

1.1. Java入门

1.1.1. 教学任务

1. 了解计算机组成
2. 掌握常用DOS命令
3. 掌握进制及二进制与十进制转换
4. 掌握Java技术体系、Java优势、特性
5. 掌握JDK的安装与配置、JDK与JRE
6. 掌握代码基本构成

1.1.2. 认识计算机

1.1.2.1. 计算机组成

计算机组成包括**硬件**和**软件**两大部分。

1.1.2.1.1. 硬件

硬件是指完成输入、存储、处理、输出功能的具体设备



1. 常见硬件设备：

- 输入设备：键盘、鼠标、摄像头、扫描仪



- 输出设备：显示器、打印机、音响



- CPU(Central Processing Unit), 中央处理单元：主要厂商有Intel、AMD、高通



- 存储器：主要包括内存、硬盘、优盘



- 其他设备：主板、机箱等



2. CPU主要性能指标：

主频：CPU内核工作的时钟频率，也就是CPU工作频率，单位：Hz，赫兹

内核：2核 4核 8核

例如：Intel 酷睿双核 i5 2.4G、i7 2.6G

3. 存储单位

计算机存储器以位和字节计量：

- 8bit —— 1 Byte
- 1024 B —— 1 KB
- 1024 KB —— 1 MB
- 1024 MB —— 1 GB
- 1024GB —— 1 TB
- 1024TB -----1 PB

1.1.2.1.2. 软件

软件有两大类：系统软件和应用软件

系统软件：控制和支撑计算机系统，例如：DOS、Windows7、Windows10、Mac、Linux、Android、IOS

应用软件：通常指一些应用软件，例如：Office、PhotoShop、WinRAR等

1.1.2.2. 常用DOS命令

进入命令提示符: windows键+R ---》打开运行窗口--->输入cmd

- cd 命令：改变当前目录（change directory）

```
cd [盘符:][目录名][子目录名]
```

正斜杠 / 和反斜杠 \：

- bat:
 - / 是表示参数，\ 是表示根路径或者路径分割符。
- shell
 - / 表示路径，\ 表示转义，“-”和“-”表示参数

目录中的特殊符号：

- ⋮：上一级目录，可以在路径组合中使用
- ⋮：当前目录，可以在路径组合中使用
- \：根目录，可以在路径组合中使用

注意：如果需要访问非当前根目录下的目录文件，应该先输入该盘符，敲回车，然后再用cd 命令

示例：

```
C:\Users\Administrator>f:
```

```
F:\test>cd .\test1
```

```
F:\test\test1>cd ..
```

```
F:\test>cd test1
```

```
F:\test\test1>cd \
```

```
F:\>cd F:\test\test1
```

```
F:\test\test1>
```

- dir命令：列出当前目录下的文件以及文件夹（directory）

```
dir [盘符:][目录路径] [/p] [/s] [/w] [/o]
```

`/p`：分页显示下一页内容，按任意键查看下一屏 `/s`：显示所有目录及其子目录下所有文件 `/w`：宽屏显示 `/o`：分类顺序显示

- md命令：创建目录（make directory）

```
md [盘符:][路径名]<文件夹名>
```

示例：

```
F:\>md test
```

```
F:\>md test\test1\test2
```

```
F:\>md e:\test1
```

```
F:\>md .\test4
```

```
F:\>
```

- copy命令：复制一个或一组文件到指定磁盘或目录中

```
copy <源文件路径> [目标路径]
```

`/s`：复制目录和子目录，除了空的。 `/e`：复制目录和子目录，包括空的。

示例：

```
copy a.txt d:\myFile 复制a.txt文件到d:\myFile文件夹中
copy myFile myPhoto 只 复制myFile文件夹下的所有文件到myPhoto中,不包括子文件夹中的文件
如果要复制所有子文件夹下文件可使用:
xcopy myFile myPhoto /s 不能复制空目录
xcopy myFile myPhoto /e 复制空目录
```

