Java 从0开始

- 高可用
- 高并发
- 高性能

构建工具：Ant, Maven, Jekins

应用服务器：Tomcat, Jetty, Jboss,Websphere, webblogic

Web 开发Spring,myBatis

开发工具：eclipse, Intellij idea

特性和优势

- 简单性
- 面向对象
- 可移植性
- 高性能
- 分布式
- 动态性(反射机制)
- 多线程
- 安全性
- 健壮性

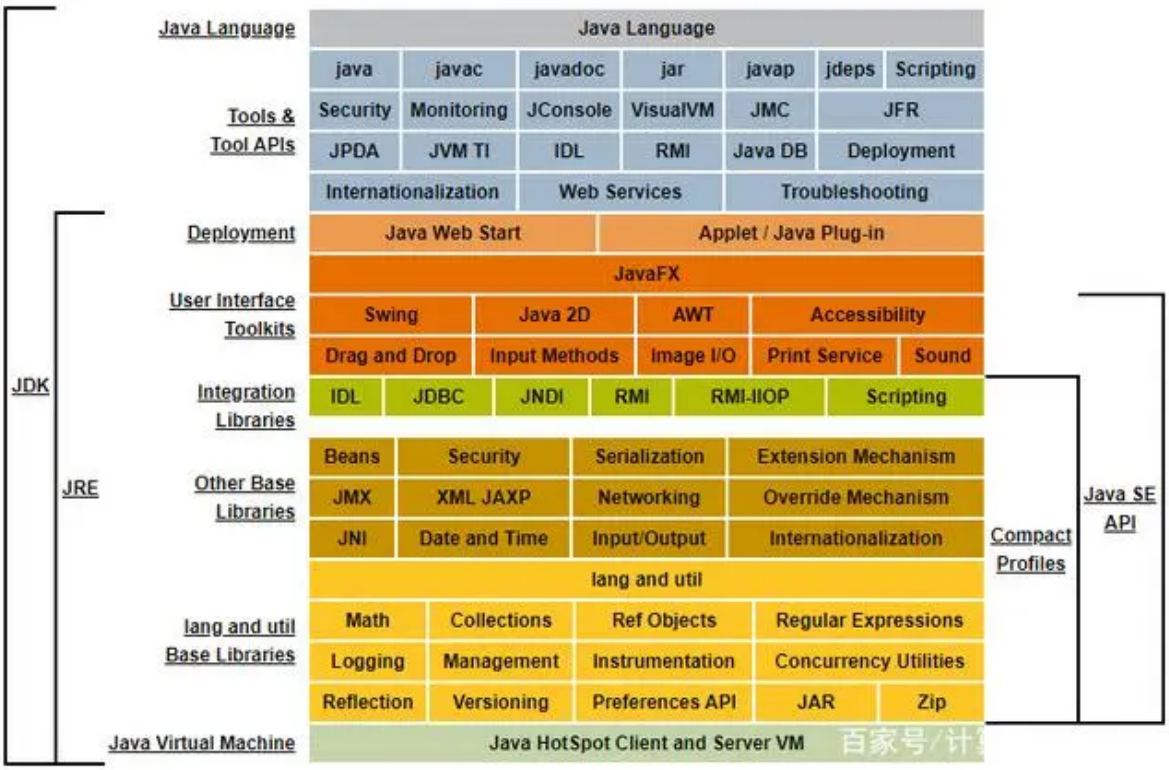
Java三大版本

- JavaSE 标准版 桌面程序 控制台
- JavaME：嵌入式开发
- javaEE：Web开发，服务器开发

JDK, JRE, JVM

- JDK:Java Development Kit
- JRE: java runtime environment
- JV: JAVA Virtrual Machine

