



干就对了！

《Java从入门到精通》

讲师：徐磊



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

干就对了

- ◆ Java是一门编程语言，本质就是一个工具
- ◆ 直接解决问题，带着问题学习
- ◆ 高效、清楚、实用

让初学者学习起来更加轻松，并能快速入门

为什么叫干就

Java



举例



如何表示微信钱包?

如何收红包?

举例



如何实现私聊?

举例



如何实现群聊?

举例



如何实现登录账户?

如何进行转账?

《干就对了，Java从入门到精通》



即使是面试薪酬 **30万+** 的Java程序员，对于基础这一块的问题本课程足以应对。

JavaSE
基础大纲

day01-Java环境搭建、IDEA、Java语言

day02-Java语法、运算符、随机数

day03-分支、循环、控制关键字

day04-数组详解、案例训练

day05-方法详解、案例训练

day06-面试常见编程案例题训练

day07-面向对象基础

day08-常用API

day09-综合项目实战

JavaSE
加强大纲

前期内容回顾

day01-static、单例、代码块、继承

day02-权限修饰符、抽象类、接口

day03-多态、内部类、常用API

day04-常用API、正则表达式、Lambda

day05-Collection集合、数据结构、List、泛型

day06-Set、案例、Collections、Map、集合嵌套

day07-不可变集合、Stream流、异常

day08-Logback日志框架、阶段项目

day09-File、递归、IO流(一)

day10-IO二

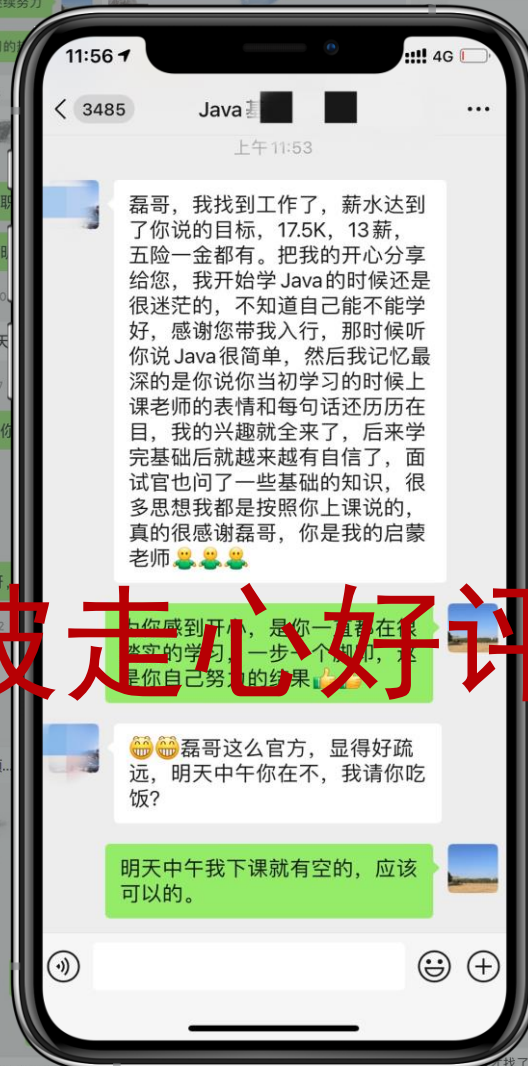
day11-多线程一

day12-网络编程

day13-Java高级技术：单元测试、反射、注解、动态代理

day14-XML、解析、设计模式等

来一波走心好评~~





咱们一起开干吧...



目录

Contents

➤ Java课程介绍

➤ Java入门

◆ Java入门环境的搭建：JDK简介、安装

◆ Java入门程序实现前的准备知识

◆ Java入门程序-HelloWorld

◆ HelloWorld程序常见问题

➤ IDEA开发工具的使用

➤ Java语言

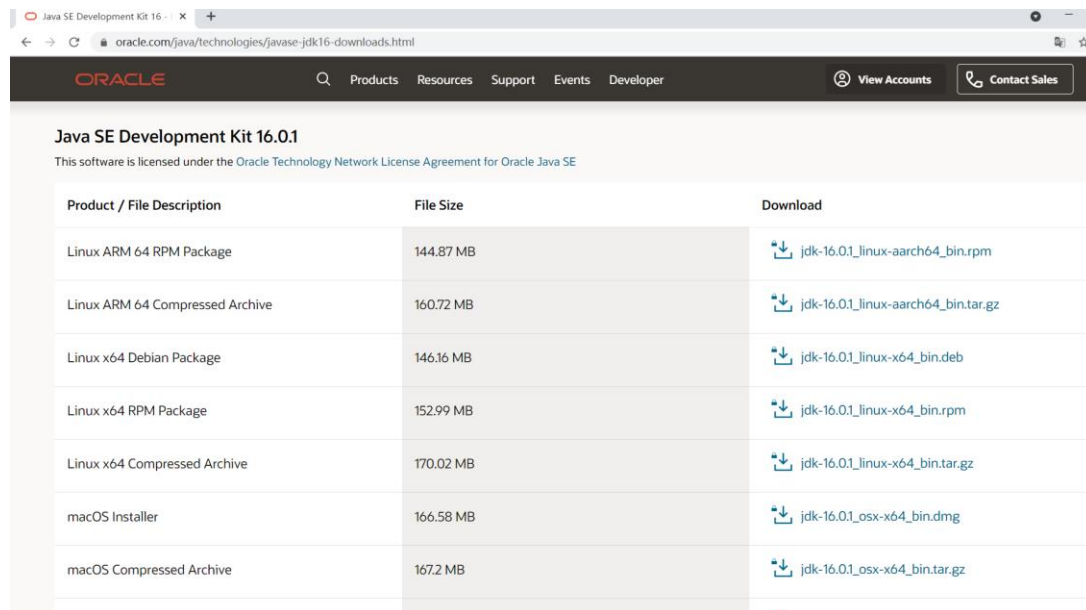
如何使用Java

- Java语言的产品是JDK (Java Development Kit : Java开发工具包)，必须有JDK才能使用Java语言。

如何获取JDK

- 去Oracle官网下载JDK。

<https://www.oracle.com/java/technologies/javase-jdk16-downloads.html>



The screenshot shows the Oracle website's download page for Java SE Development Kit 16.0.1. It includes a navigation bar with 'ORACLE' and links for 'Products', 'Resources', 'Support', 'Events', and 'Developer'. Below the navigation bar, the page title is 'Java SE Development Kit 16.0.1' with a note about the license. A table lists various download options for different operating systems and architectures, including Linux ARM 64, Linux x64, and macOS, with file sizes and download links.