- rd命令: 删除空目录 (remove directory)

```
rd [盘符:][路径名]<子目录名>
```

- 不加参数只能删除空目录, 不能删除当前目录
- 如果目录非空, 则可以加入 /s 删除, 会要求确认, 如果想省略确认, 可以加参数 /q

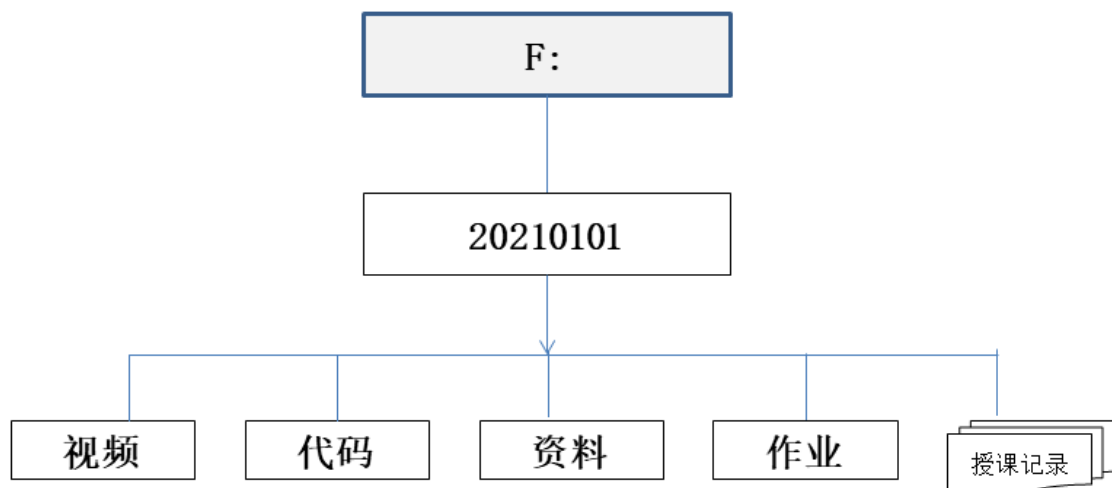
- del命令: 删除**文件** (delete)

删除目录中的一个或一组文件,不能删除目录

```
del info.txt 删除info.txt文件
del myFile\* 删除myFile目录下所有的文件
```

- cls命令: 清空屏幕 (clear screen)
- exit命令: 退出dos命令行
- mspaint: 打开画图板
- notepad: 打开记事本

随堂练习:



```
C:\Users\Administrator>f:

F:\>md 20210101

F:\>cd 20210101

F:\20210101>md 视频

F:\20210101>md 代码

F:\20210101>md 资料

F:\20210101>md 作业
//新建文件授课记录.txt, 并且输入内容-----20210101-----
F:\20210101>echo -----20210101----->授课记录.txt

F:\20210101>
```

F:\>echo abc > a.txt: 写abc到文件a.txt, 如果a.txt不存在, 则创建

F:\>echo > a.txt: 如果a.txt不存在, 则创建

1.1.2.3. 常见快捷键

- ctrl+c 复制
- ctrl+v 粘贴
- ctrl+s 保存
- ctrl+x 剪切
- ctrl+a 全选
- ctrl+z 撤销修改
- delete (shift+delete)
- F2 重命名
- windows+d 回到桌面
- windows+e 打开我的电脑
- windows+L 注销
- alt+tab 切换打开的程序

1.1.2.4. 进制

1.1.2.4.1. 二进制

计算机中的数据都以二进制数字保存。二进制：逢二进一。即只有0、1两个值。如：十进制的10在计算机内保存为二进制的1010 计算机中信息的存储单位位 (bit)：表示一个二进制数码0或1，是计算机存储处理信息的最基本的单位。

