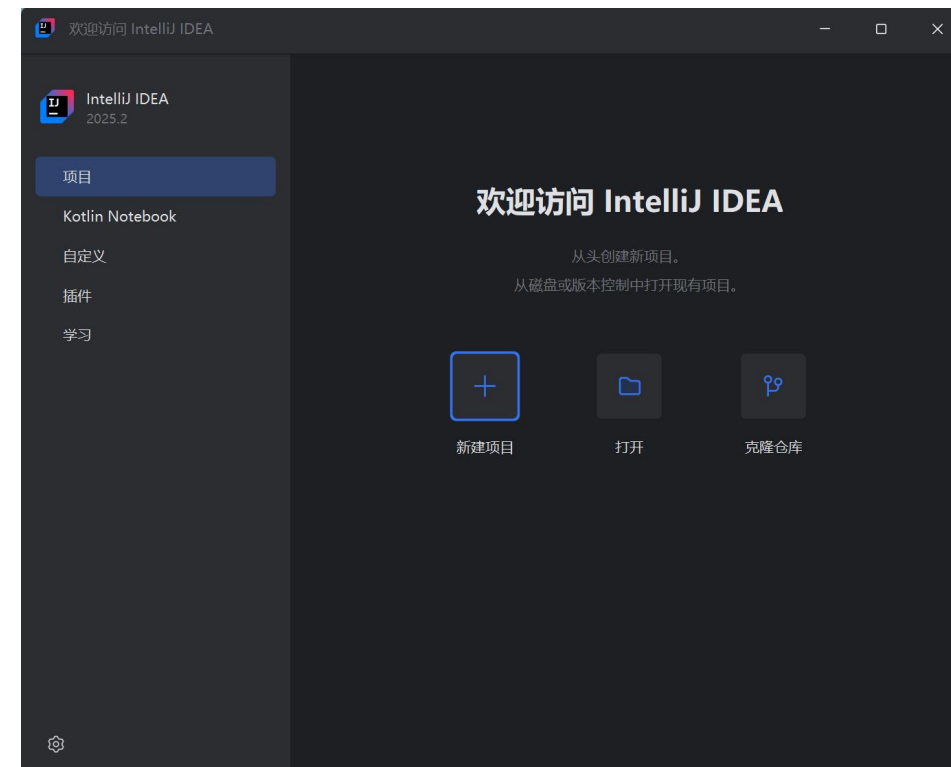
 **IntelliJ IDEA Community Edition**


The IDE for Java and Kotlin enthusiasts

Download

.exe (Windows) ▼

Free, built on open source



