

面向对象程序计的平台——C++与 JAVA 的比较

高建州

(中国电信股份有限公司海丰分公司, 广东 汕尾 516400)

摘要: 通过理论与实例的比较, 对目前比较流行的两种面向对象的编程语言 C++ 与 Java 从 7 个方面作一比较, 希望为程序的设计与教学提供帮助与选择。

关键词: C++; JAVA; 程序设计; 编程语言

中图分类号: TP311 文献标识码: A 文章编号: 1009- 3044(2008)21- 30453- 02

Comparison of the Object- oriented Procedure Counts Platform C++ and JAVA

GAO Jan- zhou

(Haifeng Subsidiary company of Chinese Telecommunication Limited Liability Company, Shanwei 516400, China)

Abstract: This article made a comparison of two of the popular object- oriented programming language- C++ and Java from 7 aspects compares with the theory and experience. And we hoped that it can provide the help and choice for the procedure design and the teaching.

Key words: C++; JAVA; Programming; Programming language

计算机语言, 自 20 世纪 80 年代到 90 年代中期, 一直以 C、PASCAL、FORTRAN 为主, 其中以 C 语言最为流行, 在面向对象编程的思想被广泛接受和实现后, C++ 和 Java 逐渐成为计算机语言的主宰。C++ 语言在兼容原有最流行的 C 语言的基础上, 加入了面向对象的有关内容和规则。由于它的很多语法规则与 C 语言相近, 所以很容易为广大的 C 程序员所接受; 同时 C++ 所具有的面向对象功能简化了应用软件的开发、设计和维护, 为开发大型软件提供了很大的方便。但是, 由于本身是面向过程语言的补充, C++ 也有很大的不足。Java 是在 C++ 之后出现的, 从语法特征上看, Java 主要源自于 C++, 但在许多方面进行了改革, 使得熟悉 C++ 的程序员能很方便地进行编程。同时为了实现其简洁、健壮、安全等特点, 也摒弃了 C++ 中许多不合理的内容。本文将从多个方面将 Java 与 C++ 加以比较, 希望为程序的设计与教学提供帮助与选择。

1 一般特点比较

C 语言设计灵活, 功能强大, 但不支持面向对象程序设计, 为了让编程人员在使用面向对象的概念时保持对 C 语言的向后兼容性, Bjarne Stroustrup 在 C 语言上引进了“类”的概念, 称作 C++ 语言, C++ 既兼容以前 C 语言的过程程序设计, 又支持面向对象的程序设计, 其主要特点是: 允许单继承和多继承, 支持数据封装; 允许函数名和运算符重载, 支持动态联编。Java 由 Sun 公司开发, 是一种纯面向对象语言, 运行在网络浏览器上, Netscape 和 IE 浏览器都迅速地把这种技术整合其中, Java 传播得极快。其主要特点是在 JVM 上运行, 语言简单, 多线程, 分布式, 解释执行, 可在不同的操作系统层面移植, 能够在大型机上运行。

2 数据转换

在 C++ 语言中, 支持数据类型的自动转换, 自动数据类型转换使得两个数据类型互不兼容的变量可以相互赋值, 不需给出显式说明, 这有时会导致一些问题, 最常见的是精度损失。例如, 如果把一个带符号的 32 位整数赋给一个无符号整数, 则所有的结果均为正数, 不安全因素大大增加。C++ 语言可以通过指针进行任意的类型转换, 而 Java 语言不支持这种自动类型转换方式, 要求编程人员显式实现数据类型转换, 系统要对对象的处理进行严格的相容性检查, 防止不安全的转换。在 C++ 语言中, 不同的平台上, 编译器对简单的数据类型如 int、float 等分别分配不同的字节数。例如: int 在 IBMPC 上为 16 位, 在 VAX- 11 上就有 32 位, 导致了代码数据的不可移植性。在 Java 中, 对数据类型的位数分配总是固定的, 而不管是在任何的计算机平台上, 因此就保证了 Java 数据的平台无关性和可移植性。

