Java的基本数据类型

Byte Int short long float double char Boolean

算术符

Java流程控制:顺序，分支，循环

/\*\*

\* 记录操作执行总时间.

\*

\* @param <T> the generic type

\* @param clazz the clazz

\* @param methodName the method name

\*/

public <T> void recordTime(Class<T> clazz, String methodName) {

long start = System.currentTimeMillis();

System.out.println("start: " + start);

Method[] declaredMethods = clazz.getDeclaredMethods();

for (Method method : declaredMethods) {

String name = method.getName();

if (name.equals(methodName)) {

try {

method.invoke(clazz.newInstance());

} catch (Exception e) {

e.printStackTrace();

}

}

}

long end = System.currentTimeMillis();

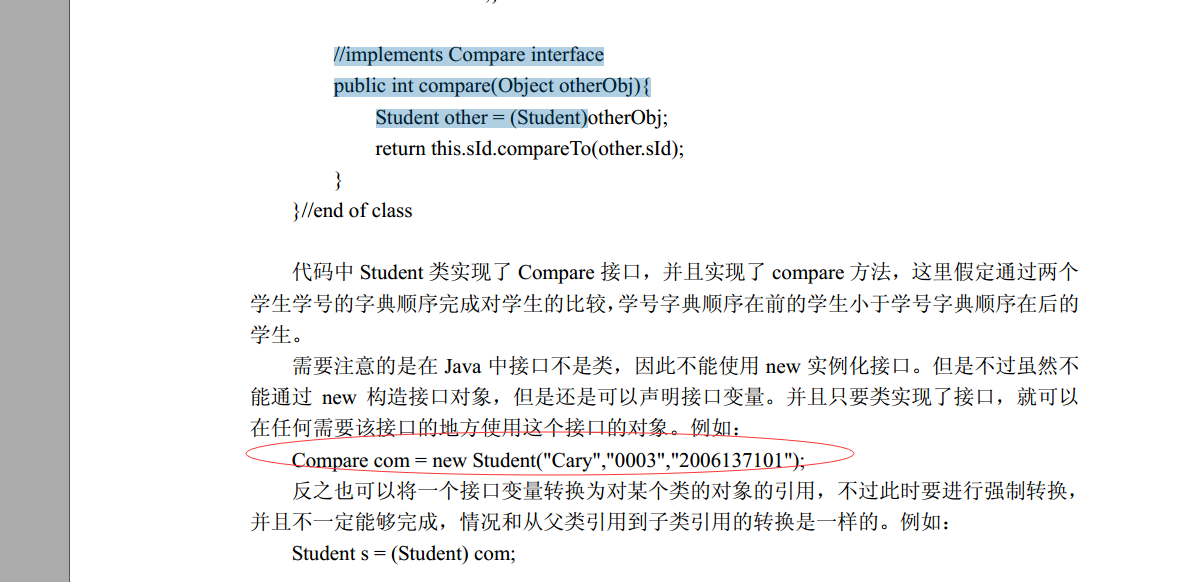
System.out.println("end: " + end);

System.out.println("duration: " + (end - start) + " ms");

}

This关键字：调用其他的构造方法，对隐式参数的引用

Super关键字：调用父类构造方法和父类方法



这种场景怎么使用？

Exception：处理异常的两种方法，try 。。。catch。。。； throws 或者throw 抛出异常

操作符new 的作用：为对象开辟足够的内存空间

数据结构：

数据

数据元素：数据的基本单元，数据集合的个体

数据对象：

数据结构： 逻辑机构 ；存储结构

逻辑结构：集合，线性结构，树形结构，图状结构

存储结构：数据元素本身的存储，以及数据元素之间的额关系表示

二元组

图形

线性结构 linearity：除了第一个元素外，每个数据元素有却只有一个直接前驱元素，除了最后一个元素外，每个数据元素有且只有一个直接后续元素，这种数据元素之间一对一关系

树形结构tree：除了第一个元素外，每个数据元素有且只有一个直接前驱元素，但可以有对个直接后续元素，这种数据元素之间1对N的关系

图状结构graph：这种数据元素之间M对N的关系

数据元素之间的关系：数据映像，非数据映像 –》

顺序存储结构和链式存储结构

数据类型：一组性质相同的数据元素的集合和加在这个集合上的操作

作用：隐藏计算机硬件及其特性和差别，使硬件对于用户来说是透明的，即用户不需要知道数据类型是怎么实现的而可以使用它；用户可以使用数据类型使用的操作，而方便的解决问题

