

Java数据类型

无隅

Java数据类型

- 基本类型 (Primitive Types)

- 八种基本类型

- (1) 六种数字类型 (四个整数型, 两个浮点型): byte, short, int, long, float, double,

- (2) 一种字符类型: char

- (3) 一种布尔型: boolean

- 有且只有这八种数据类型, 每种基本类型都有对应的包装类

- e.g.: `int num = 5; double square = 13.4; boolean isUpdated = true;`

- 引用类型 (Reference Type)

- 对象和数组 枚举都是引用类型 Class, Interface, Array, Enum

- e.g. : `Apple a = new Apple();`

- `List<Integer> list = new ArrayList<>();`

在内存中的存储

- 变量
 - 当创建变量的时候，需要在内存中申请空间
 - 内存管理系统根据变量的类型为变量分配存储空间，分配的空间只能用来储存该类型数据。
- 基本类型：存储在栈区
- 引用类型：存储在堆区

变量声明与赋值

- 基本类型：直接存储值
- 引用类型：存储真实对象的地址引用

变量声明与赋值- LHS和RHS

- LHS是指Left-hand Side，而RHS是指Right-hand Side。二者区别就是关于作用域对变量的查询目的是变量赋值还是查询。
- LHS可理解为变量在赋值操作符(=)的左侧，例如 `a = 1`, 当前对变量a查找的目的是变量赋值。
- RHS可以理解为变量在赋值操作符(=)的右侧，例如：`int b = a`, 其中对变量a的查找目的就是查询。

变量声明与赋值 - 赋值

- 基本类型： 将RHS的值直接赋给LHS
- 引用类型： 将RHS对象的地址引用安排给LHS

变量声明与赋值 - 基本类型赋值

e.g.:

```
int a = 3;  
int b = a;  
b = 4;  
System.out.println(a);  
System.out.println(b);
```

结论：对于基本类型， 不论如何给a, b赋值，
或者改变他们中的任意一个， 都
不会
影响到另一个的值

output: a = 3 , b = 4;

变量声明与赋值 - 引用类型赋值

e.g1:

```
MyObject obj1 = new MyObject(5);  
MyObject obj2 = obj1;  
obj2.value = 3;  
System.out.println(obj1.value);  
System.out.println(obj2.value);
```

output:

e.g2:

```
MyObject obj1 = new MyObject(5);  
MyObject obj2 = obj1;  
obj2 = new MyObject();  
obj2.value = 3;  
System.out.println(obj1.value);  
System.out.println(obj2.value);
```

output:

参数传递 - 基本类型

```
1
2 public void add(int a) {
3     a = a + 1;
4     System.out.print(a);
5 }
6
7 public static void main(String[] args) {
8     int a = 5;
9     add(a);
10    System.out.print(a);
11 }
```

结论:

1. 对于基本类型，传入的是基本类型值的拷贝
2. 作为参数的原始值并不会改变

参数传递 - 引用类型

```
1
2 public void add(MyObject obj) {
3     obj.value += 1;
4     System.out.print(obj.value);
5 }
6
7 public static void main() {
8     MyObject obj = new MyObject(5);
9     add(obj);
10    System.out.print(obj.value);
11 }
12
```

结论:

1. 对于引用类型，传入的是引用类型地址的拷贝
2. 更改函数内的对象（参数）将更改原始对象

参数传递 - 引用类型

```
public static void main(String[] args) {  
    MyObject obj = new MyObject(3);  
    increase(obj);  
    System.out.println(obj.value);  
}  
  
public static void increase(MyObject obj) {  
    MyObject obj2 = obj;  
    obj2.value += 1;  
    System.out.println(obj.value);  
    System.out.println(obj2.value);  
}
```

练习

```
1
2  public void change(String str) {
3      str = new String("John");
4      System.out.print(str);
5  }
6
7  public static void main() {
8      String s = new String("Tom");
9      change(s);
10     System.out.print(s);
11 }
```

总结

- java数据类型与在内存中的存储
- 变量的声明与赋值
- 参数传递