> 参考：JDK、JRE、JVM，是什么关系？



关于 JDK、JRE、JVM 之间是什么关系，在 Java 平台标准中已经明确定义了。

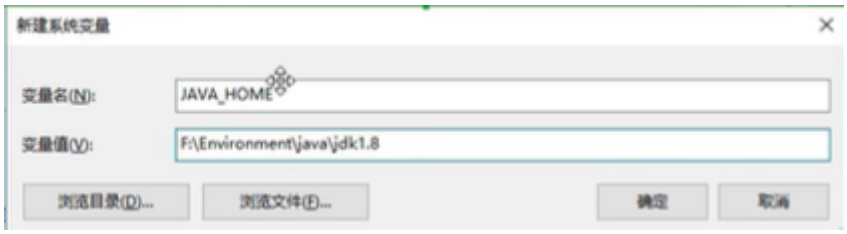
- Oracle 有两个 Java 平台标准的产品，Java SE 开发工具包(JDK) 和 Java SE 运行时环境(JRE)。
- JDK(Java Development Kit Java开发工具包)，JDK是提供给Java开发人员使用的，其中包含了java的开发工具，也包括了JRE。所以安装了JDK，就不用在单独安装JRE了。其中的开发工具包括编译工具(javac.exe) 打包工具(jar.exe)等。
- JRE(Java Runtime Environment Java运行环境) 是 JDK 的子集，也就是包括 JRE 所有内容，以及开发应用程序所需的编译器和调试器等工具。JRE 提供了库、Java 虚拟机（JVM）和其他组件，用于运行 Java 编程语言、小程序、应用程序。JVM(Java Virtual Machine Java虚拟机)，JVM可以理解为是一个虚拟出来的计算机，具备着计算机的基本运算方式，它主要负责把 Java 程序生成的字节码文件，解释成具体系统平台上的机器指令，让其在各个平台运行。

「综上」，从这段官网的平台标准介绍和概念图可以看出，我们运行程序的 JVM 是已经安装到 JDK 中，只不过可能你开发了很久的代码，也没有注意过。没有注意过的最大原因是，没有开发过一些和 JVM 相关的组件代码

环境搭建

下载，双击无脑安装

配置环境变量，我的电脑—>属性—>环境变量（系统变量）—>JAVA_HOME(JDK安装路径)



配置path 变量： 配置系统变量->path 新建

```
%JAVA_HOME%\bin
%JAVA_HOME%\jre\bin
```

在cmd 中输入`java -version`显示以下内容，显示JAVA版本

```
C:\Users\Binyun>java -version
java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
```

Java/jdk1.8/bin 可运行文件

Java/jdk1.8/jre Java运行环境

Hello world

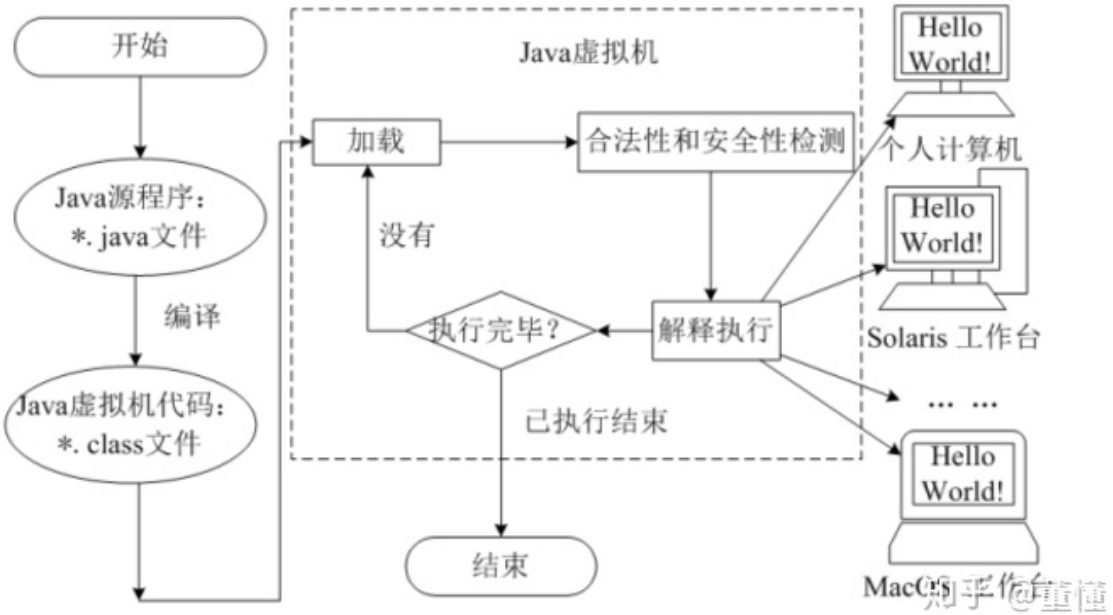
```
public class helloWorld{
    public static void main(String[] args) {
        System.out.print("Hello,World!");
    }
}
```