1.1.2.4.2. 进制转换

1. 二进制转十进制 要从右到左用二进制的每个数去乘以2的权次方相加
2. 十进制转二进制 除2取余，逆序排列

1.1.2.4.3. 八进制和十六进制

八进制：有八个基本数字：0-7，逢八进一

十六进制：有十六个基本数字：0-9，A-F，逢十六进一

二进制、八进制、十六进制之间的转换

规律1：八进制的八个基本数字对应二进制的3位

规律2：十六进制的十六个基本数字对应二进制的4位

1.1.3. 认识Java

1.1.3.1. Java是什么

Java是一门高级计算机编程语言。

- 高级：相对于机器码和汇编而言，高级语言更加偏向于自然语言
- 编程：把计算机指令组织成一组完成特定任务的序列
- 语言：和计算机沟通

1.1.3.2. Java的地位

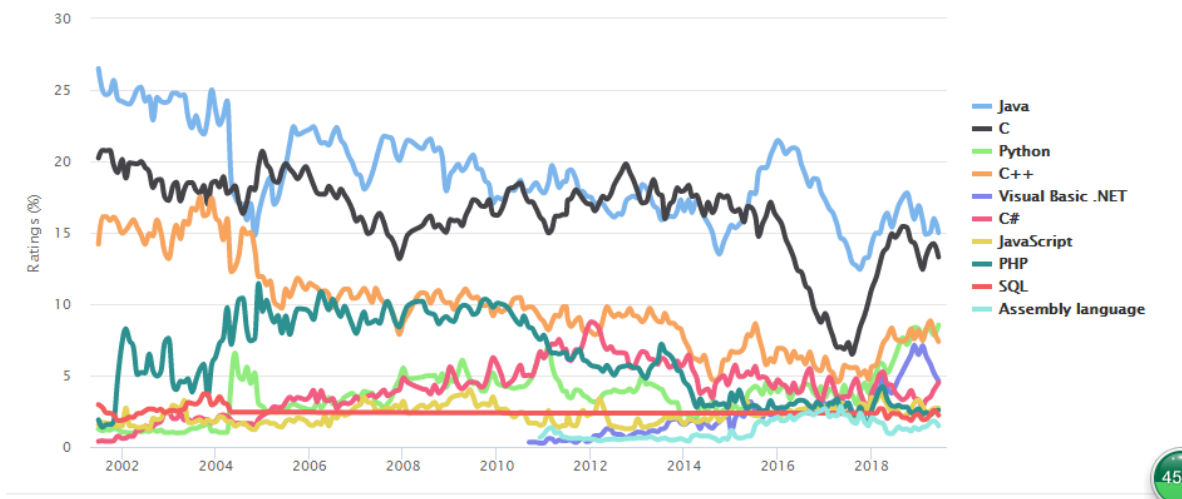
TIOBE排行榜：<https://www.tiobe.com/tiobe-index/>

TIOBE排行榜是根据从业人员、课程和第三方厂商的数量排的，数据来源于搜索引擎、百科等权威网站，可以反映编程语言的热门程度。每月更新一次

1. 编程语言热度排行榜

Jun 2019	Jun 2018	Change	Programming Language	Ratings	Change
1	1		Java	15.004%	-0.36%
2	2		C	13.300%	-1.64%
3	4	▲	Python	8.530%	+2.77%
4	3	▼	C++	7.384%	-0.95%
5	6	▲	Visual Basic .NET	4.624%	+0.86%
6	5	▼	C#	4.483%	+0.17%
7	8	▲	JavaScript	2.716%	+0.22%
8	7	▼	PHP	2.567%	-0.31%
9	9		SQL	2.224%	-0.12%
10	16	▲	Assembly language	1.479%	+0.56%

2. 编程语言指数走势



3. 编程语言热度排行榜长期走势

Programming Language	2019	2014	2009	2004	1999	1994	1989
Java	1	2	1	1	12	-	-
C	2	1	2	2	1	1	1
C++	3	4	3	3	2	2	2
Python	4	7	5	8	24	21	-
Visual Basic .NET	5	10	-	-	-	-	-
C#	6	5	6	6	20	-	-
JavaScript	7	8	8	7	17	-	-
PHP	8	6	4	5	-	-	-
SQL	9	-	-	82	-	-	-
Objective-C	10	3	34	38	-	-	-
COBOL	25	21	18	11	3	8	12
Lisp	32	13	20	13	10	5	3
Pascal	218	16	14	89	7	3	20

1.1.3.3. Java的发展

计算机程序：

为了让计算机执行某些操作或解决某个问题而编写的一系列有序指令的集合

编写程序的工具就是计算机语言，Java就是多种语言中的一种

软件：

软件是计算机程序、程序所用的数据以及有关文档资料的集合。



Java之父: [詹姆斯·高斯林]