抽象数据类型ADT(数据对象，关系集，操作集)：由一组数据模型和该数据模型的一组操作

算法：指令的集合，为解决特定问题的一系列操作。有着明确的计算过程，一一组数据进行输入，并产生一组数据集合作为输出

五个特性：输入，输出，可行性，有穷性，确定性

可行性：算法的每条指令都是可以实现的，均可在有限的时间内完成

有穷性：算法的指令是有限，而且每条指令都可在有限的时间内完成

确定性：对于每一条合法的输入，其输出都是唯一的

确定性算法；随机算法

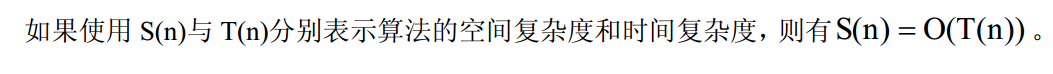
算法的时间复杂度（渐进时间复杂度）：如选择排序的时间复杂度为Ο(n^2)

Ο:运行时间的上限，即时间复杂度的上限

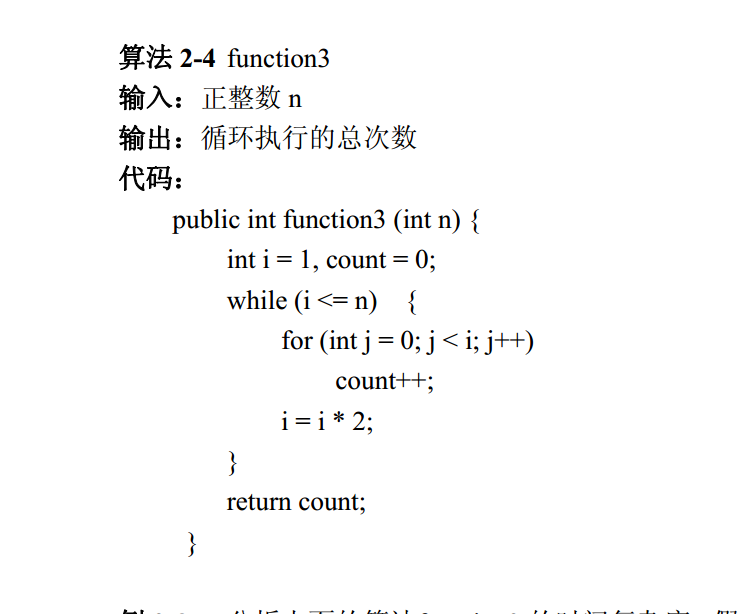
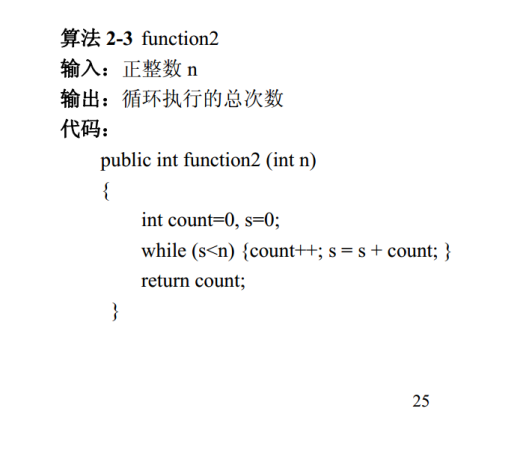
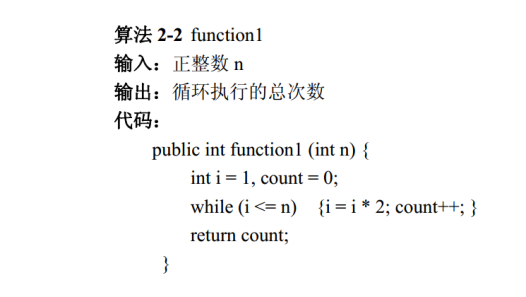
Ω：在时间复杂度常数因子的范围内给出时间复杂度的下界Ω(n^2)

θ：算法时间复杂度的精确阶

空间复杂度：为了求解实例而求解的步骤所使用的存储空间的数目但并不包括存储输入实例的空间



计算循环次数：

分析最高频度的基本操作：

最好，最坏和平均分析：