1. 注释

作用：使部分内容只为程序员可见，不为编译器所编译、虚拟机所执行；

位置：类声明前后、方法声明前后、属性声明前后、方法体中。几乎可以在一个源文件任意位置，但不能在一个关键字字符中插入注释。

注释不会被编译器编译到.class文件中

类型：

1) 单行注释： //text

——从"//"到本行结束的所有字符均作为注释而被编译器忽略

2) 多行注释： /\*text\*/

——从"/\*"到"\*/"间的所有字符会被编译器忽略

示例：

//package declaration package ch01; //允许；

package /\*package declaration\*/ch01; //允许；

class /\*class declaration\*/ FirstJavaProgram { //允许； ...}

System.out./\*out content to console\*/println("Hello Java"); //允许；

System.out.print/\*out content to console\*/ln("Hello Java"); //不允许；

3) 文档注释： /\*\* text \*/

——从"/\*\*"到"\*/"间的所有字符会被编译器忽略。当这类注释出现在任何声明(如 类的声明、类的成员变量的声明或者类的成员方法的声明)之前时，会作为 JavaDoc文档的内容；

@author 作者

@version 版本

@param 方法的参数

@return 方法返回类型

@exception 返回抛出的异常

javadoc默认不会提取@author @version

javadoc [options] [packagenames] [sourcefiles] [@files]

javadoc -help

javadoc -encoding utf-8 -d 目录 -author -version \*.java

示例：

package com.briup.ch02;

/\*\*

java中的 注释问题

@author tom

@version 1.0

\*/

public class CommentTest {

/\*\*

@param args 主方法的参数 在java命令的时候可以传入<br>

eg: java CommentTest 1 3 4

@exception null

\*/

public static void main(String[] args) {

//这个是单行注释， 下面代码单纯打印

System.out.println("你好 我被打印了");

/\*

第一行注释

第二行注释

第三行注释

\*/

System.out.println("现在开始起飞了。。。");

}

}

2.数据类型：

数据的容器，不同杯子。

Java语言把数据类型分为基本类型和引用类型。 基本类型分为数值类型、布尔类型。数值类型包含浮点类型、整数类型、字符类型。引用类型包含类类型、接口类型、数组类型、null。

八大基本类型：

boolean 取值只能是true/false

byte 字节类型 8位长度

char 字符类型 16位长度

double 双精度浮点型 64位 取值为小数 需要在数字后面添加D

float 单精度浮点型 32位 取值为小数 需要在小数后面添加F

int 整型 32位 取值为整数

long 长整型 64位 取值为整数 需要在数字后面添加L

short 短整型 16位 取值为整数

3. 标识符

1）用在 类名、方法名和变量名等地方

1) java标识符以字母、\_和$开头，不能以数字开头，后跟字母、数字、“\_”和“$”的组合;

2) 大小写敏感

3) 没有长度限制

4）不能是JAVA关键字和保留字，但可以包含关键字和保留字

举例：

合法标识符 非法标识符

try try#

GROUP\_7 7GROUP

openDoor open-door

boolean1 boolean

详细解释PPT上程序代码中各符号的实际运用；

4. 推荐命名规则：驼峰命名法则

1) 类名以大写字母开头；

例1:class Student{}

例2:class StudentUtil{}

2) 接口名以大写字母开头；

例1:interface Score{}

3) 方法名以小写字母开头；

class Test{

public void show(){}

public void findStudentById(){}

}

4) 变量名以小写字母开头；

class Test{

int age;

String name;

}

5) 常量名全部大写，多个单词以"\_"连接；

class Test{

public static final String NAME = “tom”;

public static final int MAX\_CONNECT = 20;

}

5. 关键字

Java语言的关键字是程序代码中的特殊字符。包括：

. 类和接口的声明——class, extends, implements, interface

. 包引入和包声明——import, package

. 数据类型——boolean, byte, char, double, float, int, long, short

. 某些数据类型的可选值——false, true, null

. 流程控制——break, case, continue, default, do, else, for, if, return, switch, while

. 异常处理——catch, finally, throw, throws, try

. 修饰符——abstract, final, native, private, protected, public, static, synchronized, transient, volatile

. 操作符——instanceof

. 创建对象——new

. 引用——this, super

. 方法返回类型——void

Java语言的保留字是指预留的关键字，它们虽然现在没有作为关键字，但在以后的升级版本中有可能作为关键字:goto const。

. 所有关键字都是小写；

. friendly, sizeof不是Java语言的关键字，这有别于C++；

. 程序中标识符不能以关键字命名；

6. 变量的申明和赋值

程序的基本功能是处理数据，程序用变量来表示数据；

编程的本质就是对内存中数据的访问和修改。

程序中必须先定义变量才能使用；

定义变量是指设定变量的数据类型和变量的名字，

定义变量的基本语法为： 数据类型 变量名[=初始值]；

Java语言要求变量遵循先定义，再初始化，然后使用的规则。

变量的初始化是指自从变量定义以后，首次给它赋初始值的过程。

例：

int a; //定义变量a

a = 1; //初始化变量a

a++; //使用变量a

int b=a; //定义变量b, 初始化变量b, 使用变量a;

b++; //使用变量b

int c,d,e;同时声明多个变量

强类型：

. 所有变量必须先声明，后使用

. 指定类型的变量只能接受与之匹配的值

7. 类型转换

1) 使用在基本数据类型和实例对象之间的转换。

2) 隐式转换和显式转换

3) 隐式转换是在运行期间转换，从子类转换到父类。第五章会详细讲解。

4) 显式转换，缩小变化。