Product / File Description	File Size	Download
Linux ARM 64 RPM Package	144.87 MB	jdk-16.0.1_linux-aarch64_bin.rpm
Linux ARM 64 Compressed Archive	160.72 MB	jdk-16.0.1_linux-aarch64_bin.tar.gz
Linux x64 Debian Package	146.16 MB	jdk-16.0.1_linux-x64_bin.deb
Linux x64 RPM Package	152.99 MB	jdk-16.0.1_linux-x64_bin.rpm
Linux x64 Compressed Archive	170.02 MB	jdk-16.0.1_linux-x64_bin.tar.gz
macOS Installer	166.58 MB	jdk-16.0.1_osx-x64_bin.dmg
macOS Compressed Archive	167.2 MB	jdk-16.0.1_osx-x64_bin.tar.gz

JDK提供了编译和执行Java程序的工具

此电脑 > 本地磁盘 (D:) > soft > Java > jdk-16.0.1 > bin		
名称		类型
jaccesswalker.exe		应用程序
jar.exe		应用程序
jarsigner.exe		应用程序
java.dll		应用程序扩展
java.exe	2、执行工具	应用程序
javaaccessbridge.dll		应用程序扩展
javac.exe	1、编译工具	应用程序
javadoc.exe		应用程序
javajpeg.dll		应用程序扩展
java.exe		应用程序



总结

1. Java的产品叫什么？去哪里下载安装？

- JDK (Java Development Kit) 开发工具包、Oracle官网

2. 安装JDK有哪些要求？

- 不要安装在有空格或中文名称的路径下。

3. JDK中两个重要的工具是谁？

- javac 编译工具、java 执行工具



目录

Contents

➤ Java课程介绍

➤ Java入门

◆ Java开发环境的搭建：JDK简介、安装

◆ Java入门程序实现前的准备知识

◆ Java入门程序-HelloWorld

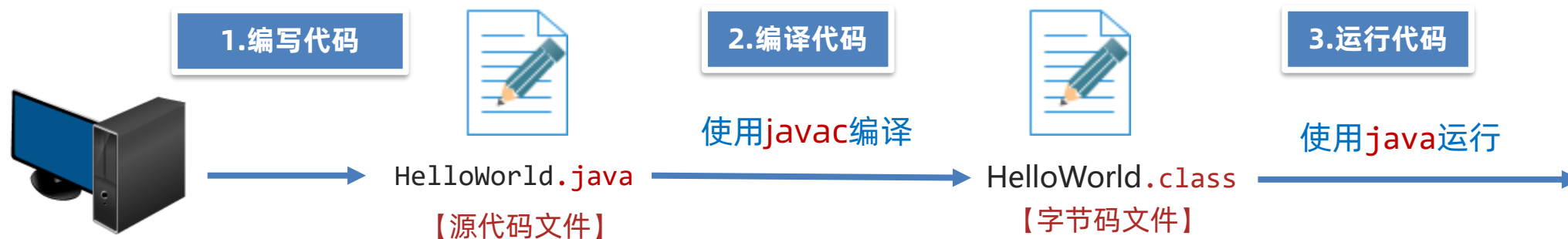
◆ HelloWorld程序常见问题

➤ IDEA开发工具的使用

➤ Java语言

Java 程序开发的三个步骤

- 开发 Java 程序，需要三个步骤：编写代码，编译代码，运行代码



注意事项：

- 第一个 Java 程序建议使用记事本书写。
- 建议代码文件名全英文，首字母大写，满足驼峰模式，源代码文件的后缀必须是 `.java`。

编写代码

- 第一个程序的代码如下：



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

The screenshot shows a Notepad window titled 'HelloWorld.java - 记事本'. The menu bar includes '文件(E)', '编辑(E)', '格式(O)', '查看(V)', and '帮助(H)'. The code is written in a standard font. The status bar at the bottom indicates '第 5 行, 第 2 列', '100%', 'Windows (CRLF)', and 'UTF-8'.

注意：文件名称必须与代码中的类名称一致。

保存文件：ctrl + s

编译代码、运行代码

- ① 编译: javac 文件名.java

范例: **javac HelloWorld.java**

- ② 运行: java 类名

范例: **java HelloWorld**

```
D:\code>javac HelloWorld.java  
D:\code>java HelloWorld  
Hello World
```

HellWorld程序介绍

class: 定义一个类，后面跟上的HelloWorld是**类名称**

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

程序执行时的**入口点**，main方法也称之为**主方法**

打印语句，使程序在**控制台打印双引号所包裹的内容**



总结

1、开发一个Java程序要经历哪些步骤?

- 编写、编译(javac)、运行(java)

2、Java代码编写有什么基本要求?

