1.1缩进

4个空格常被作为缩进排版的一个单位，以表示缩进，显示程序的层次级别，增加程序的可读性。不要在源代码中保留tab字符，这是因为tab字符会随着不同用户的不同设置和不同的资源管理工具（打印、文档、版本控制等）而代表不同的宽度。

1.2行长度

尽量避免一行的长度超过80个字符，因为很多终端和工具不能很好处理。

1.3换行

当一个表达式无法容纳在一行内时，可以依据如下一般规则断开之：

在一个逗号后面断开

在一个操作符前面断开

宁可选择较高级别(higher-level)的断开，而非较低级别(lower-level)的断开

新的一行应该与上一行同一级别表达式的开头处对齐

如果以上规则导致你的代码混乱或者使你的代码都堆挤在右边，那就代之以缩进8个空格。

1.4空行

空行将逻辑相关的代码段分隔开，以提高可读性。下列情况应该总是使用两个空行：一个源文件的两个片段(section)之间类声明和接口声明之间下列情况应该总是使用一个空行：两个方法之间方法内的局部变量和方法的第一条语句之间块注释或单行注释之前一个方法内的两个逻辑段之间，用以提高可读性。

1.5每行声明变量的数量推荐一行一个声明，因为这样以利于写注释。亦即，

intlevel;//indentationlevel

intsize;//sizeoftable

要优于，

intlevel,size;

不要将不同类型变量的声明放在同一行，例如：

intfoo,fooarray[];//避免

1.6初始化

一般情况下在声明局部变量的同时初始化。

1.7简单语句

每行至多包含一条语句，例如：

argv++;//正确

argc--;//正确

argv++;argc--;//避免！

1.8复合语句

复合语句是包含在大括号中的语句序列，形如"{语句}"。例如下面各段。

大括号中的语句应该较之复合语句缩进一个层次。

左大括号"{"应位于复合语句起始行的行尾；右大括号"}"应另起一行并与复合语句首行对齐。

大括号可以被用于所有语句，包括单个语句，只要这些语句是诸如if-else或for控制结构的一部分。这样便于添加语句而无需担心由于忘了加括号而引入bug。

1.9返回语句

一个带返回值的return语句不使用小括号"()"，除非它们以某种方式使返回值更为显见。例如：

return;

returnmyDisk.size();

return(size?size:defaultSize);

1.10for语句

一个for语句应该具有如下格式：

for(initialization;condition;update){

statements;

}

1.11try-catch语句

一个try-catch语句应该具有如下格式：

try{

statements;

}catch(ExceptionClasse){

statements;

}

一个try-catch语句后面也可能跟着一个finally语句，不论try代码块是否顺利执行完，它都会被执行。

try{

statements;

}catch(ExceptionClasse){

statements;

}finally{

statements;

}

2.1基本类型变量命名规则

变量类型标识+逻辑名变量类型标识用于标识该变量所存储的数据类型，小写字母开头。

逻辑名为变量所具有的表示具体应用含义的名称，要做到见名知意，书写时要以字母开头，首字母大写。

2.2命名类

标准Java约定是使用完全的英文描述符，所有单词的第一个字母要大写，并且单词中大小写混合。类名应是单数形式。

举例：

Customer

Employee

2.3接口命名

在接口名前加前缀“I”。建议在一个接口名的前面附加上字母“I”，结果使名字变为如ISingleton或IRunnable这样。这种方法有助于将接口名与类和包的名字区分开来。