

低年级讨论班

第15章

漂亮的设计会给你带来长远的益处

数学科学学院 杨嘉骐

作者简介

- 作者Adam Kolawa, Caltech理论物理学博士, 现任Parasoft(以软件测试技术闻名的公司)的CEO, 重点关注领域为物理学中的辅助计算与并行计算。

漂亮代码的定义

- 最基本观点：必须是可工作的。
- 代码必须精确、高效的完成我们再设计它所预期的任务，而不能在行为上存在任何的歧义。
- 只有那些我可以使用，以及重复使用，并且不会对他们存在任何“他们是否可以为我提供正确答案”怀疑的代码才可以被称为漂亮的代码。

漂亮代码的定义

- 业界最重要的使命：与他人共享代码，同时并不要求别人去分析代码内部找出它的工作原理。就像小汽车一样，我们不会去研究其内部机理，但需要欣赏它的外观和信任它的功能。
- 我们必须清晰的告诉别人如何使用、理解它们，将它们应用到人们所面对的问题上去、并且还能够很容易让人们验证它的用法是否正确。

外在美

- 本文第一部分讨论代码的外在美：即那些保证代码准确以及可靠并使得开发人员愿意重用它的因素，以及如何使重用变得简单的因素。

外在美

- CERN库
- 欧洲核子研究中心（CERN）是世界上最大的粒子物理研究实验室。它成立于1954年，位于日内瓦的西北部，目前拥有世界上最大的粒子加速器（LHC）和与之配套的世界最大的计算网络，CERN的研究人员Tim Berners-Lee于20世纪90年代初在CERN发明了万维网，以便使世界各地不同大学和研究所的科学家能够自动共享信息。
- CERN库是CERN开发的用于数学计算的程序库，目前仍然在更新并有了Windows下的版本。

外在美

- LAPACK与LINPACK
- CERN库的线性代数部分发展成了LAPACK (Linear Algebra Package) ，由橡树岭国家实验室、加州大学戴维斯分校和伊利诺伊大学等联合开发，是用Fortran 77编写的数值线性代数库。
- LAPACK最新版本为lapack-3.0，LAPACK网站：
<http://www.netlib.org/lapack/index.html>

外在美

- Fortran语言
- Fortran 77语言为一种实现了结构化的编程语言，语法与C语言有相似之处。下面列举一些与C语言差距较大的语句。
- 逻辑表达式：N.LT.0“.LT.”代表小于，整个语句含义是若 $N < 0$ 则返回true.
- CALL调用子例程

外在美

- 循环

```
DO 10 I = 1, J - 1
```

```
    WORK13( I, J ) = ZERO
```

```
10 CONTINUE
```

DO 标号(,)I = 循环初始值,循环终止值,每次循环增加量 (不填为1)

循环体执行语句

标号 继续执行语句 (通常为CONTINUE)

外在美

- 本文选取LAPACK库中SGBSV例程作为例子。
- 此例程用于计算带状矩阵（矩阵中不为0的值集中在对角线附近）求解线性方程组。矩阵线性方程式的时间复杂度为 $O(Bn^2)$, 其中 B 为带状矩阵宽度, n 为矩阵大小, 而 $B \ll n$; 高斯消去法解一般线性方程式的时间复杂度为 $O(n^3)$, 针对不同矩阵采取不同方式计算效率较高。

设计之美

- * Arguments
- * =====
- *
- * N (input) INTEGER
- * The number of linear equations, i.e., the order of the
- * matrix A. $N \geq 0$.
- *
- * KL (input) INTEGER
- * The number of subdiagonals within the band of A. $KL \geq 0$.
- *
- * KU (input) INTEGER
- * The number of superdiagonals within the band of A. $KU \geq 0$.
- *
- * NRHS (input) INTEGER
- * The number of right hand sides, i.e., the number of columns
- * of the matrix B. $NRHS \geq 0$.

例程开始处有大段注释，注释描述了使用该例程的功能和使用方法，便于使用者在不深入理解每一行代码的基础上就能理解其作用和如何使用。

同时注释详细给出了每一个参数的意义和输入范围，易于正确使用。

设计之美

- * AB (input/output) REAL array, dimension (LDAB,N)
- * On entry, the matrix A in band storage, in rows KL+1 to
- * 2*KL+KU+1; rows 1 to KL of the array need not be set.
- * The j-th column of A is stored in the j-th column of the
- * array AB as follows:
- * $AB(KL+KU+1+i-j,j) = A(i,j)$ for $\max(1,j-KU) \leq i \leq \min(N,j+KL)$
- * On exit, details of the factorization: U is stored as an
- * upper triangular band matrix with KL+KU superdiagonals in
- * rows 1 to KL+KU+1, and the multipliers used during the
- * factorization are stored in rows KL+KU+2 to 2*KL+KU+1.
- * See below for further details.
-

参数AB只存储带状以内的部分，不同的存储方式要求使用不同的算法，这样不存储带状以外的0元素有避免了把带状以外的0元素误写为其他元素的风险。同时AB也被当做输出参数，在不同的例程中有不同的输出。从而节省了内存。