- 文件名称的后缀必须是java结尾。
- 文件名称必须与代码的类名称一致。
- 必须使用英文模式下的符号。



Java入门程序练习

🕒 10 分钟

需求：完成入门程序的编写、编译、运行。

```
D:\code>javac HelloWorld.java  
D:\code>java HelloWorld  
Hello World
```



目录

Contents

➤ Java入门

- ◆ JDK介绍、安装
- ◆ Java入门程序-HelloWorld
- ◆ HelloWorld程序常见问题
- ◆ 补充知识：JDK、JRE、跨平台原理
- ◆ 补充知识：Path环境变量的注意事项

➤ IDEA开发工具的使用

➤ Java语言

HelloWorld案例常见错误

1. 扩展名被隐藏。
2. 文件名和类名不一致。
3. 大小写错误，单词拼写错误，存在中文符号，找不到main方法。
4. 括号不配对。
5. 编译或者执行命令使用不当。

HelloWorld案例常见错误

- 扩展名被隐藏



HelloWorld案例常见错误

- 文件名和类名不一致



```
D:\code>javac HelloWorld.java
HelloWorld.java:1: 错误: 类 HelloWolrpd 是公共的, 应在名为 HelloWorld.java 的文件中声明
public class HelloWolrpd {
1 个错误
D:\code>
```

HelloWorld案例常见错误

- 大小写错误，单词拼写错误，存在中文符号，找不到main方法



```
Public class HelloWorld {  
    public static void mian(String[] args){  
        System.out.println("Hello World");  
    }  
}
```

```
D:\code>javac HelloWorld.java  
HelloWorld.java:3: 错误: 编码 GBK 的不可映射字符 (0x9B)  
        System.out.println("Hello World")  
HelloWorld.java:1: 错误: 需要 class、interface、enum 或 record  
Public class HelloWorld {  
HelloWorld.java:3: 错误: 需要';'  
        System.out.println("Hello World")
```

HelloWorld案例常见错误

- 括号不配对



```
HelloWorld.java - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)
public class HelloWorld {
    public static void main(String[] args){
        System.out.println("Hello World");
    }
}
```

第 4 行, 第 2 列 100% Windows (CRLF) UTF-8

```
D:\code>javac HelloWorld.java
HelloWorld.java:5: 错误: 进行语法分析时已到达文件结尾
}
~
1 个错误
D:\code>_
```

HelloWorld案例常见错误

- 编译、执行使用不当。

```
D:\code>javac HelloWorld.java
```

```
D:\code>java HelloWorld.class
```

```
错误: 找不到或无法加载主类 HelloWorld.class
```

```
原因: java.lang.ClassNotFoundException: HelloWorld.class
```

```
D:\code>java HelloWorld.java  
Hello World
```



**JDK 11开始支持java直接运行
源代码文件**

```
D:\code>_
```



目录

Contents

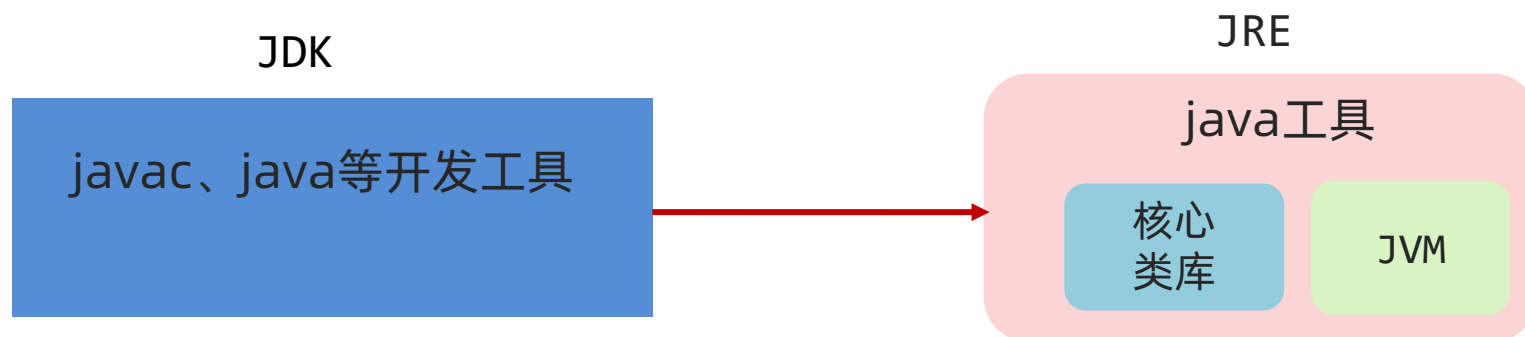
➤ Java入门

- ◆ JDK介绍、安装
- ◆ Java入门程序-HelloWorld
- ◆ HelloWorld程序常见问题
- ◆ 补充知识：JDK和JRE的关系、跨平台原理
- ◆ 补充知识：JDK安装时Path环境变量的注意