3 数组

尽管 C++ 数组与 Java 数组表面上类似, 但与 C++ 相比, 数组在 Java 中用类来实现, 而且对于下标进行检查, 每个 Java 数组都有一个只读的成员, 通过它可知道数组有多大。而且一旦超过数组边界, 运行期检查会自动丢弃一个异常。Java 这种对于数组下标进行检查的机制, 很好地解决了数组越界这一 C++ 中不作检查的错误, 确保 Java 程序不能破坏不属于它的内存空间。所有的动态内存申请都将通过 new 运算符来完成, 包括数组内存空间的申请。所有数组都是在内存“堆”里创建的, 我们可将一个数组分配给另一个。数组标识符属于第一级对象, 它的所有方法通常都适用于其他所有对象。和 C++ 完全不同的是, Java 数组类型定义不同时产生数组所需要的内存空间而必须通过 new 运算符指定数组的维长并分配内存后, 才可以使用数组元素。而且, 在 Java 中得到的不是指针, 而是引用。程序只能通过引用来访问数组元素或者对象, 不能像指针那样通过修改引用的值来指向另外一块内存区域, 减少了出错的机会, 也可以使内存动态地由系统进行回收。

4 参数传递

C++ 中的参数传递

收稿日期: 2008- 05- 17

作者简介: 高建州(1978-), 男, 广东汕尾人, 助理工程师, 学士, 主要从事计算机通信维护工作。

```
#include<iostream.h>
typedef int* plnt;
void change1(int x)(x=1;)
void change2(int x)(* x=1;)
void chaeege3(plnt x)(* x=1;)
void change4(int &x) (x=1;)
main(){
int i;j;k;l; plnt pk; i=j=k=l=();pk=&k;
cout<< "i(i): "<<i; changel(i); cout<< "-> "<<i<< "\t ";
cout<< "j(&j): "<<j;change2(&j);cout<< "-> "<<j<< "\t ";
tout<< "l(pk): "<<k "change3(pk) "cout<< "-> "<<k<< "\t ";
tout<< "l(1): "<<l;chaeege4(1);cout<< "-> "<<l<<endl;
}
```

运行结果为: i(i): 0 -> 0 j(&j): 0 -> 1 k(pk): 0 -> 1 l(0) 0 -> 1。表明在 C++中,既有传值的方式。也有传引用的方式,由被调函数的形参是否是引用类型来决定。

Java 中参数的传递方式

```
import java.awt.*;
public class Class1{
public static void change(int i){i=1;}
public static void changeInteger i){i=new Integer(1);}
public static void change(String s){s= "A ";}
public static void change(StringBuffer st){st.insert(0, "A ");}
public static void change(Button bt){bt.setLabel( "Delete ");}
public static void main(String[] args)throws Exception{
int i=0;String Integer j=new Integer(0); s=new String( "a "); StringBuffer st=
new StringBuffer( "a ");
```

程序运行结果为:

Int:0 -> 0 Integer: 0 -> 0 String:a -> a StringBuffer: -> Aa Button:Print -> Delete

上例中的结果显示,被调方法的形参都是变量,在被调方法中对变量值的改变,有的影响到了主调方法,有的却没有。显然,被调方法中,形式参数从形式上是一致的,不同的是形式参数的变量类型。

