一、基本数据类型：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| byte：字节类型 | short：短整型 | int：整型 | long：长整型 | float：浮点型， | double：双精度浮点型 | char：字符型 | boolean：布尔类型 |

**二、引用数据类型：**

**类（class）、接口类型(interface)、数组类型(arr[])、枚举类型(enum)、注解类型。**

**JDK和JRE和JVM三者之间的关系：**  
JDK是java运行的必要程序工具，在src里有一些封装好的类；

JRE是Java语言运行的环境；

JVM：虚拟机是可供Java跨平台运行在不同系统环境的；

重载：在同一个类里，参数个数不同，类型不同，顺序不同（）

重写：子类必须与父类的方法同名同参

This就是统领上下文，是对当前对象的引用，super就是调用父类方法，构造方法用的，这俩同时只能出现一个。必须在首行。

**注意：**Static静态中不能用this,super ,局部变量不能用static 修饰

**多态**就是事物的多种形态，比如：父类的引用指向子类的对象 fu f=new zi();

**注意**：父类的引用指向子类的对象，不能调用子类的自定义方法，必须用继承的方法。

**抽象类：不可以实例化（不能new），必须重写父类的抽象方法，否则子类也是抽象类**

**Set集合：**无序、不可重复 一般用HashSet，底层是HashMap。

**Properties**是hashtable的子类，不需要泛型，键值对必须都是String.用于读取配置文件。

**接口**中的方法是没有方法体的 public abstract void fuck();而且必须抽象化

**修饰符的权限：public 🡪 protected 🡪 default 🡪 private;**

StringBuilder是可变的字符序列，常用方法：append insert添加，替换是replace

**集合只能存储引用数据类型，数组都行。单列集合：list set**

**迭代器不建议用，知道就行，万物皆可foreach。**

//3.entry的foreach

Set<Entry<Integer, String>> entries =map.entrySet();

for (Entry<Integer, String> entry : entries) {

System.out.println(entry.getKey()+"... "+entry.getValue());

}

**List的特有方法：**添加数据add(int index, Object obj（任意引用数据）)

删除索引位置的数据：remove(index);

修改指定位置的数据set（index，obj）

获取索引位置的数据get(index )

**Map 中的常用方法：**

**增加键值对：put(key value);**

**删除：remove(key); 清空：clear();**

**获取方法：长度：.size()；根据键获取值： get（K key）；**

**判断是否存在：containsKey/Value；**

**HashMap与Hashtable的区别：**

1. Hashtable线程安全,运行速度慢 ; HashMap 线程不安全,运行速度快
2. Hashtable 不支持存储null的键或者值

HashMap 支持存储null的键或者值

关于异常：

**Try –catch**

1. try : 检测的含义,检测写在try大括号中的代码是否具有异常
2. catch : 捕获,捕获try代码块中发生的异常情况
3. try {//进行检查，是否存在异常
4. } catch (Exception e) //捕获异常类型{//异常类型匹配。
5. e.printStackTrace();//打印异常详细信息
6. }

**throw和throws的区别**

1. throw关键字写在方法的内部,throws关键字写在方法上

2. throw 表示异常的抛出 , throws 表示异常的声明

3. throw 每次只能抛出一个异常, throws 声明多个异常

递归：

int i=4;

int res=sun(i);

System.out.println(res);

}

public static int sun(int i) {

if(i==1) {

return 1;

}

return i+sun(i-1);

**关于工具类：**

Arrays 工具类 : 功能 ,针对于数组做一些操作//Arrays.sort(salary);//升排序

Collections 工具类 : 功能,针对于集合进行一些操作, 来自于java.util包,所有方法都是静态方法,所以通过类名.直接调用

//其他的详见API 常见的就是sort + max + min；

例如：

//Collections.sort(list);

**单例模式随便看看就行**