➤ IDEA开发工具的使用

➤ Java语言

JDK、JRE的关系



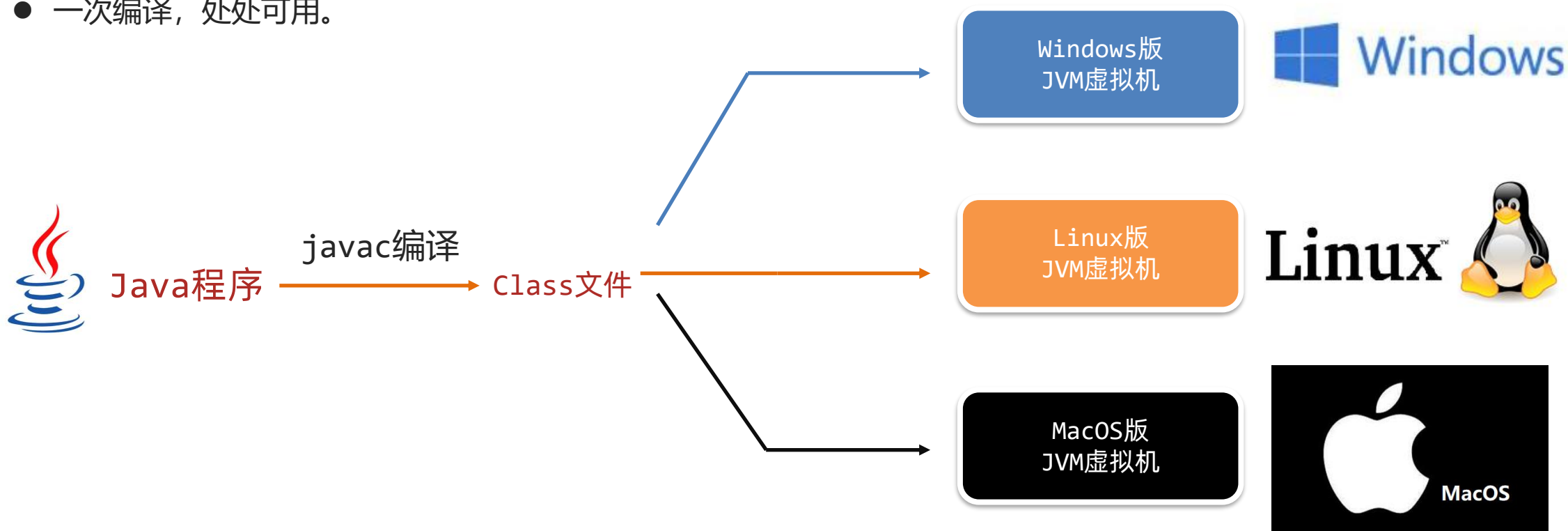
- JVM (Java Virtual Machine) : Java虚拟机, 真正运行Java程序的地方。
- JDK (Java Development Kit) : Java开发工具, 包含java编译器、java虚拟机、以及java帮我们写好的程序...等所有java的开发、运行需要的东西。
- JRE (Java Runtime Environment) : Java运行时环境, 只包含java虚拟机、以及java帮我们写好的程序等java运行所需的東西。

结论:

- 我们现阶段只要安装了JDK即可拥有java程序开发和运行需要的全部环境。
- 如果是在企业的部署环境下, 其实只需要安装JRE就可以运行编译好的Java程序了。

跨平台工作原理

- 一次编译，处处可用。





总结

1. JDK和JRE的关系是什么样的?

- JDK中会关联JRE。
- 安装JDK时包含了: java、javac、jvm和核心类库等全部java开发、运行的环境。
- JRE中只包含了: java、jvm和核心类库等Java的运行环境。

2. Java的跨平台是什么含义，如何实现的?

- 一次编译、处处可用
- 不同平台有对应的JRE环境。



目录

Contents

➤ Java入门

- ◆ JDK介绍、安装

- ◆ Java入门程序-HelloWorld

- ◆ HelloWorld程序常见问题

- ◆ 补充知识：JDK、JRE、跨平台原理

- ◆ 补充知识：JDK安装时Path环境变量的注意

➤ IDEA开发工具的使用

➤ Java语言

Path环境变量的作用

- Path环境变量是用来配置命令的路径的，这样可以在命令行窗口中的任何位置直接使用这些命令。

JDK安装时Path需要注意的事项:

- 目前较新的JDK安装时会自动配置javac、java命令的路径到Path环境变量中去，所以javac、java可以直接使用。
- 但是以前下载的老版本的JDK是没有自动配置的，此时必需要自己配置Path环境变量。

环境变量的位置在：我的电脑 -> 属性 -> 高级系统设置 -> 高级 -> 环境变量。



建议还是自己配置环境变量

- ① **JAVA_HOME**: 告诉操作系统JDK安装在了哪个位置 (未来其他技术要通过这个找JDK)

```
JAVA_HOME C:\soft\java\jdk-16.0.1
```

- ② **Path**: 告诉操作系统JDK提供的javac(编译)、java(执行)命令安装到了哪个位置

```
Path C:\soft\java\jdk-16.0.1\bin
```

或者

```
Path %JAVA_HOME%\bin
```

注意: 新版本的JDK只是自动配置了Path, 没有自动配置JAVA_HOME。

如何检测是否配置成功

- 打开cmd，输入javac -version 及 java -version分别看版本提示。
- 如果版本提示都是自己安装版本号，则代表JDK环境搭建成功。

```
D:\code>javac -version
javac 16.0.1

D:\code>java -version
java version "16.0.1" 2021-04-20
Java(TM) SE Runtime Environment (build 16.0.1+9-24)
Java HotSpot(TM) 64-Bit Server VM (build 16.0.1+9-24, mixed mode, sharing)
```



总结

1. 什么是Path环境变量?

- 在操作系统中配置的命令路径，实现在系统中直接访问命令。

2. JDK安装时，环境变量需要注意什么?

- 较新版本的JDK会自动配置PATH环境变量。
- 建议自己配置“JAVA_HOME”、“Path”



目录

Contents

➤ Java入门

➤ IDEA开发工具的使用

- ◆ IntelliJ IDEA 开发工具概述、安装

- ◆ IDEA 编写第一个程序

- ◆ IDEA 字体、主题、背景色设置

- ◆ IDEA 常用快捷键介绍

- ◆ IDEA 工程、模块常用操作

➤ Java语言

之前的开发工具

- 文本编辑工具：记事本 、 UltraEdit 、 EditPlus 、 TextPad 、 NotePad++、 sublime..
- 存在的问题：编写代码时没有错误提醒、没有智能提示、需要自己进行编译、执行。功能不够强大。

集成开发环境（ IDE， Integrated Development Environment ）

- 把代码编写，编译，执行等多种功能综合到一起的开发工具，可以进行代码智能提示，错误提醒，项目管理等等。
- 常见Java的IDE工具有：Eclipse、MyEclipse、IntelliJ IDEA、Jbuilder、NetBeans等。

IntelliJ IDEA开发工具

- **IntelliJ IDEA** 一般简称IDEA，是Java编程语言开发的集成环境，在业界被公认为最好的java开发工具。
- IDEA 在代码错误提醒，智能代码补全等多方面表现的都非常优秀，是Java开发企业级首选工具。



