## java学习笔记

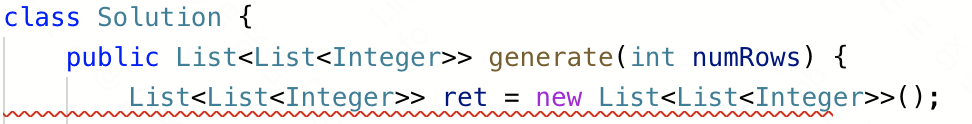
# QA

## 1. Array List 和ArrayList的区别与联系

参考：<https://www.runoob.com/java/java-collections.html>

简而言之：array是普通数组，arrayList是动态数组，List是ArrayList的父类

另外，List是抽象类，不能被实例化



java数组的特点：基本类型数组，存储的是具体的内容。例如，int数组存储的就会一个一个的数值，但是对象数组存储的不是具体的内容而是对象元素的地址。

因此，虽然抽象类不可以被实例化，因为抽象类的定位就不是一个终态的class需要被继承。但是抽象类可以作为数组的元素。因为数组在声明的时候，并不

## 2 什么叫做多态？

## 3 为什么喜欢用接口，直接用实现类代替不好吗？

接口只关注要做什么，而不限制怎么做。从这个意义上来说：基于实现类的代码是基于接口代码的子集，好处是更加易于代码的维护。

## 4.什么叫做依赖倒置？

不要直接去使用子类，而是用父类的引用去调用子类的方法。这样做可以实现多态和解耦。

**多态**：,比如 People 类有一个 Run 方法，Man 和 Woman 这两个类都是继承自 People 的类，并且都重写(override)了 Run 这个方法(男人女人跑起步来不一样)。

现在有一群人一起跑步，有男人有女人。我们可以把这些都装进一个People数组(假设为peoples)。然后:

foreach(People p in peoples) // peoples中对象不同（即有男有女），用于实例化的子类就不同。

{

p.Run(); // 故而，调用的方法也不同，实现了多态

}

由于多态性,在调用 p.Run() 的时候 p 对象本身如果是男人就会自动调用男人的 Run 方法,是女人就会调用女人的 Run 方法。

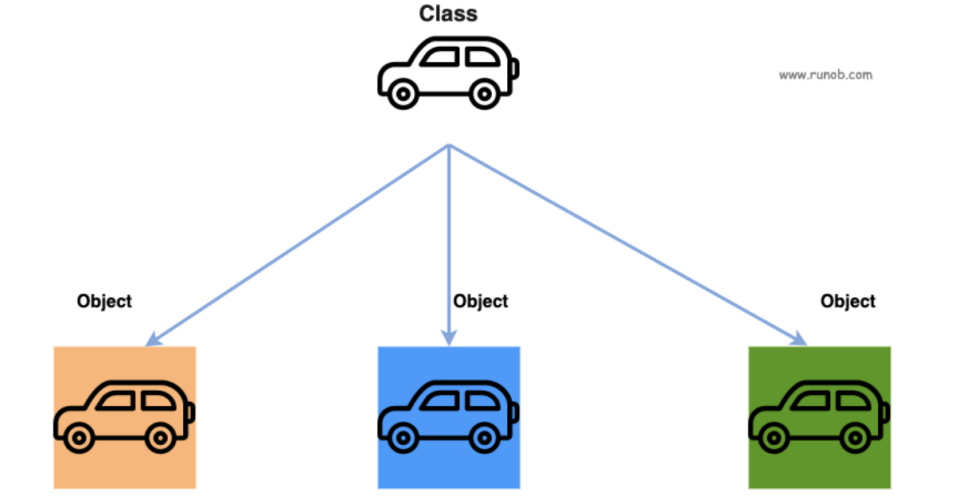
**解耦**：在spring框架下，基于依赖注入，我们可以基于type去扫描实例化的类。这样就算我们把子类换掉了，下游也是无感知的，这样就实现了解耦。