编译：`javac helloWorld.java` 运行 `javac helloWorld`

输出：hello,world

Java 运行机制

> 参考



1.编写：指的是Java源代码的编写，生成后缀名为 .java的代码文件，该文件可用于编译

2.编译：指的是使用Java编译器对 .java文件进行编译，生成后缀名为 .class的字节码文件，该文件可用于被JVM（java虚拟机）的解释器读取

3.运行：指的是JVM的解释器将编译生成的 .class文件翻译成机器码，并执行程序显示结果

4.字节码文件是一种和任何具体机器环境及操作系统环境无关的中间代码，它是一种二进制文件，是Java源文件由Java编译器编译后生成的目标代码文件。编程人员和计算机都无法直接读懂字节码文件，它必须由专用的Java解释器来解释执行，因此Java是一种在编译基础上进行解释运行的语言。只要计算机上安装了JVM，就可以跨平台跨操作系统运行Java程序。

5.Java虚拟机（JVM）是一种抽象机器，有自己的一套机器指令、栈、寄存器等运行Java程序必备的组件，是附着于具体的操作系统上的软件实现。编译后的Java程序指令不在计算机的CPU上执行，而是在JVM上执行。JVM由多个组件构成，包括可以解释编译后的Java指令的字节码解释器、负责加载类，并完成类的链接和初始化工作的类装载器、按照一定的安全策略对JVM中指令的执行进行控制的安全管理器、用于检测不再使用的对象，进行垃圾回收的垃圾收集器。

集成开发工具—IDEA安装和介绍

> 官网: https://www.jetbrains.com/idea/

下载后默认安装

生成激活码：http://idea.955code.com/

常用的代码简写：

简写	效果
`psvm`	`public static void main(String[] args)`
`sout`	`System.out.println("");`

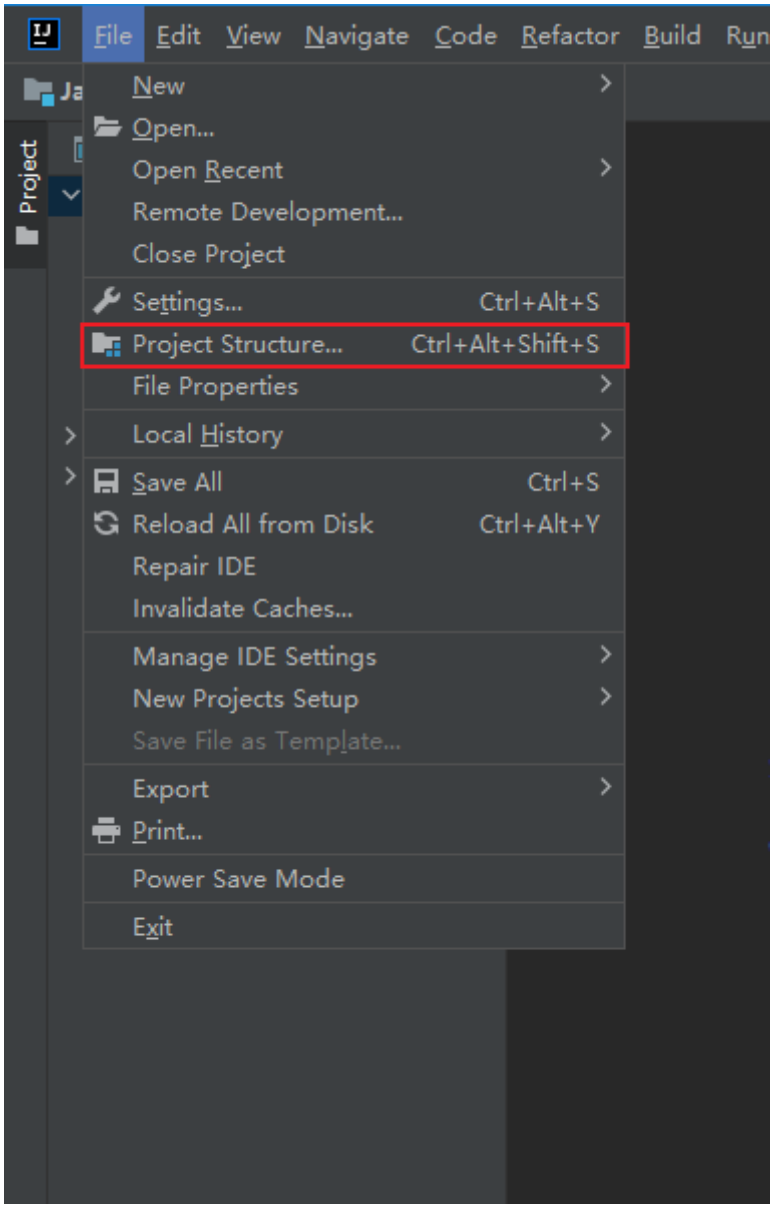
IDEA优化

> https://blog.csdn.net/qq_43192537/article/details/108170139
>
> idea 离线安装translation 谷歌翻译