设计之美

- 编程时的思维局限
- 一般来说写程序时会带着这样的思维模式
- 1. 我需要编写什么
- 2. 我把它编写出来看能否工作
- 3. 如果能够工作，试图将它推广至其他问题

设计之美

- 于是我们在编写程序时实际上没有太多考虑扩展性，于是在很多方面给程序造成了或多或少的局限性。比如编程者觉得内存微不足道而不加以节省，那么这个程序就不能应用于大规模问题处理。如果编程者没有节省对数据的访问次数。那么遇到建立在数据库系统上的应用时，频繁访问数据库势必造成速度的下降。
- 所以我们应该优先考虑扩展性，而不是在编写完后再优化扩展性。

设计之美

- INFO = 0
- IF(N.LT.0) THEN
- INFO = -1
- ELSE IF(KL.LT.0) THEN
- INFO = -2
- ELSE IF(KU.LT.0) THEN
- INFO = -3
- ELSE IF(NRHS.LT.0) THEN
- INFO = -4
- ELSE IF(LDAB.LT.2*KL+KU+1) THEN
- INFO = -6
- ELSE IF(LDB.LT.MAX(N, I)) THEN
- INFO = -9
- END IF

INFO参数用来处理错误机制，不同的异常返回不同的值，然后根据不同的值用户可以了解到具体出了什么问题。

在编程前需要经过充分考虑，要么正确的处理问题，要么告诉用户它做不到（最好能告知用户哪里出了问题），但程序不会崩溃。

设计之美

- 大多数例程都提供了测试程序，符合“不需要深入理解代码就能知道程序是否符合用户的要求”的准则。提高了用户对程序正确性的信心。

内在美

- 简洁之美
- 节俭之美
- 流程之美

内在美

- 简洁之美
- 漂亮的代码必是短小的代码。如SGBSV的主例程，首先检验输入参数，然后调用一个子例程进行LU因数分解，然后再调用一个子例程求解，整个流程非常清晰简明。
- 将不用作用的代码编写成例程（函数、方法）不仅可以使代码简明易读、结构清晰。同时可以便于代码重用，当代码过于复杂，阅读难度过高时，程序员宁愿重写一遍。

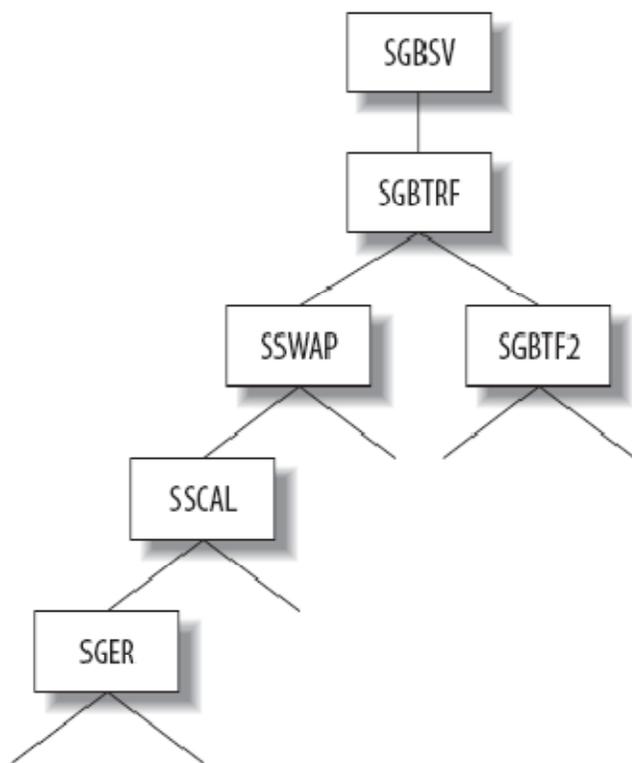
内在美

- 漂亮的代码应该很容易被理解，炫耀程序员对编程语言知识掌握能力的代码是不可取的，太多的继承与重载导致某些代码需要了解整个继承结构才能了解每一个类的作用，让使用者难以知道其功能及代码是否正确。
- Rem.继承与重载作为面向对象程序设计的主要特点，目的是为了更好的与人类在正常世界中的思维拟合，从而提高程序的可读性、可重用性、可扩展性等等。但不恰当的应用造成程序局部可读性差，是一种对继承不正确使用的表现。

内在美

- 节俭之美
- 程序员需要认真的考虑对计算速度和内存使用的优化，如果为扩展性考虑，在细节处如使用浮点运算还是整数运算，内存是否可以重用等等也是值得考虑的，假设运算时间和内存都是无限制的代码不能算是漂亮的代码。

内在美



- 流程之美
- 将问题细化，在不同情形使用不同的方法以达到最好的效果是有益处的。效果可能好于在单一算法上改进。
- 结构良好的流程有益于程序的可读性，同时分割的子例程的重用性较好。

结论

- 漂亮的代码必须是简短、明确、（对计算机资源的使用）节俭以及考虑到现实情况的代码。作者认为对美的最终界定是它是否能经的起时间的考验。CERN库经过30年后依然被使用，证明了他是真正的漂亮代码。



Thanks