# 一 基础语法

## 1.1 对象和修饰符

Q1：什么叫做对象Object和类class？

class是一个模板，object是class的实例



Q：java中有哪些修饰符，有什么不同特点？

(1) java中有两类修饰符：访问修饰符和非访问修饰符。

访问修饰符：控制类、变量、方法的可被访问范围，如public、private

非访问修饰符：使类、变量、方法收到某种约束/保护，如static、final

(2) Java中有4种访问修饰符，可以作用于类、方法、变量。具体而言，包括：



其中，public和default可以修饰类、接口、方法、变量；protect和private只可以修饰方法和变量[以及内部类]。

被public修饰的类或者方法变量等，可以被所有类访问，如果互相访问的几个类在不同的package中，则需要import操作。如果想让其他包访问，必须声明未public类型，否则default类型就可以满足要求。

(3) 方法的继承规则：

父类方法:public，子类必须public;

父类方法:protected，子类可选public或者protected;

父类方法private，不能够被继承。

Q2:java文件有哪些规则？/源文件声明规则

代码顺序：package声明 -- import语句 -- 类定义

命名：源文件名称 = public类名

Public数量：一个源文件中只能有一个public类，多个非public类

1. 常用的非访问修饰符有static、final

Static：修饰静态变量或者静态方法，静态方法中不能使用this，加入有些方法只想在这个包使用，为不想被import，那么可以不加static。

final：修饰修饰不可更改的变量或者重写的方法

Q3:什么叫做方法的重载?

Java是强类型语言，声明变量必须先指出变量类型。重载指定义一个方法名与已有方法相同，但是参数列表不同的方法的过程。java会根据方法签名决定使用哪一个方法。例如已经定义了一个求int型最大值的方法，再定义一个求float类型最大值的方法就是重载。

Q4：类中不同变量的作用域和生命周期是怎样的?

|  |  |  |  |
| --- | --- | --- | --- |
| **变量类型** | **作用域** | **生命周期** | **其他** |
| 局部变量 | 声明-方法结束 | 方法的周期 | 定义在方法中 |
| 成员变量 | 类中所有方法+特定类的语句块 | 实例的周期 | 定义在类中方法外 |
| 类变量 | ？ | ？ | 必须用static修饰 |

Q5:什么叫命令行参数？

执行程序的时候，给main函数传递的参数。就是main函数args对应的实参。

## 1.2 基础数据类型

Q1：基本数据类型有哪些？

### 基本数据类型汇总

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类型 | 名称 | 位数 | 范围 | 备注 |
| 整形 | byte | 8 | 127 |  |
|  | short | 16 | 32767 |  |
|  | int | 32 | 2147483647 | 21亿 |
|  | long | 64 | 很大 | 十进制19位 |
| 浮点型 | float | 32 | 38次方 |  |
|  | double | 64 | 308次方 |  |
| 布尔型 | boolean | 1 | 2种 |  |
| 字符型 | char | 16 | 0-65535 | 转化为int后 |

### 1.2.1 字符型

Q：有哪些初始化方法？

char a = ‘a’ // 一定要是单引号

Character a = ‘a’ // 装箱为对象

Character a = new Character(‘a’) // 在堆栈区新开辟空间

Q:有哪些转义字符，为什么叫转义字符？



以\t为例，如果只是t的话，name就会输出一个字符‘t’，加上反斜杠就是在文中插入了一个tab键，所以由于\的加入，字符的含义/功能发生了改变，所以叫转义字符。用插入特别形象，因为反斜线本身就像一个漏斗，也像汉语中的插入符号\ /。