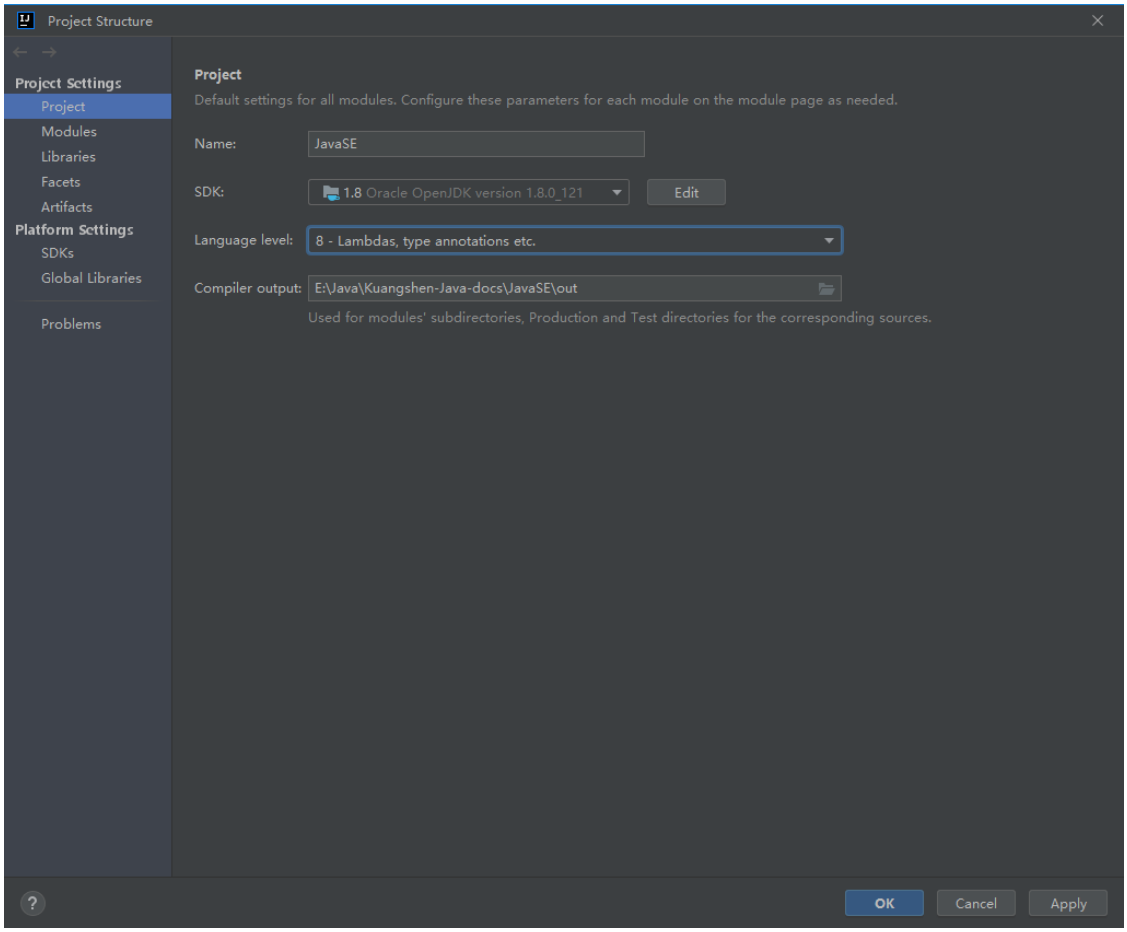
JAVA基础语法

新建项目

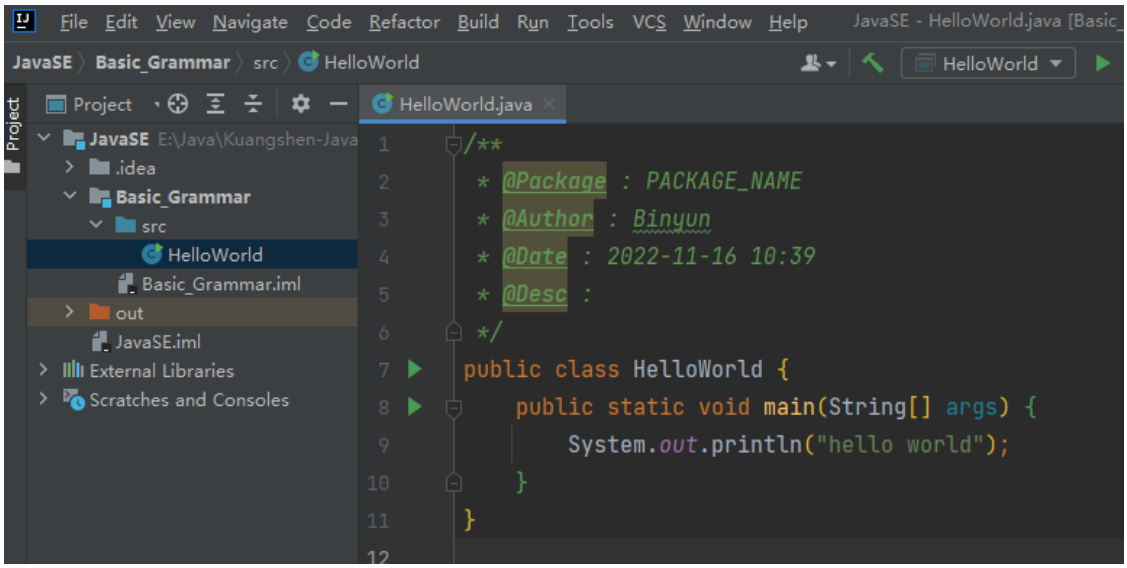
1. file > new project > Empety project> 新建项目名JavaSE
2. new> new module> 新建module名 BaseGrammar
3. file > project structure