 idealC-2025.2.exe



## 1.4.2 创建项目和文件



### 1. 创建新项目

在IntelliJ IDEA中，选择“文件”菜单中的“新建”选项，将弹出“新建项目”对话框。

### 3. 创建新文件

在IntelliJ IDEA中，选择“文件”菜单中的“新建”选项，将弹出“新建文件”对话框。



### 2. 设定项目名称和位置

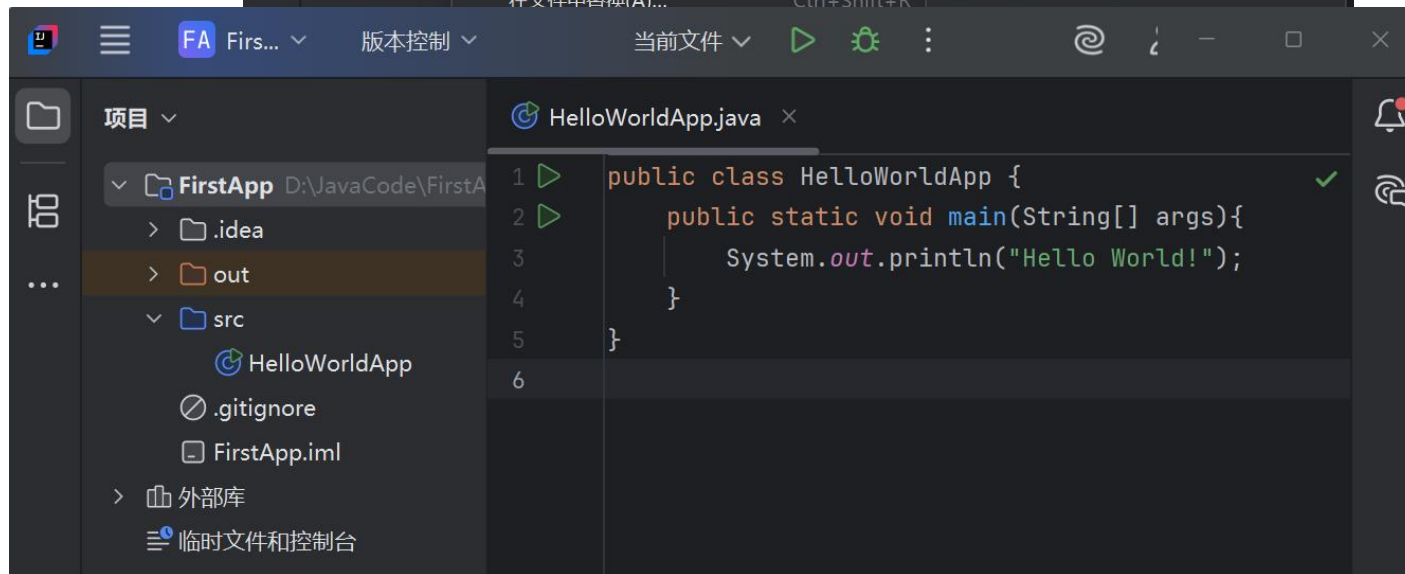
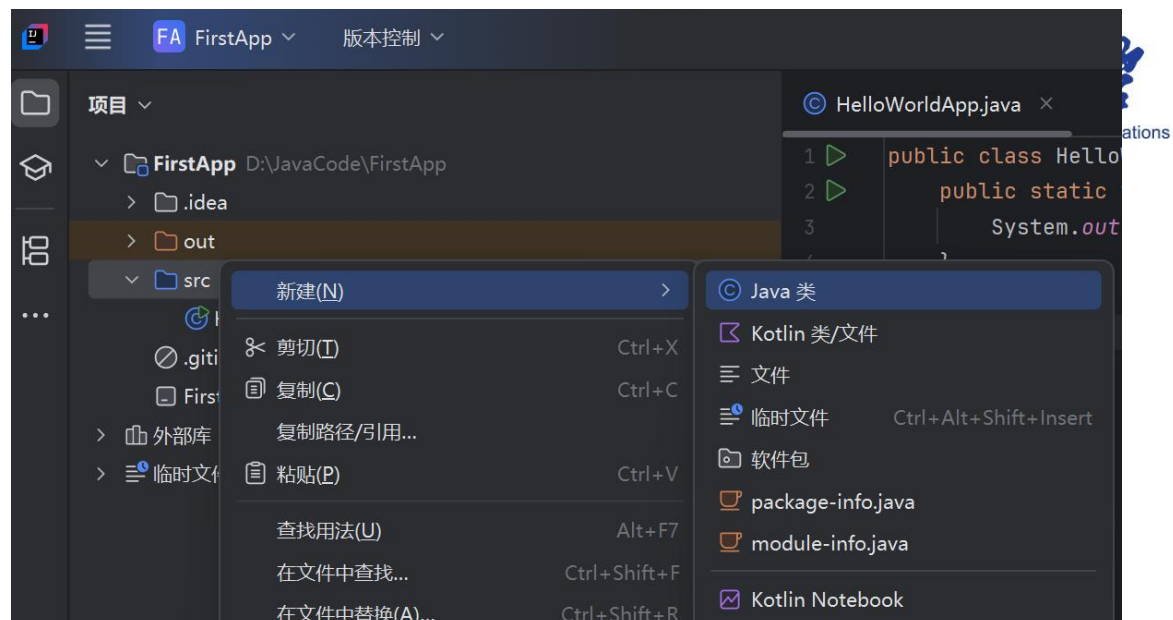
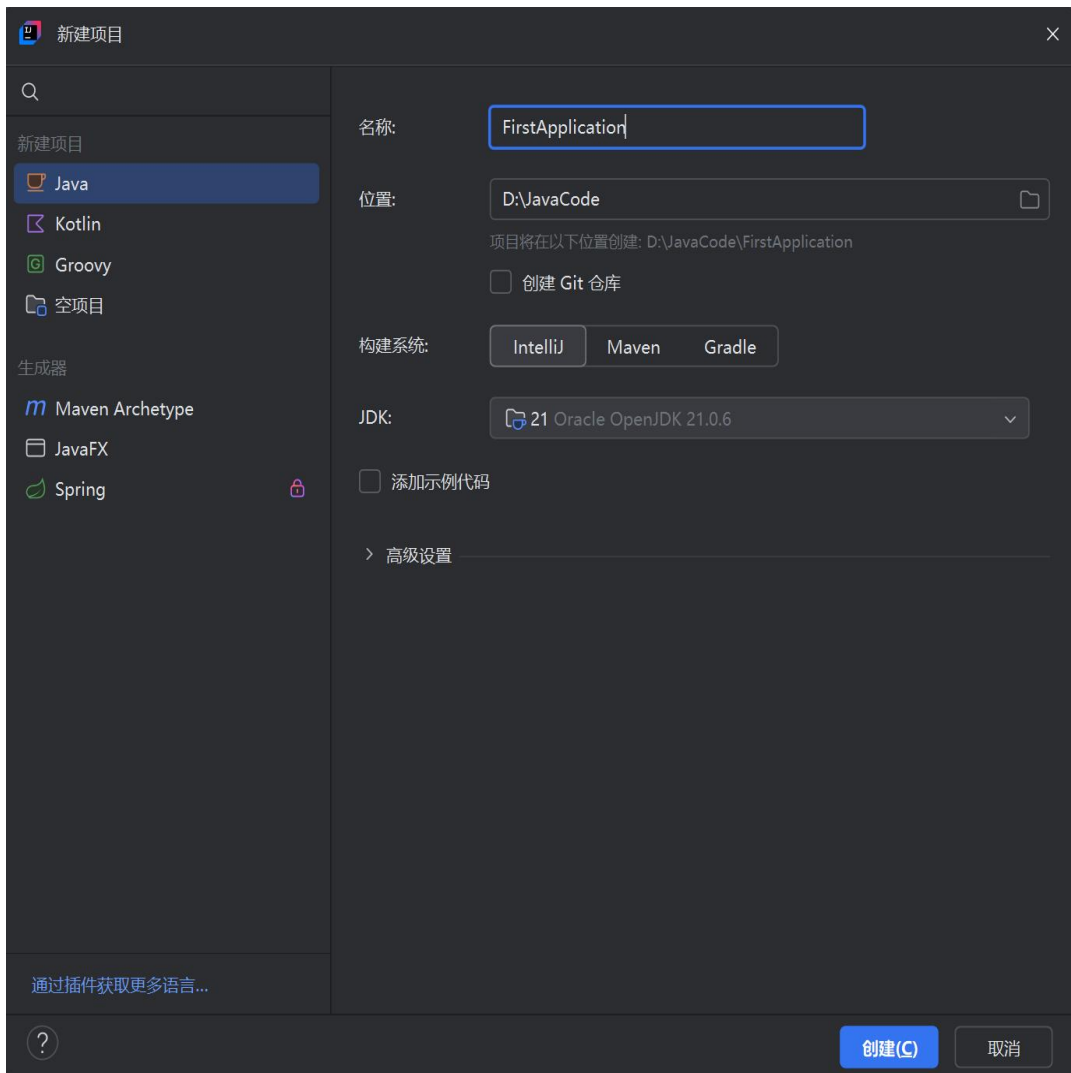
在对话框中，输入项目的名称和保存的位置，然后单击“确定”按钮以创建新项目。