1995年 JDK1.0, Sun (Stanford University Network) 公司推出

1998年 JDK1.2, 更名为Java2 包括三个版本 J2ME J2SE J2EE

2000年 JDK1.3, JDK1.4

2004年 JDK1.5, 更名为JDK 5.0

2005年 JDK6.0 J2EE更名为JavaEE, J2SE更名为JavaSE, J2ME更名为JavaME

2009年 Oracle (甲骨文) 74亿美元收购Sun。取得Java的版权。

2011年 Java7.0 内部版本1.7

2014年3月19日, java8.0 内部版本1.8

2017年9月22日, java9.0发布

2018年3月21日, java10.0发布

2018年9月26日, java11发布

1.1.3.4. Java平台

为了应对不同的使用场景, Java的技术体系分为三块:

1. Java SE (Java Platform Standard Edition) 。Java SE 以前称为 J2SE。它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。
2. Java EE (Java Platform Enterprise Edition) 。这个版本以前称为 J2EE。开发企业级的应用
3. Java ME (Java Platform Micro Edition) 。这个版本以前称为 J2ME。Java ME 为在移动设备和嵌入式设备 (比如手机、PDA、电视机顶盒和打印机) 上运行的应用程序提供一个健壮且灵活的环境。

1.1.3.5. Java使用场景

- 1.开发桌面应用程序: 银行软件、商场结算软件
- 2.开发面向Internet的web应用程序 中国移动等门户网站、阿里巴巴等电子商务网站
- 3.提供各行业的解决方案 金融、电信、电力 医疗 商务...
- 4.开发android手机应用程序
- 5.大数据开发的基础

1.1.3.6. Java的特性

1. 开源、免费的生态圈

经过多年的发展已经积累了庞大的从业人员、形成了成熟的技术体系，而且Java生态圈的好多好东西是免费使用的

JavaSE、JavaME、JavaEE、Spring、SpringMVC、Spring Boot、MyBatis、MySQL、Eclipse。。。。。

2. 跨平台

众多平台上都可以允许，操作系统层面可以做到“编写一次、到处运行”

