数学对设计 C++ 语言里标准模板库的影响

蒋 迅

数学是科学的女王，与计算机等很多应用学科都有着千丝万缕的联系。有人说，从事数学研究的人可以轻易转行从事其他科学研究，但是从事其他科学研究的人想转行从事数学研究却并非易事，由此可见数学的威力。数学来源于实践，又反作用于实践，在理论和实践的发展中交叉融合，历久弥香，亦表现出累积的特性，标准模板库就是最大公测度经过 2500 年的沉淀和发酵之后精炼成型的，它在计算机语言中发挥着重要作用。

1. C++ 语言的标准模板库

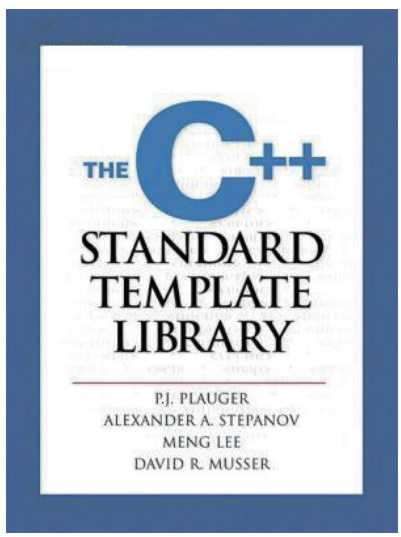

图 1 《C++ 标准模板库》封面 /Prentice Hall 出版社 [1]

C++ 语言（以下简称为 C++）比 C 语言增加了一个很好的功能：标准模板库（Standard Template Library，简记为 STL）。标准模板库使得 C++ 不但有了同 Java 语言一样强大的类库，而且还有了更大的可扩展性。标准模板库是由斯特潘诺夫（Alexander Stepanov）在 1979 年前后发明的（如图 1），这也正是计算机科学家斯特劳斯特鲁普（Bjarne Stroustrup）发明 C++ 的年代。巧合的是，他们二人不但同一年出生，而且学的都是数学专业（后者兼得计算机科学学位），就好像两人相约来到这个世界上一样，共同为计算机语言的大树添枝加叶，使之愈加繁茂。斯特潘诺夫素有标准模板库之父

[1] P. J. Plauger, A. A. Stepnanov, M. Lee, D. R. Musser, The C++ Standard Template Library, Prentice Hall, 2000.

的赞誉，斯特劳斯特鲁普则被冠以 C++ 之父的美名，注定都会名列史册了。

C++ 标准程序库是类库和函数的集合，使用核心语言写成。标准模板库是 C++ 标准程序库的子集，包含 4 个组成部分：算法、容器、函数对象和迭代器。这 4 部分构成 C++ 中现有的共同类的集合，只要支持一些基本运算法则，就可以用于任何一个内置型和用户自定义的型。斯特潘诺夫之所以能做出这项杰出工作，实际上源于他对数学特别是抽象数学的深刻理解和对数学发展历史的了悟。

2. 斯特潘诺夫和数学

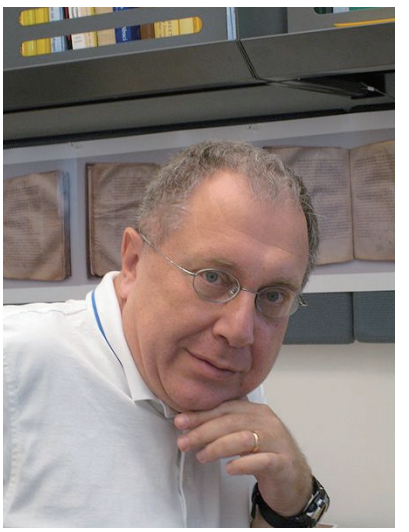

图 2 斯特潘诺夫 / 维基百科

斯特潘诺夫 1950 年 11 月出生于莫斯科。1967 年到 1972 年间，在莫斯科国立大学学习。1973 年在莫斯科市立师范学院（Moscow District Pedagogical Institute）获得教师资格证书。他并不是很喜欢抽象的数学，从来就没有对像玉河数（Tamagawa number）或考克斯特群（Coxeter group）之类的数学产生过特别的兴趣。与哈代热衷从事数学研究不同，他渴望做点具体的实事。尽管他一心要离开数学，梦想到其他领域施展才华，但是他毕竟在数学学堂里接受了严格的数学训练，所以他拥有坚实的数学基础，对数学发展史亦有深刻理解。毕业后，他有幸成为了一名程序员，而计算机科学领域与数学正好有着千丝万缕的联系，使他可以乘上数学的东风，最终在计算机领域成为了足以流芳百世的业界翘楚。不过，他讲的一个小故事说明数学基础并不等于计算能力：他第一个计算机考试是通过 - 不通过那种，而他却考了好几次。他以为读完课本就可以考过，最后发现必须亲自写程序才能学会。