IDEA 下载

下载: <https://www.jetbrains.com/idea/>

安装: 傻瓜式安装, 建议修改安装路径 (不要安装在有空格和中文的路径下)。





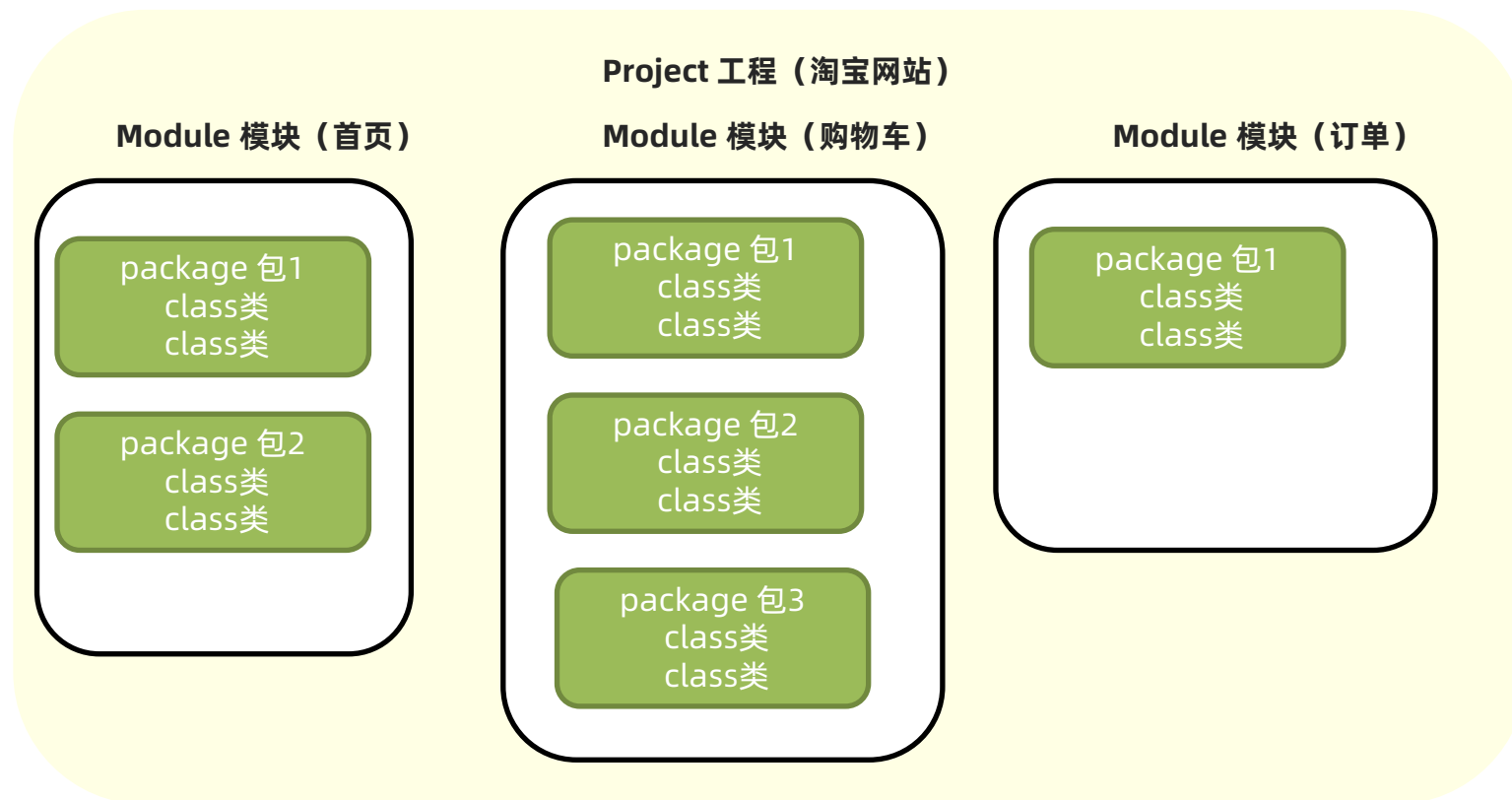
目录

Contents

- **Java入门**
- **IDEA开发工具的使用**
 - ◆ IntelliJ IDEA 开发工具概述、安装
 - ◆ **IDEA 编写第一个程序**
 - ◆ IDEA 字体、主题、背景色设置
 - ◆ IDEA常用快捷键介绍
 - ◆ IDEA工程、模块常用操作
- **Java语言**

IDEA项目结构介绍

- project (项目、工程)
- module (模块)
- package (包)
- class (类)