- 硬件平台，微型设备、PC、Server都可以
- 操作系统：Linux、Solaris（Unix衍生版本）、Windows、Mac OS X

跨平台可以大幅减少开发、测试、运维成本，可以大幅缩减开发周期。

1.1.3.7. 跨平台原理

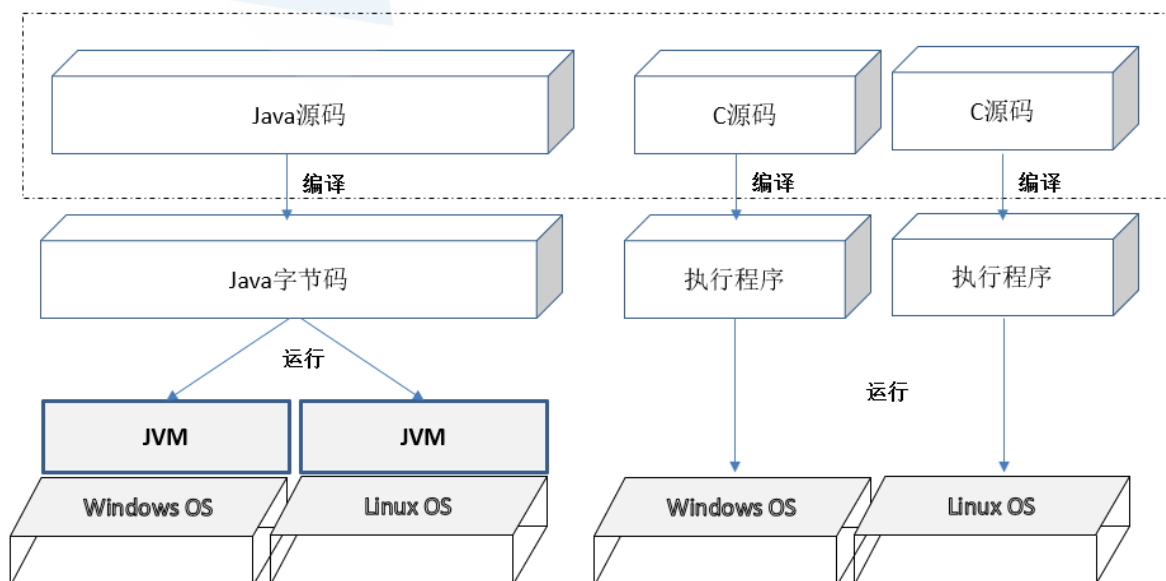
计算机的高级编程语言类型：

- **编译型** C、C++、Java，会生成二进制文件比如：.exe文件、字节码文件
- **解释型** JavaScript、basic 不会生成二进制文件

而Java 语言是两种类型的结合；**先编译,再解释执行**

Java与C的不同：

- Java的平台差异性不需要程序员去处理，由对应的JVM(Java Virtual Machine)来实现，**同样的字节码由不同的VM运行**。JVM大多数情况下就是一个特殊的程序。
- C的平台差异性在编译阶段完成，甚至有些部分需要在源码中处理(标准的系统调用是通用的，特殊的系统调用需要在源码中分别处理)，这样会导致开发、编译、测试的工作量增加



1.1.4. Java开发环境

使用Java语言编写好一个程序后，我们最起码需要编译为字节码，然后由虚拟机执行字节码。编译工具从何而来？

JVM在哪里？需要安装JDK(Java Development Kits)：Java开发工具集

1.1.4.1. JDK的下载

JDK下载：

下载页面：<https://www.oracle.com/technetwork/java/javase/archive-139210.html>

选择JavaSE8-->jdk-8u202-windows-x64.exe(32位, jdk-8u202-windows-i586.exe)

