

2.1 关键字

用于定义数据类型:

class interface enum byte short
int long float double char
boolean void

用于定义数据类型值的關鍵字

true false null

用于定义流程控制的关键字

if else- switch case default
while do for break continue
return

用于定义访问权限修饰符的关键字

private protected public

用于定义类, 函数, 变量修饰符的关键字

abstract final static synchronized

用于定义类与类之间的关键字

extends implements

用于异常处理

try catch finally throw throws

用于包

package import

其他

native strictfp transient volatile assert

保留字

byValue cast future generic inner operator outer
rest var goto const

2.2 标识符

命名规则:

- > 组成: 字母 0-9 下划线 \$
- > 数字不可开头
- > 不得和关键字和保留字重名
- > 区分大小写, 长度无限制
- > 不能包含空格

命名规范:

包名: 多单词组成时所有字母都小写

类名、接口名: 多单词组成时所有单词首字母大写

变量名、方法名: 多单词组成时, 第一个单词小写, 第二个单词大写

常量名: 所有字母大写, 单词间用下划线连接

2.3 变量

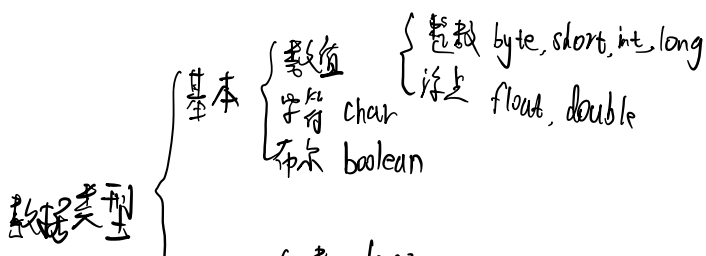
定义: 数据类型 变量名 = 变量的值

int i = 1;

X: Java 中的变量必须先定义, 后使用

变量定义时, 必须同时初始化 (int i; 不正确)

作用域在一个 {} 内有效



引用 { 无
接口 interface
数组 []

整型数据类型范围, 受具体 OS 的影响:

byte	1B	-128 ~ 127
short	2B	相应 大小
int	4B	
long	8B	

定义 long 时, 需要在最后加上 "L" 或 "L"

long i = 1L;

浮点与整型数相同:

float	4B	单精度
double	8B	双精度

定义 float 时, 需要在最后加上 f 或 F

float i = 1f;

字符:

char a = '1';

boolean:

boolean a = true;

boolean b = false;

不可使用 0 和非 0 代替, 不同于 C.

引用类型之 String:

String str = "hello world";

凡是引用类型, 都可以用 null 作为值

基本类型和引用类型的重大区别:

基本类型: 1. 不可变, 2. 不可空

基本类型是变量 (变量类型)

```
int a = 1;
```

```
int b = a;
```

// a和b两个变量使用了两块内存空间

引用类型是标签 (不可变类型):

```
String a = "Hello";
```

```
String b = "Hello";
```

// a和b指向同一内存空间

// 这里和python一样进行优化

// 如果执行 String a = new String("Hello");

// String b = new String("Hello");

// a == b 为 false

```
String s3 = "he" + "llo" + "o";
```

// 使用 + 拼接

自动类型转换:

char

short → int → long → float → double

byte

这些不会相互转换, 他们在相互之间计算时先转为 int.

* 将任何基本类型的值和字符串值进行连接运算时,

基本类型的值将自动转化为字符串类型

例:

```
3.5f + "" // "3.5"
```

```
3 + 4 + "Hello!" // "7Hello"
```

```
"Hello!" + 3 + 4 // "Hello!34"
```

```
'a' + 1 + "Hello" // "q8Hello"
```

```
"Hello!" + 'a' + 1 // "Hello!a1"
```

强转:

int a=1;

byte b=a; ~~X~~

byte b=(byte) a; ✓

boolean 不可转变为其他类型.