### 4. 设定文件名称和内容

在对话框中，输入文件的名称和内容，然后单击“确定”按钮以创建新文件。



# 1.4.2 创建项目和文件



```
"C:\Program Files\Java\jdk-21\bin\java.exe"  
Hello World!
```

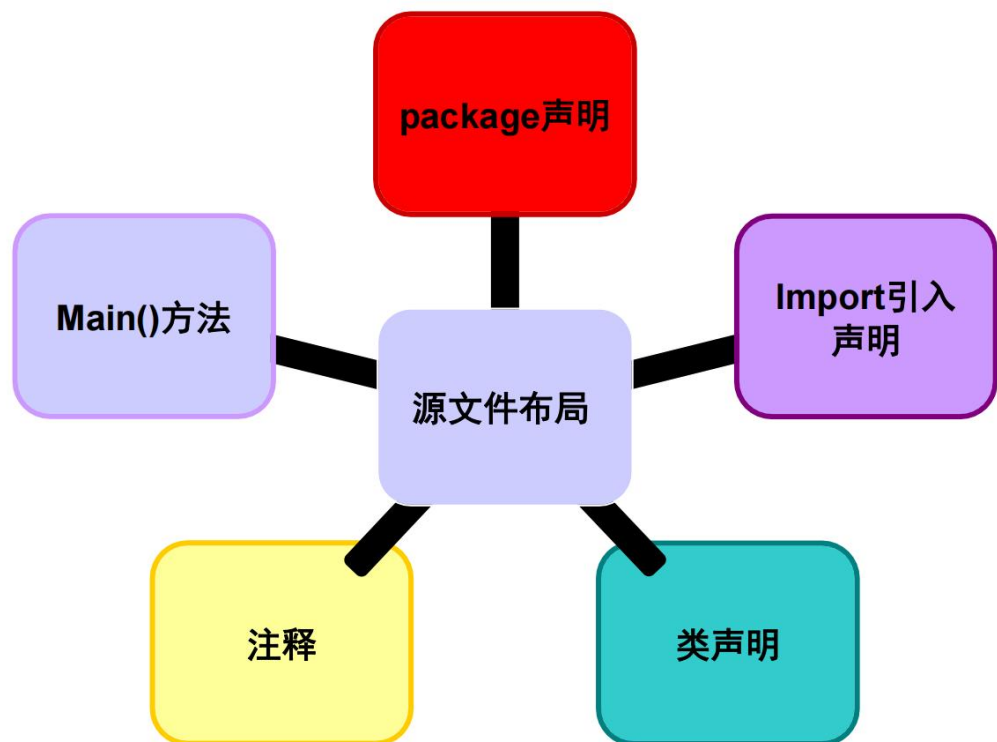




## 1.4.2 创建项目和文件



### Java 源文件布局



```
package demo;
import java.util.Scanner;
public class HelloJava {
    //输入两个整数进行求和
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        int n1 = s.nextInt();
        int n2 = s.nextInt();
        int sum = n1 + n2;
        System.out.println(sum);
    }
}
```





# 常见集成开发环境



01

## IntelliJ IDEA

IntelliJ IDEA是一款流行的集成开发环境，由JetBrains公司开发。它具有强大的代码编辑器、智能提示、自动完成等特性。

02

## Visual Studio

Visual Studio是微软公司开发的集成开发环境，适用于Windows平台。它具有丰富的功能和便捷的界面，使开发者能够更高效地开发软件。

03

## Eclipse

Eclipse平台是一个开放源代码的、基于Java的可扩展开发平台。其核心设计为一个框架和一组服务，通过插件组件构建定制化开发环境，默认集成Java开发工具（JDK）等基础插件，常用于Java集成开发环境（IDE）。



# 使用集成开发环境



## 1. 安装软件

下载并安装所需的软件，包括代码编辑器、数据库管理工具等，以便在集成开发环境中使用。

## 2. 打开软件

打开下载并安装的软件，进入其主界面，等待软件自动扫描已安装的软件，并列出可用的功能和选项。

## 创建新项目

选择“文件”菜单中的“新建”选项，将弹出“新建项目”对话框，输入项目名称和保存位置，点击“确定”按钮。

## 创建新文件

选择“文件”菜单中的“新建”选项，将弹出“新建文件”对话框，输入文件名称和内容，点击“确定”按钮。

## 保存文件

在完成文件编辑后，点击“保存”按钮，系统将自动将更改保存到文件中，并提示您是否要立即编译和运行该文件。

## 运行程序

如果您希望立即运行程序，请点击“运行”按钮，系统将执行您的代码并显示运行结果。如果您希望手动运行程序。



## Java21 API在线文档:

<https://docs.oracle.com/en/java/javase/21/docs/api/index.html>

JDK 24 JDK 21 GraalVM for JDK 24 GraalVM for JDK 21

## Java SE Development Kit 24.0.2 downloads

JDK 24 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#) (NFTC).

JDK 24 will receive updates under these terms, until September 2025, when it will be superseded by JDK 25.

Linux macOS Windows

Product/file description	File size	Download
ARM64 Compressed Archive	229.37 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_linux-aarch64_bin.tar.gz">https://download.oracle.com/java/24/latest/jdk-24_linux-aarch64_bin.tar.gz</a> (sha256)
ARM64 RPM Package	228.98 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_linux-aarch64_bin.rpm">https://download.oracle.com/java/24/latest/jdk-24_linux-aarch64_bin.rpm</a> (sha256) (OL 9 GPG Key)
x64 Compressed Archive	232.15 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.tar.gz">https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.tar.gz</a> (sha256)
x64 Debian Package	200.21 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.deb">https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.deb</a> (sha256)
x64 RPM Package	231.71 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.rpm">https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.rpm</a> (sha256) (OL 9 GPG Key)

Documentation Download

jdk-21.0.8\_doc-all

今天

index.html

specs

api

legal

resources

离线API文档

## IDEA 常用快捷键

- 组合几个键一起按下来完成某件事，可以提高开发效率。

快捷键	功能效果
main/psvm、sout、...	快速键入相关代码
Ctrl + D	复制当前行数据到下一行
Ctrl + Y	删除所在行，建议用Ctrl + X
Ctrl + ALT + L	格式化代码
ALT + SHIFT + ↑ , ALT + SHIFT + ↓	上下移动当前代码
Ctrl + /	单行注释（对当前行进行注释）
Ctrl + Shift + /	对选中的代码进行多行注释。



# THANKS