为什么不选最新的版本？

企业技术的迭代需要过程，而且最新的版本有新特性，但是肯定也有未知的坑，大家都想走别人试过的路。目前使用最多的还是Java8。

在实际工作中选择版本的时候还要去考虑周边环境，如：生产环境的JDK是否由我们部署？使用的第三方代码是哪个编译版本？我们的程序是否要交由第三方集成？

JetBrains 开发人员生态系统调查：网站右上角可以调整文章的语言，默认English，可以调为中文

<https://www.jetbrains.com/lp/devecosystem-2020/>

<https://www.jetbrains.com/lp/devecosystem-2020/java/> 可以查看使用最多的Java版本

<https://www.jetbrains.com/lp/devecosystem-2019/>

<https://www.jetbrains.com/lp/devecosystem-2019/java/>

stackoverflow 开发人员生态系统调查：可读性比较差

<https://insights.stackoverflow.com/survey/2019/>

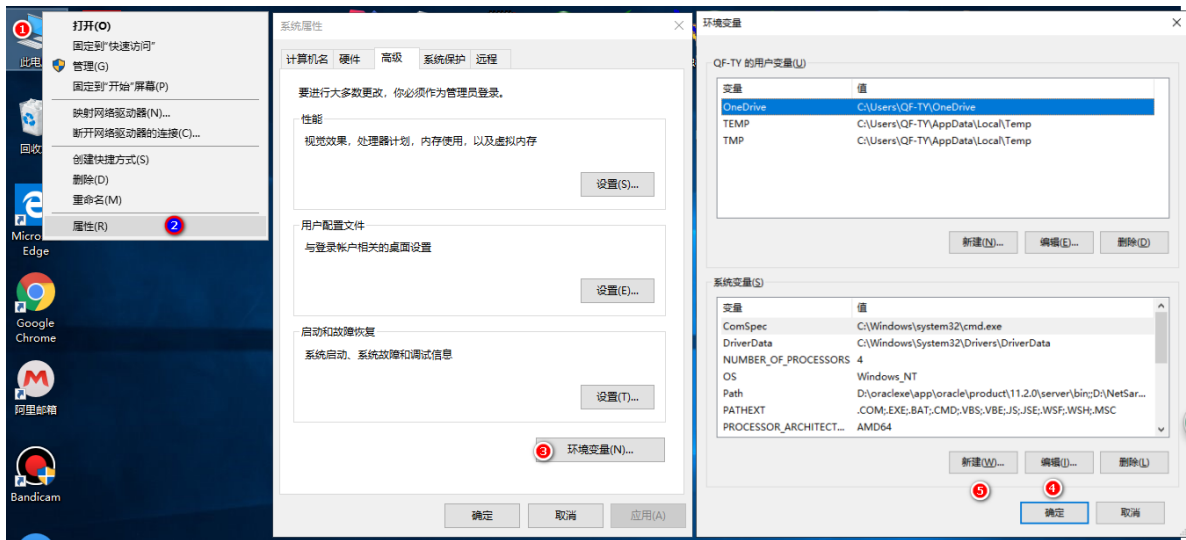
1.1.4.2. JDK的安装

1. 安装过程

。。。下一步

2. 配置环境变量

我的电脑右键菜单-->属性>高级系统设置-->环境变量



- 新建环境变量JAVA_HOME，便于通过脚本启动Java程序时找到JDK安装在哪里，可以不编辑
新建方法：新建环境变量JAVA_HOME，值是JDK安装根路径，如：D:\Java\jdk1.8.0_172\
- 编辑环境变量PATH，便于cmd启动JDK中的工具，如：java、javac，可以不编辑，但是每次使用jdk中的工具都需要全路径，或者cd到Java安装路径下bin目录
编辑方法：编辑环境变量PATH，在PATH的值后面追加Java安装路径下的bin目录即可，如：
D:\Java\jdk1.8.0_172\bin，注意使用分号分割前面的值。当然可以使用%JAVA_HOME%\bin，
%JAVA_HOME%表示使用%JAVA_HOME%环境变量的值。

注意：为了便于引用class、jar文件，JDK5.0以前还会新建环境变量classpath，但是从JDK5.0以后默认就会到当前工作目录以及JDK的lib目录中查找

3. 验证安装成功否

- 查看jdk的版本信息：

不配环境变量也可以直接运行java，是因为安装程序自动编辑了path环境变量，但是指向的不是jdk安装目录下的bin，而是系统盘中的一个目录，该目录下没有javac

```
java -version
```

- 运行javac验证path环境变量配置是否生效：

```
javac
```

上机练习：

1. 安装JDK
2. 配置环境变量

1.1.5. 认识Java程序

1.1.5.1. Hello World

编写第一个Java程序，理解Java的编程过程以及JDK的作用。

第一步：编写源程序

Java源码就是纯文本文件，所以常见的纯文本编辑工具都可以编辑Java源码文件，如：记事本、EditPlus或者是 notepad++等。

使用记事本新建一个文件保存为HelloWorld.java,以下是文件中的内容（区分大小写）：

```
// HelloWorld类名和文件名相同
public class HelloWorld {
    // 公开的 static 静态的 void 没有返回值 String[] args 参数
    public static void main(String[] args){
        //向外输出 打印
        System.out.println("Hello world!");
    }
}
```

- Java源程序(文件): 是一个后缀面是.java的文件，里面写的是java代码
- 解释说明：
 1. public 关键字 公开的
 2. class 关键字 表示定义一个类
 3. HelloWorld 类名 和文件名一致
 4. static 静态的,先记住意思，后面详细讲
 5. void 表示无返回值，先记住意思，后面讲
 6. main 主函数（方法），程序入口函数
 7. String[] 字符串数组类型
 8. args 方法参数
 9. 9 System.out.println("Hello World!");输出Hello World

第二步：编译(Compiler)

打开CMD，cd到HelloWorld.java的保存路径，然后使用javac编译源码

```
javac HelloWorld.java
```

如果运行出现乱码，查看源文件的字符编码，然后加入编译属性：javac -encoding utf-8 HelloWorld.java

编译后会生成后缀名为class的文件,我们称为**字节码文件**

第三步：运行

在命令行中输入：

```
java HelloWorld
```

随堂练习:

- 1 向控制台输出 “世界你好!!!”
- 2 向控制台输出 姓名, 年龄, 爱好

1.1.5.2. 源文件的组成

1. 类 Class

```
public class HelloWorld{  
  
}
```

1. 如果类不是public修饰的, 类名可以和源文件的名字不同, 但是如果类名是public修饰的, 那源文件的名字必须和类名相同。
2. 一个源文件中可以有多个类。但是至多只有一个用public修饰的类