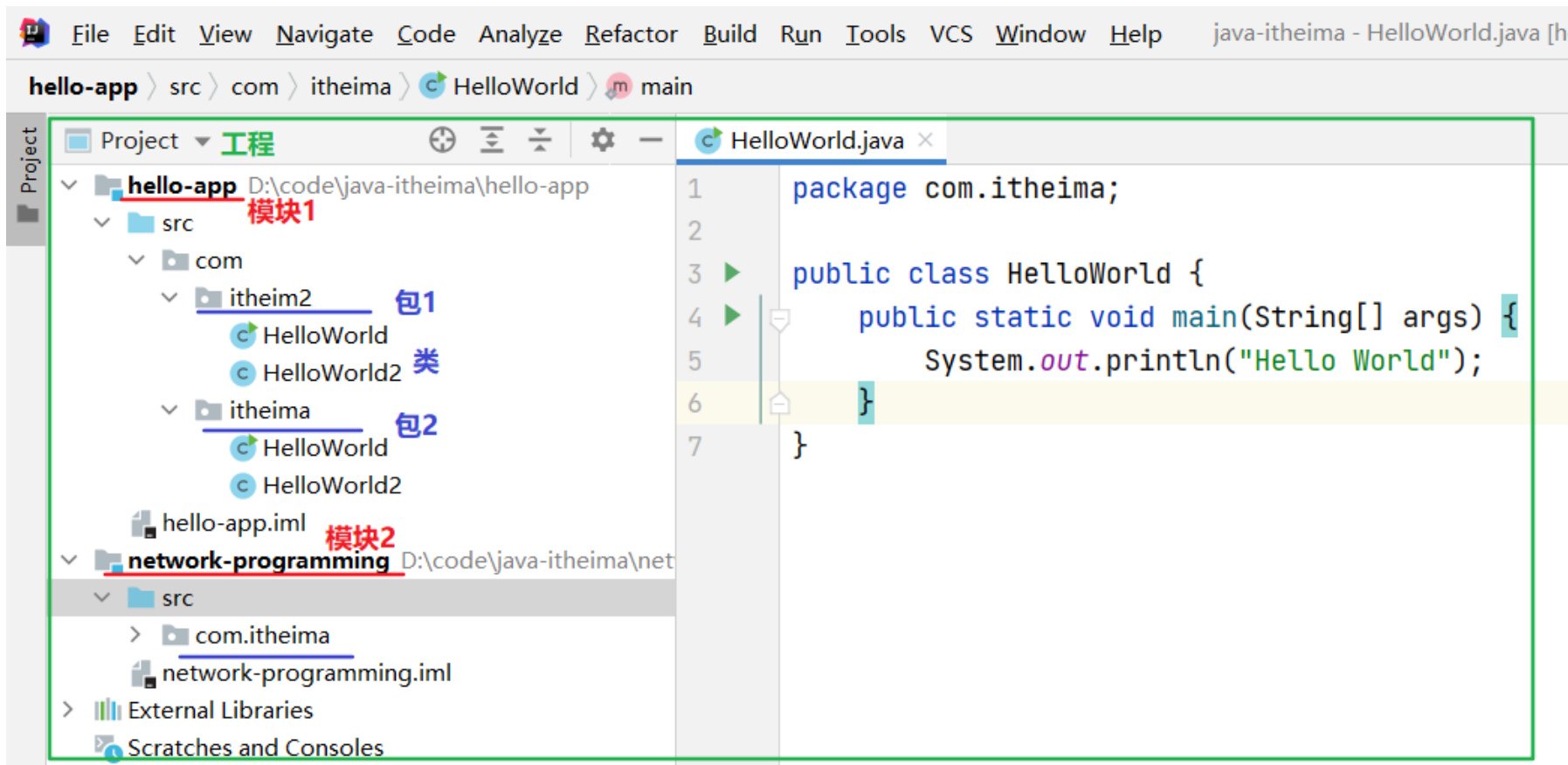


结论：这些结构的划分，是为了方便管理项目技术和代码文件的



使用步骤：

- ① 创建Project空工程
- ② 创建Module模块
- ③ 创建Package包
- ④ 创建class 类
- ⑤ 在类中编写代码
- ⑥ 完成编译运行





总结

1. IDEA的结构都是什么样的?

- **project - module - package - class**
- **project中可以创建多个module**
- **module中可以创建多个package**
- **package中可以创建多个class**

2. 新建都是哪些操作

- **New project/module/package/class**

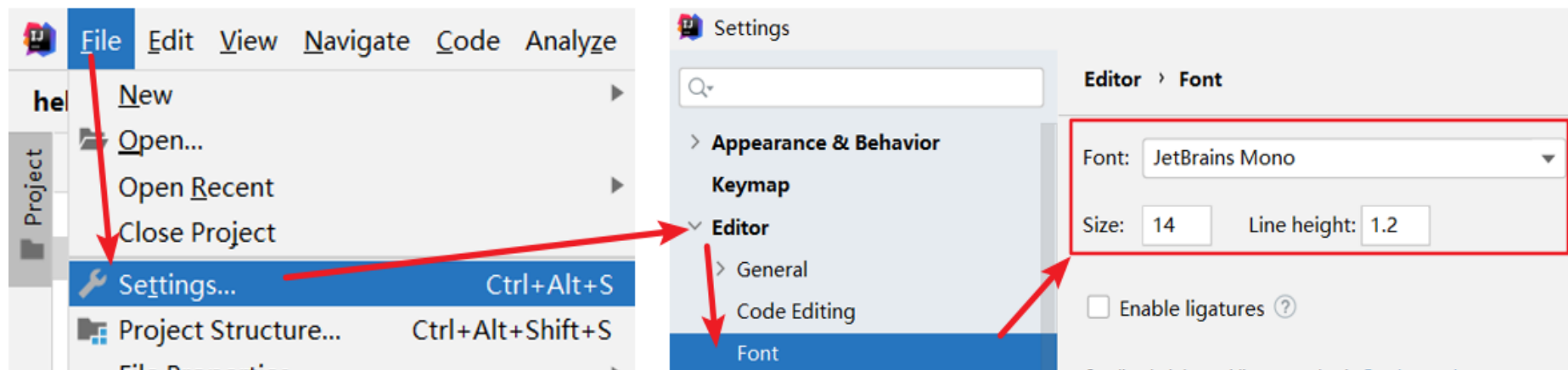
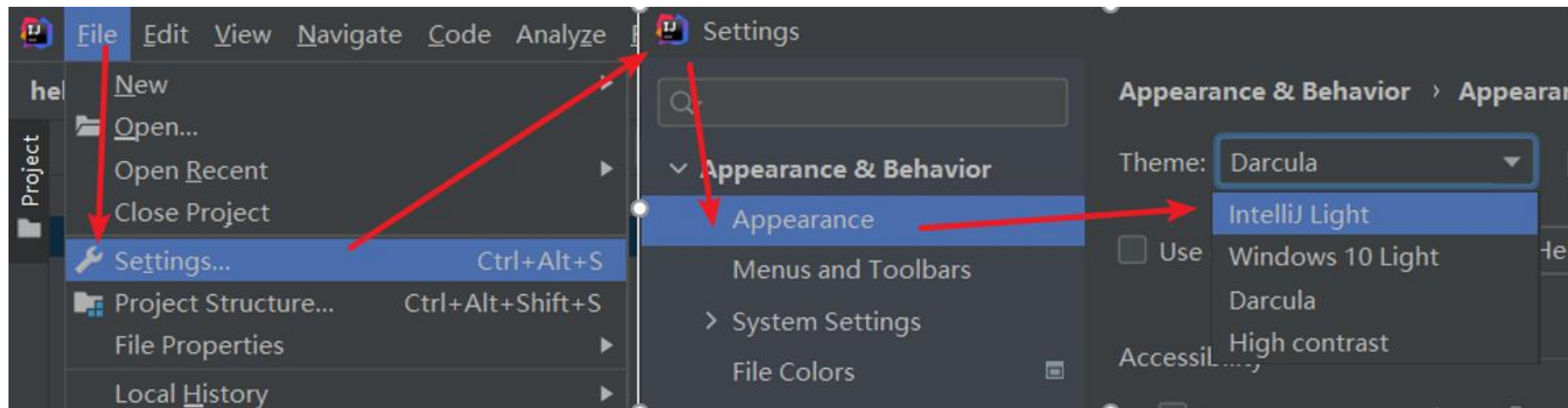