4. 配置SDK,并选择语言版本为8

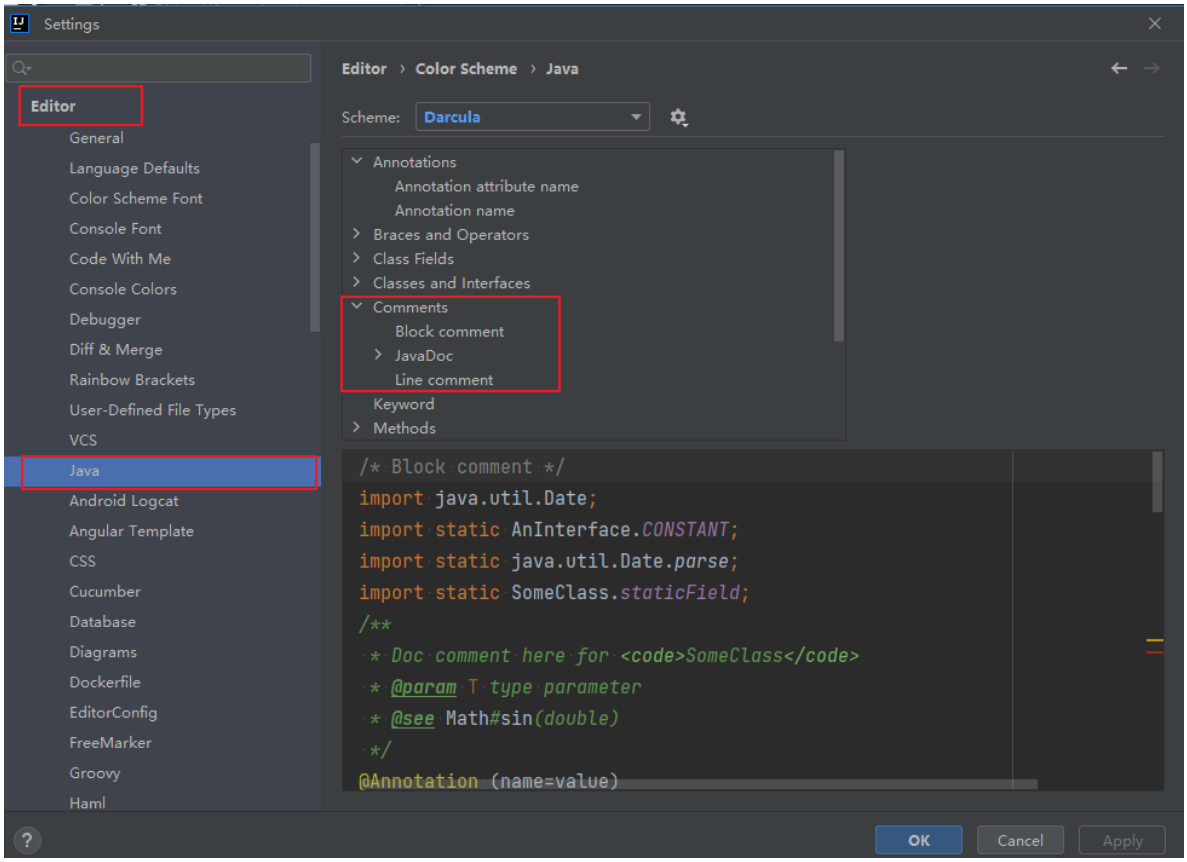


5. 新建HelloWorld.java



成功运行即项目新建成功

代码注释



- 单行注释

  1、第一种 单行注释（ctrl+/ ）
  光标处于当前需要写注释的这一行，在这行任何位置都可以，可以调整的，ctrl+/ 即可实现单行注释，，当想取消时，也可以使用ctrl+/取消行注释

- 多行注释/**/

  多行注释，先选中需要注释的这一行，使用ctrl+shift+/即可实现多行注释，当然，想取消的话，也可以使用ctrl+shift+/

- 文档注释

  在需要注释的位置，输入/**，然后按一下enter即可实现，自动根据参数和返回值生成注释

  /**

  *

  */

关键字

1. 48个关键字：`abstract`、`assert`、`boolean`、`break`、`byte`、`case`、`catch`、`char`、`class`、`continue`、`default`、`do`、`double`、`else`、`enum`、`extends`、`final`、`finally`、`float`、`for`、`if`、`implements`、`import`、`int`、`interface`、`instanceof`、`long`、`native`、`new`、`package`、`private`、`protected`、`public`、`return`、`short`、`static`、`strictfp`、`super`、`switch`、`synchronized`、`this`、`throw`、`throws`、`transient`、`try`、`void`、`volatile`、`while`
2. 2个保留字（现在没用以后可能用到作为关键字）：`goto`、`const`
3. 3个特殊标量`true`、`false`、`null`

标识符

一. 定义：Java对变量，方法，类等要素命名的字符串称为标识符
二. 标识符的规则