2. 方法

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

命名的一个代码块

3. 语句

```
System.out.println("Hello world!");
```

语句用以完成业务逻辑, 语句要以;
结束, 此处使用系统预定义方法在控制台窗口打印Hello World!, 其他打印方法如下:

- print() 输出内容没有换行
- println() 输出内容后有换行
- printf() 格式化输出, "%"表示进行格式化输出, "%"之后的内容为格式的定义 (了解并查看帮助文档)
 - %f "f"表示格式化输出浮点数(小数)。
 - %.2f 保留小数点后2位
 - %d "d"表示输出十进制整数
 - %s "s"表示输出字符串
 - %x "x"表示输出16进制形式, X 表示大写

十六进制 基本数字0-9 A B C D E F

规则 逢16进1

```
public class Demo3{
    public static void main(String[] args){
        System.out.printf("%f",2.5);
        System.out.println();
        System.out.printf("%.2f",2.5);
        System.out.println();
        System.out.printf("%d",20);
        System.out.println();
        System.out.printf("%s","helloworld世界你好");
        System.out.println();
        System.out.printf("%x",15);
    }
}
```

4. 注释

注释的作用就是帮助程序员理解程序的作用，编译后再class文件中不存在。共有三种注释: 单行注释、多行注释、文档注释

- 单行注释

```
//适用于注释一行
```

- 多行注释

```
/*多行注释
适用于同时注释很多行
注释内容
注释内容
*/
```

- 文档注释

```
/**
可通过javadoc命令生成API文档，主要放在类、方法、成员变量前
*/
```

1.1.6. JDK与JRE

1.1.6.1. 安装目录认知

- `bin/`: 包含JDK的可执行程序。如: `javac`、`java`、`jar`、`javadoc`等
- `db/`:
- `include/`: C语言头文件，由Java Native Interface和Java Virtual 使用
- `jre/`: 私有jre。

1.1.6.2. JDK与JRE

JRE是Java运行时环境 (Java Runtime Environment)。它是运行已编译Java程序所需的所有内容的集合, 包括Java虚拟机 (JVM), Java类库, java 命令和其他基础结构。但是, 它不能用于创建新的Java程序。

JDK是Java开发工具包, 它拥有JRE所拥有的一切, 还有编译器 (javac) 和其他工具 (如 javadoc 和 jar)。它能够创建和编译Java程序。

简单来说:

JDK = JRE + 开发/调试工具

JRE = JVM + jar (如util, math, lang, awt, swing等) + 运行时库。

JVM = 类加载子系统+垃圾回收子系统+执行引擎。

1.1.6.3. 两个JRE的区别

- 私有JRE: JDK目录下的jre, jdk9就没有这个目录了
 - 包含使用attach API所需的库 (attach.dll)
 - 将与JDK一起卸载
- 公共JRE: 安装JDK时独立选择安装的jre
 - 在浏览器中注册Java Plug-in和java web start
 - 此JRE在系统中注册, 例如作为Java应用程序的运行时, Java控制面板
 - 可以与JDK分开卸载

1.1.7. 关键知识点默写

1. 源文件必须以.java结尾
 2. 一个类里可以有main方法(主方法),即程序的入口,程序要执行的部分 其格式为: public static void main(String args[]){}
 3. 一个源文件里可以有多个class声明的类,但只能有一个class被public修饰
 4. 被public修饰的类的类名必须与源文件名相同
 5. 每个语句必须由";"结尾
 6. 执行程序分为两步:
 - 编译:使用javac命令,将.java的源文件编译成.class字节码文件. 注意:必须包含扩展名.java
 - 运行:使用java命令,虚拟机将字节码文件载入内存,开始执行.注意:必须不包含扩展名.class
 7. Java注释:
 - 单行注释: //注释的内容
 - 多行注释:/*注释的内容,可多行 */
 - doc注释:/** 注释的内容,可多行 */
-

1.1.8. 常见面试题

1. 一个".java"源文件中是否可以包括多个类？有什么限制？
2. Java跨平台的原理？
3. Java编程的过程？