5 自动垃圾收集机制

Java 中没有 C++破坏器”,变量不存在“作用域”的问题。Java 有个 finalize()方法是每一个类的成员,它在某种程度上类似于 C++的“破坏器”。但 finalize()是由垃圾收集器调用的,而且只负责释放“资源”,如打开的文件、套接字、端口、URL 等等。如需在一个特定的地点做某件事情,必须创建一个特殊的方法,并调用它,不能依赖 finalize()方法。由于 Java 不支持破坏器的概念,针对类内的基础类以及成员对象,Java 采用垃圾收集器自动清除,所有对象都会被当作“垃圾”收集掉。自动垃圾收集意味着在 Java 中出现内存漏洞的情况会少得多,但也并非完全不可能。若调用一个用于分配存储空间的固有方法,垃圾收集器就不能对其进行跟踪监视。而在另一方面,C++中的所有对象必须用 finalize()方法破坏,然而,内存漏洞和资源漏洞多是由于编写不当的 finalize()造成的,或是由于在已分配的一个块尾释放一种资源造成的。Java 垃圾收集器是在 C++基础上的一种极大进步,使许多编程问题得以解决。

6 出错与异常控制

在 PC 机上编写程序经常会遇到由于程序中的错误而使计算机崩溃的情况。传统的面向过程的编程的语言,如 Fortran、C 等,出错处理的任务几乎完全由编程人员来承担。与面向过程语言不同,面向对象语言采取了另一种出错处理方式,即所谓“捕捉——抛出”方式。按照这种方式,运行时系统和应用程序抛出各种标准类型的错误,程序捕捉这些错误并进行相应的处理。这样一来,出错和异常情况处理变得比较规范,编程人员可用一致的方式对错误进行处理。C++和 Java 都采用了这种处理方式,而且 Java 还增加了一项新内容-Try 方法,更加丰富了这种处理方式。Java 的异常规范比 C++的出色得多。Java 中的所有异常都是从基础类 Throwable 里继承而来的,所以可确保我们得到的是一个通用接口。丢弃一个错误的异常后,不是象 C++那样在运行期间调用一个函数,Java 异常规范是在编译期间检查并执行的。被取代的方法必须遵守那一方法的基础类的异常规范,它们可丢弃指定的异常或者从那些异常衍生出来的其他异常。这样一来,最终得到的是更为“健壮”的异常控制代码。

7 内存管理

用 C++写软件时,程序员需要仔细地处理内存的使用,当一个内存块不再使用时,必须释放它。这样程序员需要仔细留意内存的使用情况,在大的项目中这是十分困难的。内存管理不当通常是造成系统故障和存储空间浪费的主要原因之一。在 Java 中,程序员不需要关心内存管理的问题,Java 系统内嵌了垃圾收集器,它扫描内存,自动地释放不再使用的内存块。Java 允许你建立尽可能多的对象,但从来不需要你去消除它们。这就使 Java 程序的编写变的简单了,而且减少了程序中因内存管理问题而出错的可能性。

8 发展前景

Java 是由 C++衍生而来,其语言风格与 c++十分类似。如果你已经懂得一些 C 和 C++语言,则很快可以学会 Java 语言;另一方面 Java 又比 c++简单,Java 抛弃了 c++中的一些不是绝对必要的功能,如头文件、预处理器、指针、结构联合和隐式的类型转换、操

(下转第 461 页)

本栏目责任编辑: 谢媛媛

造(Construction)和移交(Transition)四个连续的阶段。每个阶段都终结于一个良好定义的里程碑。其实,每个阶段就是两个主要里程碑之间的时间跨度^[6]。在每个阶段结束时要依据里程碑目标进行工作评估,以确定是否实现了该阶段的目标以及是否可以进入下一个阶段。如图5所示为各阶段和里程碑的关系。

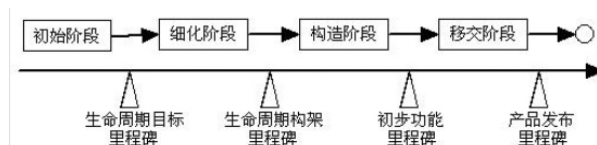


图5 阶段和里程碑

在初始阶段,项目组集中精力理解最初的需求,确定范围并组织项目。要理解最初的需求,可能要进行业务建模及其他基本的建模活动。细化阶段是建立系统架构的基线,以便为构造阶段的主要设计和实施工作提供一个稳定的基础的阶段。构造阶段是将主要精力集中在现象设计、实现以及测试来充实一个完整的系统的阶段。移交阶段是系统正式投入运行前的阶段,要达到的主要目标是确保软件完全满足用户需求。