Q：java中如何插入引号和斜线呢？

用\\插入一个斜线，为什么呢？哈哈哈哈，因为\本身是用来转义的字符，双重否定等于肯定，\\表示把转义再转回来，当然是输出\本身了。

引号，本身在程序语言中就是定界符，直接出现在文中会造成错误，所以转义后作为普通字符。

所以归纳来讲，转义有两种功能：把普通到字符转为特殊字符，把特殊字符转为普通字符。本身就是普通字符，转义后变成有特殊含义到字符，反之亦然。

Q：字符包装类常用方法

判断是否为数字：isDigit

判断是否为字母：isLetter

判断字母大小写：isUpperCase

转化为字母大小写：toLowerCase

转化为长度为1的字符串：toString

### 1.2.2 数值型

Q：数字类型的初始化方法？

对于基本数据类型，常用的数字类型为int、float、double，其中float类型初始化的时候要加f后缀。

对于包装类型，有两种初始化方式：

**① 公共池初始化**

② 堆初始化[独立性]

Q：什么叫基本类与包装类？

在java中，原始数据类型,没有属性方法,只能进行数学运算，

但是java中有些方法的参数只能是对象，这时候数值变量就无法使用这些方法。

包装类型是基本数据类型的对象化。

例如：Long、Double、Float等

Q：基本类型与包装类型的区别与可比性是怎样的？

**区别：基本数据类型存储的是变量值，包装类型存储的是引用对象的地址。**

关于比较方法：在java中，基本数据类型相等判断只有==，包装类型除了可以使用==，还有类的equals方法，==会比较值和地址，equals只判断值。

另外，java会对-128-127的int型数值进行缓存，引用后包装类存储地址相同，影响②

对于包装类与基本类的比较，不同初始化方式的包装类之间的比较，把握以下原则即可：

①：堆初始化的结果因为地址不一样，所以结果总是不相等

②：常量池初始化，-128-127之间可以认为地址是一样的，所以相等，其他范围不等

③：基本数据类型和包装类型比较时，会拆箱比较，所以相等。

Q：如何使用java的math方法？

Math中的方法都被定义为static类型，可以在函数中直接调用而不需要import。例如：

Math.round()/Math.pow(底数, 指数)等。

Q2：java中包括哪些类型转换，有什么特点？

### 1.2.3 类型转换

#### 1.2.3.1 自动类型转换

整形、实形、字符型可以进行混合运算，不同类型的变量运算时会先转换为同一类型，然后再计算。转换基本遵循以下过程：

byte,short,char—> int —> long—> float —> double

即转换会朝着整形--实形，短--长的方向进行

#### 1.2.3.2 强制类型转换

范式：type a = (type) b

例如：int a = (int) 7L

#### 1.2.3.3 隐含类型转换

整形默认为int，long类型后缀必须加L或则会l，否则报错

小数默认为double，float初始化的时候必须后缀加F或者f，斗则会报错。

Q3：两个short类型的变量相加是什么类型

java中，比int小的混合运算会同一转换成int类型计算。

Q4：java中字符变量的初始化用单引号还是双引号？

char用单引号初始化，String用双引号初始化。

## 1.3 数组

\* 1 定义与初始化可以一起

\* 2 直接sout会返回数组的地址，需要for循环输出

\* 3 中括号可以用来 修改和访问

\* 4 length返回长度

## 1.3对象数据类型

### 1.3.1字符串

Q1:String有哪些初始化方法，不同方法的区别是什么？

公共区初始化：

堆初始化：

Q2：String有哪些常用的方法？

长度:length

拼接:concat

Split:split

大小写:toUpperCase

索引操作:charAt + indexOf

格式化:format

Q：为什么有了String还有StringBuffer和StringBuilder？其区别是什么？

string对象不可修改，String不会有增删改的api。

Q：buffer和builder之间的区别是什么，方法相同吗？

StringBuilder 快，但不是线程安全的（不能同步访问）。

怎么选呢？无脑选builder。其常用方法包括：

增加：insert-开始位置，append-起始位置

删除：delete-起始位置

修改：replace-起始位置，setcharAt-位置

查找：charAt，indexOf

Q3:字符串如何格式化输出？

整形格式化：% [标识] [宽度] 转换格式

标识包括：

|  |  |  |
| --- | --- | --- |
| 标识 | 含义 | 例子 |
| + | 结果总是带有符号 | 负数带符号，正数带正号 |
| 0 | 结果用0来填充 | 003.2 |
| , | 十进制时，每个三位加， | 123,456 |
| ‘ ’ | 正值加空格，负值加符号 |  |
| # | 8进制和16进制前导符 | 0x12 |