目录

Contents

- **Java入门**
- **IDEA开发工具的使用**
 - ◆ IntelliJ IDEA 开发工具概述、安装
 - ◆ IDEA 编写第一个程序
 - ◆ **IDEA 字体、主题、背景色设置**
 - ◆ IDEA常用快捷键介绍
 - ◆ IDEA工程、模块常用操作
- **Java语言**

IDEA 中设置主题、字体





目录

Contents

- **Java入门**
- **IDEA开发工具的使用**
 - ◆ IntelliJ IDEA 开发工具概述、安装
 - ◆ IDEA 编写第一个程序
 - ◆ IDEA 字体、主题、背景色设置
 - ◆ **IDEA常用快捷键介绍**
 - ◆ IDEA类、模块、工程常用操作
- **Java语言**

IDEA 常用快捷键

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 + / , Ctrl + Shift + /	对代码进行注释(讲注释的时候再说)



目录

Contents

- **Java入门**
- **IDEA开发工具的使用**
 - ◆ IntelliJ IDEA 开发工具概述、安装
 - ◆ IDEA 编写第一个程序
 - ◆ IDEA 字体、主题、背景色设置
 - ◆ IDEA常用快捷键介绍
 - ◆ **IDEA工程、模块常用操作**
- **Java语言**

IDEA 中其他操作

- 删除类文件
- 修改类名称
- 修改模块
- 导入模块
- 删除模块
- 打开工程
- 关闭工程



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

什么是注释

- 在程序中对代码进行解释说明的文字，方便自己和其他人理解、查看，**不会影响程序的正常执行。**

注释有哪些

- 单行注释

```
// 注释内容，只能写一行
```

- 多行注释

```
/*  
    注释内容1  
    注释内容2  
*/
```

- 文档注释：

一般用在类和方法上的注释、以后会讲解区别。

```
/**  
    注释内容  
    注释内容  
*/
```

快捷键进行注释	功能效果
Ctrl + /	单行注释。
Ctrl + Shift + /	选中代码进行多行注释。



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

字面量

- 告诉程序员，数据在程序中的书写格式。

字面量分类

字面量类型	说明	示例
整数	不带小数的数字	666, -88
小数	带小数的数字	13.14, -5.21
字符	必须使用单引号，有且仅能一个字符	'A', '0', '我'
字符串	必须使用双引号，内容可有可无	"HelloWorld", "黑马程序员"
布尔值	布尔值，表示真假，只有两个值：true, false	true、false
空值	一个特殊的值，空值	值是：null



总结

1. 字符、字符串的字面量的格式要求?

- 字符必须单引号围起来，有且仅能一个字符。
- 字符串必须用双引号围起来。
- 布尔类型只有2个值：false 、 true



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

什么是变量

- 变量就是用来存储一个数据的内存区域（可以理解成盒子），且里面存储的数据可以变化。

变量定义的格式

数据类型 变量名称 = 初始值;

存储的初始数据

取名称，首字母建议小写，有意义。

强制限制盒子中存储数据的形式。例如：int（整数类型）、double(小数类型)

```
double money = 6.0;
```

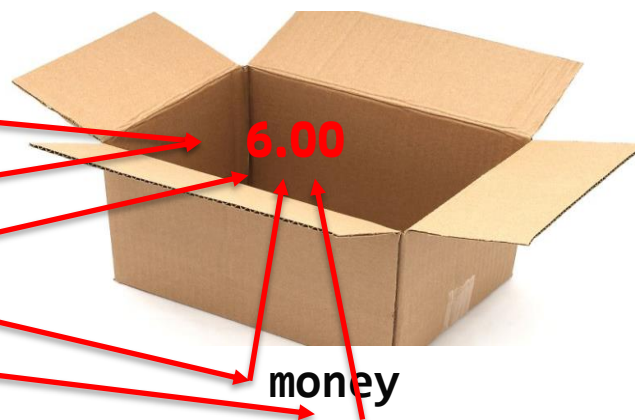


修改变量中存储的值

```
double money = 6.0;  
System.out.println(money); // 6.0
```

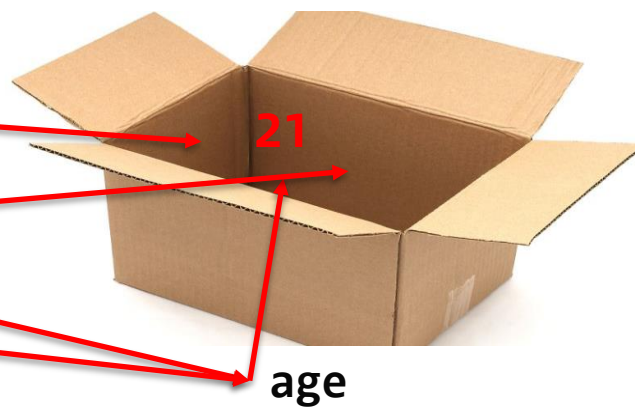
```
money = money + 4.0;  
System.out.println(money); // 10.0
```

+ 4.0 = 10.0



修改变量中存储的值

```
int age = 21;  
System.out.println(age); // 21  
  
age = 25;  
System.out.println(age); // 25
```





总结

1. 变量的作用?