工作流,是产生具有可观察结果的活动序列,每个工作流产生一些有价值的工件。RUP共有9个工作流,其中分为6个核心过程工作流:业务建模、需求、分析和设计、实现、测试、部署工作流;3个核心支持工作流:项目管理、配置和变更控制、环境工作流。这9个工作流并不是顺序执行的,而是在项目中轮流被使用,在每一次迭代中以不同的重点和强度重复。核心工作流的迭代开发循环模型如图6所示。

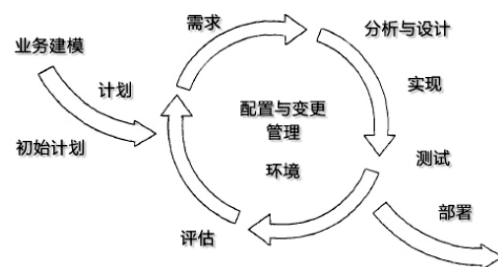


图6 核心工作流的迭代开发循环

4 结论

RUP是新一代软件工程开发方法,近年来在软件开发领域中得到了广泛关注。RUP采用面向对象技术,在迭代的开发过程、需求管理、分析与设计、可视化软件建模、验证软件质量及控制软件变更等方面为软件开发人员提供了相应的准则、模板和工具。在软件开发过程中应用RUP,可以规范管理,降低软件复杂性,减少软件开发风险,提高软件质量。

参考文献:

- [1] Ian Sommerville. Software Engineering[M]. 北京:机械工业出版社,2006.
- [2] Scott W. Ambler. The Unified Process Elaboration Phase Best Practices in Implementing the UP[M]. 北京:机械工业出版社,2005.
- [3] 余八一. 现代软件工程过程方法探析[J]. 科技资讯,2007,1(02):45.
- [4] 张发生. 基于RUP的软件过程及应用[J]. 计算机工程与应用,2003(30):104-107.
- [5] 陆永忠,饶璟祥. 小型软件项目RUP裁剪模型的研究[J]. 计算机工程与设计,2007,28(03):3027-3030.
- [6] 吕西红,陈志刚. 统一软件开发过程RUP中的关键技术研究[J]. 信息技术,2006(1):27-29.

(上接第454页)

作符重载等,使Java比C++更容易学习,其程序的可读性也更强。同时Java是一种更纯粹的面向对象程序设计语言。面向对象编程具有多方面的吸引力。对管理人员,它实现了更快和更廉价的开发与维护过程。对分析与设计人员,建模处理变得更加简单,能生成清晰、易于维护的设计方案。对程序员,对象模型显得如此高雅和浅显。此外,面向对象工具以及类库的巨大威力使编程成为一项更使人愉悦的任务。

总之,C++语言功能强大,操作系统和Office应用程序都是由C++程序编写的,可满足对功能的需求,Java语言是一门很优秀的语言,具有面向对象、与平台无关、安全、稳定和多线程等优良特性,是目前软件设计中极为健壮的编程语言。C++与Java已成为网络时代最重要的语言之一。

参考文献:

- [1] 池静,邢秀娥. 基于Java和C++语言的安全性讨论[J]. 河北建筑科技学院学报,2005(4):86-88.
- [2] 刘哲,史诗. Java和C++语言安全机制及网络安全规范的讨论[J]. 电化教育研究,2002(7):65-67.
- [3] David L H, Mirmca SLA practical flow-sensitive and context-sensitive C and C++ memory Leal(detector)[J]. ACM press,2003,38(5):168-181.
- [4] Bryan Bates. C as a first language: comparisons with C++[J]. The Consortium for Computing in small College,2004,19(3):89-95.