# 第2章

### 思考题

1. Java语言的标识符应遵守如下规则：只能由大小写英文字母A~Z或a~z、数字0~9、下画线“\_”、美元符号“$”或人民币符号“￥”组成，开头不能是数字，不能使用其他特殊符号，不允许使用Java关键字，标识符区分大小写。

除上述规定外，标识符的命名还应遵守以下约定：标识符不宜过长，建议使用英文单词及其组合，做到“见名知意”；常量全部用大写字母，变量和方法用小写字母开始，类以大写字母开始。

1. Java的数据类型分为基本数据类型和引用数据类型。

8种基本数据类型包括：字节型（byte）、短整型（short）、整型（int）、长整型（long）、字符型（char）、单精度型（float）、双精度型（double）和布尔型（boolean）。

引用数据类型包括：类（class）、接口（interface）以及数组。

1. Java的分支选择语句包括if-else和switch-case两种。if语句用来判断分支条件，条件为真时执行if语句块，否则执行else语句块，if-else结构可以多层嵌套。switch语句用于多个分支的匹配，当变量值与某个case值相等，则执行该case语句块，若遇到break语句则跳出分支判断，否则继续执行下一个case语句块中的代码，当没有匹配到case语句时可以使用default语句定义默认代码段。
2. Java的循环结构包括while、do-while和for循环。while循环判断条件为真时进入循环体。do-while循环先执行一次循环体，然后再判断循环条件。for循环通过设定循环变量的初值、循环条件和循环变量的变化方式（递增或递减）来确定循环次数。
3. Java的运算符可以分为：算术运算符、赋值运算符、关系运算符、逻辑运算符、位运算符和条件运算符。优先级从高到低的大体顺序依次为：括号、算术运算符、关系运算符、位运算符、逻辑运算符、条件运算符、赋值运算符。
4. while语句会首先判断循环条件，当条件为真时进入循环体，循环体每次执行后都会重新检查循环条件。而do-while语句无论如何都会先执行一次循环体内的语句，然后再检查循环条件，若满足条件则重新执行循环体。从循环次数上看，while循环会执行0次或多次，而do-while循环会至少执行1次。

### 选择题

1. BCD

整型数的取值范围是-231~231-1，A选项是浮点数。

1. AD

A选项中的ch2=’\’错误，反斜杠“\”在字符类型中是转义符，正确的定义应该是ch2=’\\’；D选项的x和y之间应用逗号“,”分隔，分号表示一条语句的结束。

1. C

C选项缺少数据类型的声明，static关键字代表静态变量，可以修饰变量声明。

1. A

B选项单精度浮点应定义为6.6F或6.6f，否则为双精度浮点类型数据；C选项字节型的取值范围是-27~27-1，即-128~127，超出取值范围；D选项以双引号包裹的值为字符串常量，属于引用数据类型，布尔型的取值true或false本身就是Java语言中的保留字面量。

1. CD

C选项成对的单引号中没有任何字符，非法定义；D选项的双引号应该为单引号。

1. AD

B选项使用了双等号，是关系表达式，而不是赋值语句；C选项将浮点数赋给整型变量必须进行向下的强制类型转换，且会丢失精度；D选项是a先与3做按位异或，结果再与a的原值相加，该项正确。

1. C

其他选项包含非法的特殊字符。

1. C

C选项等号左侧是字符常量，而不是变量，不允许赋值；BC选项中字符类型可以使用整形数值表示，能够与整形进行算术运算。

### 填空题

1. false
2. 8
3. false
4. false

若逻辑与符号“&&”左侧的表达式为假，则跳过右侧表达式的运算，整个逻辑表达式的结果为假，这是逻辑与的短路运算特性。

1. 8

首先计算的是(18-4)/7，而不是7+6。

1. false

逻辑与的短路运算，第二个表达式8<8为false，则略过后面计算。

1. true
2. n%7==0

### 编程题

1. 参考代码如下：

|  |
| --- |
| **public** **class** Question4\_1 {  **static** **double** volume(**int** r) {  **double** pi = 3.1415926;  **return** pi \* r \* r \* r \* 4 / 3;  }    **public** **static** **void** main(String[] args) {  **int** r = 3;  **double** v = *volume*(r);  System.***out***.println("圆球的体积：" + v);  }  } |

1. 参考代码如下：

|  |
| --- |
| **public** **class** Question4\_2 {  **public** **static** **void** main(String[] args) {  **double** n = 1;  System.***out***.println("第1格放 " + n + " 粒");  **for** (**int** i = 2; i <= 64; i++) {  n \*= 2;  System.***out***.println("第" + i + "格放 " + n + " 粒");  }  }  } |

1. 参考代码如下：

|  |
| --- |
| **public** **class** Question4\_3 {  **public** **static** **void** main(String[] args) {  **for** (**int** i = 1; i < 10; i++) {  **for** (**int** j = 0; j < 10; j++) {  **for** (**int** k = 0; k < 10; k++) {  **int** n = i \* 100 + j \* 10 + k;  **if** (n == (i \* i \* i + j \* j \* j + k \* k \* k)) {  System.***out***.println(n);  }  }  }  }  }  } |

1. 参考代码如下：

|  |
| --- |
| **public** **class** Question4\_4 {  **public** **static** **void** main(String[] args) {  **double** sum = 0;  **for** (**int** i = 0, a = 2, b = 1, tmp = 0; i < 20; i++, tmp = a, a = a + b, b = tmp) {  sum += (**double**)a / b;  System.***out***.println("分数：" + a + "/" + b + "\t序列和：" + sum);  }  }  } |

1. 参考代码如下：

|  |
| --- |
| **public** **class** Question4\_5 {  **public** **static** **void** main(String[] args) {  **int** a = 3, b = 1;  **boolean** reverse = **false**;  **do** {  **for** (**int** i = 0; i < a; i++) {  System.***out***.print(" ");  }  **for** (**int** i = 0; i < b; i++) {  System.***out***.print("\*");  }  System.***out***.println();  **if** (a == 0)  reverse = **true**;  **if** (!reverse) {  a--;  b += 2;  } **else** {  a++;  b -= 2;  }  } **while** (a <= 3);  }  } |

1. 参考代码如下：

|  |
| --- |
| **public** **class** Question4\_6 {  **public** **static** **void** main(String[] args) {  **for** (**int** i = 100; i <=200; i++) {  **boolean** isPrime = **true**;  **for** (**int** j = 2; j < i; j++) {  **if** (i % j == 0) {  isPrime = **false**;  **break**;  }  }  **if** (isPrime) {  System.***out***.print(i + " ");  }  }  }  } |