1. 由2个大小写的英文字母 0-9 $ _组成
2. 数字不可以开头
3. 不可以使用关键字和保留字，但可以包含关键字和保留字
4. Java严格区分大小写，长度没有限制
5. 标识符不能包含空格

三. 标识符的命名规范

1. 包名：多单词组成的所有的字母都是小写 xxxyyyzzz
2. 类名 接口名多单词组成所有单词的首字母大写XxxYyyZzz
3. 变量名 方法名多单词组成第一个单词的首字母小写，第二个单词开始每个单词的首字母大写 xxxYyyZzz
4. 常量名：所有字母大写多单词用下划线连接XXX_YYY_ZZZ

四. 注意事项

1. 命名规则是必须严格执行的，而规范是可以不遵守
2. 起名字，为了提高阅读性，见名知义

数据类型（Java 是强类型语言，先定义后使用）

> 参考博客
>
> http://t.csdn.cn/emvbf
>
> http://t.csdn.cn/fcsEQ
>
> http://t.csdn.cn/PdBph

			默认值	占用字节
基本数据类型	整型	byte	0	1
		short	0	2
		int	0	4
		long	0	8
	浮点型	float	0.0	4
		double	0.0	8
	字符型	char	\u0000	2
	布尔型	boolean	false	
引用数据类型	数组		null	
	类		null	
	接口		null	

基本数据类型的封装类

数据类型	封装类
boolean（布尔型）	Boolean
byte（字节型）	Byte
char（字符型）	Character
short（短整型）	Short
int（整型）	Integer
long（长整型）	Long
float（浮点型）	Float
double（双精度浮点型）	Double

封装类为各个数据类型提供一些数据的操作方法，可以直接使用；

```
 String a = "21";int b = Integer.parseInt(a);//String 转int。
int a = 21; String b = String.valueOf(a);//int转String。
```