浮点格式化：% [标识] [宽度.精度] 转换方式

|  |  |  |
| --- | --- | --- |
| 转换方式 | 含义 | 例子 |
| f | 十进制普通浮点数 | 1.23 |
| e | 科学计数 | 1.2e3 |
| g | 自动选择计数法 |  |
| a | 16进制 |  |

字符串格式化：% [标识] [宽度] 转换方式：例如%-5s / %+5.2f

### 1.3.2枚举

枚举常用方法

.values():返回所有枚举常量

.ordinal():返回枚举常量的序号

.valueOf():返回枚举常量

Jackson ObjectMapper

readValued(): json内容反序列化为java对象

### 1.3.3 集合类型

#### ArrayList

|  |  |  |
| --- | --- | --- |
| 功能 | 方法 | 备注 |
| 增加 | Add | 索引添加+append添加 |
| 删除 | remove | 索引删除+条件删除 |
| 修改 | set |  |
| 查找 | get/contains/indexOf |  |
| 排序 | sort |  |

#### LinkList

|  |  |  |
| --- | --- | --- |
| 功能 | 方法 | 备注 |
| 增加 | Add | 索引添加+append添加 |
| 删除 | remove | 索引删除+删除头部/尾部 |
| 修改 | set |  |
| 查找 | get/contains/indexOf/getFirst/getLast |  |
| 排序 | 无 |  |

#### HashSet

Q：hashset有哪些特点？

① 不允许有重复元素，允许有null

② 无序

③ 非线程安全：多个线程同时修改hashset元素，结果是不确定的

Q：hashset有哪些常用的方法？聚焦于增加删除与查找。

|  |  |  |
| --- | --- | --- |
| 功能 | 方法 | 备注 |
| 增加 | add |  |
| 删除 | remove | 有的话返回true，否则false |
| 查找 | contains |  |

#### Hashmap

Q:hashmap的常用方法有哪些？

|  |  |  |
| --- | --- | --- |
| 功能 | 接口 | 备注 |
| **增加元素** | **put** | **put(k, v)** |
| 删除元素 | **remove** | remove(k) // **会返回value** |
| **修改元素** | **replace** | **replace(key, value)** |
| 查找元素 | get | 基于key查找value |
| 排序 | 无 |  |
| 遍历 | hashmap.foreach(k, v) | 遍历k, v |
| keySet() # keys() | 返回所有的key |
| values() # values() | 返回所有的values |

#### Iterator

Q:Iterator有哪些常用的方法，聚焦于遍历。

|  |  |  |
| --- | --- | --- |
| 功能 | 接口 | 备注 |
| 步进 | **Next** |  |
| 判断是否迭代完毕 | hasNext |  |
| 删除元素 | Remove | 被封装成迭代器的对象对应元素也被删除 |

汇总

ArrayList：add/remove/set/get+contains+indexOf, sort

LinkList： add/remove/set/get+contains+indexOf+getFirst+getLast, 无sort

HashSet: add/remove/没有改/contains，无sort

HashMap：put/remove/replace/get+containsKey+containsValue

Iterator: 无增加/remove/无修改/next

Array的改是基于索引位置来改，链表也是基于索引来修改

所以，数组链表字典的增删改拆接口都是完备的，集合没有修改功能。

迭代器聚焦于遍历，也可以删除

Q：Array中如故有重复，那么如何返回所有的index呢？

Q：哪些对象可以被迭代器处理

## 1.4 分支语句

### 1.4.1 if -else范式

if(){

}else if(){

}else{

}

{}里面是语句块，可以嵌套if、for等。

### 1.4.2 switch-case

变量a

Switch (a){

case value1:

Process1;

Break; // opt

case value2:

Process2;

Break;

default:

Process\_default;

}

## 1.5 循环语句

# 二 面向对象

## 2.1 继承

### 2.1.1