- 存储一个数据的，可以改变里面存储的数据。

2. 变量的格式

- 数据类型 变量名称 = 初始值;



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

定义变量需要注意的几个问题

- 某个变量存储了其他类型的数据。

```
public static void main(String[] args) {  
    int age = 23;  
  
    double money = "有钱";  
}
```

- 变量的有效范围是从定义开始到 “}” 截止, 同一个范围内部不能定义2个同名的变量。

```
public static void main(String[] args) {  
    int age = 23;  
    System.out.println(age);  
    int age = 21;  
}
```

- 变量定义的时候可以没有初始值，但是使用的时候必须给初始值。

```
public static void main(String[] args) {  
    int age;  
    System.out.println(age);  
}
```



总结

1. 变量定义有哪些注意点?

- **什么类型变量存储什么类型的字面值。**
- 同一个范围变量名不能重复
- 变量在使用时要有初始值
- 变量存在访问范围



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量的注意事项
 - ◆ 数据类型
 - ◆ ASCII编码表
 - ◆ 关键字、标志符

数据类型的作用

- 数据类型就是约束变量存储数据的形式。

数据类型 变量名称 = 初始值;

```
public class HelloWorld {  
    public static void main(String[] args) {  
        double money = 9.9;  
    }  
}
```

```
int age = 23;  
}
```



数据类型的分类

- 引用数据类型（除基本数据类型之外的，如String，其他的后面学习）
- 基本数据类型：4大类8种。

数据类型	关键字	取值范围	内存占用（字节数）
整数	byte	-128~127	1
	short	-32768~32767	2
	int(默认)	-2147483648~2147483647 (10位数)	4
	long	-9223372036854775808 ~ 9223372036854775807 (19位数)	8
浮点数	float	1.401298e-45到3.402823e+38	4
	double (默认)	4.9000000e-324 到1.797693e+308	8
字符	char	0-65535	2
布尔	boolean	true, false	1

数据类型定义变量的几个补充知识点

- 随便写一个整数字面值，默认是int类型的，如果希望随便写一个整数默认是long型的必须在数据后加L或者l表示。
- 随便写一个小数字面值，默认是double类型的，如果希望这个小数是float类型的，必须在数据后加F或者f表示。



总结

1. 数据类型分为几种?

- 引用数据类型:String
- 基本数据类型: 4大类8种。
 - ✓ byte short **int (默认)** long 整型
 - ✓ float **double (默认)** 浮点型
 - ✓ char 字符型
 - ✓ boolean 布尔型

2. 随便写的整数、小数字面值默认什么类型?

- 23 , 默认是int类型 , 加上L/l就是long类型的数据了。
- 23.8, 默认是double类型, 加上F/f就是float类型了。



目录

Contents

- Java入门
- IDEA开发工具的使用
- Java语言
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ 关键字、标志符

ASCII (American Standard Code for Information Interchange)

- 美国信息交换标准编码，就是一套现代英语、数字、和其他西欧字符（不含中文）的编号规则。
- 字符在计算机底层就是存储其编号的二进制形式。

字符	十进制	二进制
A	65	0100 0001
B	66	0100 0010
a	97	0110 0001
b	98	0110 0010
0	48	0011 0000



```
char ch = 'a';
```



总结

1. 字符在计算机底层是怎么存储的?
 - 存储的是二进制的形式。
2. ASCII编码是什么?
 - 对如英文字符进行编号的规则，规定英文等字符在底层存储的二进制形式。



目录

Contents

- **Java入门**
- **IDEA开发工具的使用**
- **Java语言**
 - ◆ 注释
 - ◆ 字面量
 - ◆ 变量
 - ◆ 变量使用注意事项
 - ◆ 数据类型
 - ◆ ASCII编码
 - ◆ **关键字、标志符**

关键字

- Java自己保留的一些单词，作为特殊功能的，例如：public、class、byte、short、int、long、double...
- 我们不能用来作为类名或者是变量名称，否则报错。
- 注意：关键字很多，不用刻意去记。

abstract	assert	boolean	break	byte
case	catch	char	class	const
continue	default	do	double	else
enum	extends	final	finally	float
for	goto	if	implements	import
instanceof	int	interface	long	native
new	package	private	protected	public
return	strictfp	short	static	super
switch	synchronized	this	throw	throws
transient	try	void	volatile	while

标识符

- 标识符就是由一些字符、符号组合起来的名称，用于给类，方法，变量等起名字的规矩。

标识符的要求

- 基本要求：由数字、字母、下划线(_)和美元符(\$)等组成
- 强制要求：不能以数字开头、不能是关键字、区分大小写

bj	b2	2b	class	\$2
_2b	#itheima	ak47	Class	HelloWorld



命名指导规范

- 变量名称：满足标识符规则，建议全英文、有意义、首字母小写，满足“驼峰模式”，例如：int studyNumber = 59。
- 类名称：满足标识符规则，建议全英文、有意义、首字母大写，满足“驼峰模式”，例如：HelloWorld.java。



总结

1. 关键字的特点

- Java自己保留的，我们不能用来取名字。

2. 标识符

- 取名字的规则，数字，字母，下划线，美元符等组成。
- 数字不能开头，不能是关键字。



传智教育旗下高端IT教育品牌