1972 年，斯特潘诺夫和别人一起为一个水电站开发一台微型计算机，他参与了设计、测试、操作系统和编程工具的全部环节，从中得到了软件可靠性和有效性的第一手经验。同时，他还读到了著名计算机科学家高德纳（Donald Ervin Knuth）和戴克斯特拉（Edsger W. Dijkstra）的书。而他们两人也都是从数学起家的。一直到很多年以后，斯特潘诺夫还经常阅读这两位计算机学家的著作并得到新的感悟。

1976 年，他移民美国，受聘于通用电气研究中心。在通用电气公司的 5 年对他来说至关重要。在那里，斯特潘诺夫与其同事一起为通用电气公司开发出一种新的程序语言“Tecton”。为此，他大量地阅读文献，其中有关于计算机语言设计的论文，也有

亚里士多德的著作，还有 14 世纪逻辑学家奥卡姆的威廉（William of Occam）的逻辑理论。由于他经受过严格的数学训练，所以从古代逻辑学理论中领悟到了在自然语言中的逻辑结构及其性质。

1984 年，斯特潘诺夫成为了纽约大学理工学院（Polytechnic Institute of New York University）计算机系的助理教授。他一边教一边学，从而掌握了更多的计算机理论。他还开发了一套算法库和数据结构的框架，这就是他后来合作开发的 Ada 泛型（Ada Generic Library）的雏形。

在离开纽约大学理工学院后，斯特潘诺夫在贝尔实验室短暂工作了一段时间，负责开发 C++ 的算法库。1988 年，他又到惠普总部工作，主要负责计算机的数据储存系统。但是他仍然对泛型念念不忘。1993 年，他终于有机会从事泛型计算的研究，成功开发出了 STL。但在 STL 被 ANSI/ISO 接受为“标准”（1994 年 7 月）后仅仅 5 个月，惠普却意外停止了他的研究。由于他对 STL 情有独钟，万般无奈下不得不离开惠普。1995 年，他跳槽到硅图 [计算机系统] 公司（Silicon Graphics [Computer System] Inc.，简记为 SGI），继续从事他喜爱的事业，组织起一批人进一步完善 STL。

斯特潘诺夫对计算机科学的最大贡献就是发明了 STL，STL 也自然成为了他的一个标签。说起来，他也是因祸得福，因为 STL 缘于他的一次食物中毒事件。他曾幽默地说：“STL 是一次细菌感染的结果。”那是 1976 年，当时他还在苏联。有一次，在吃了生鱼后不幸食物中毒，不得不住进了医院。但躺在病床上的他并未停止思索。他在思考一个并行约简算法时，突然灵光一闪，这个算法之所以能够并行运行是由于其运算对象的半群结构，亦即算法是定义在代数结构之上的。但是对于一个算法，如何找到一个结构，使得这个算法是定义在这个结构上的呢？之后，又过了两年的时间，他逐步意识到，必须在常规的公理之上增加一定的限制，扩展自己要研究的结构，于是他决心搞出一套抽象的结构来，由此开始了长达 15 年的潜心研究。他要为算法找到可以更广泛表达的途径，有时候可以为一个算法琢磨一个月。起初，身边的人不能理解他为什么要这样做，但他不管这些，坚持在自己认定的方向上努力。

总之，从 1972 年以来，斯特潘诺夫先是在苏联，后来在美国，一直在从事与操作系统、编译器和编程库等与编程相关的工作。似乎从表面看来，他做的工作与数学无关，俨然一副已和数学分道扬镳的态势。其实不然，正如他自己所说的那样，我们不需要把编程硬性地联系到数学上，说到底，编程的性质是数学学科，因为计算机程序就是操作整数、形状、序列等等一些抽象的数学对象，编程和数学的联系是内在的。可以说，他创造的 STL 就是这一思想的自然产物。

斯特潘诺夫说：“我相信，迭代器对于计算机科学的重要性就像环论和巴拿赫空间对数学的重要性一样。每当我看一个算法时，都会试图找到一个定义它的结构。”因此，斯特潘诺夫不餍足于仅给出一个具体的算法，他力图给出算法的一般性描述。功夫不负有心人，有了这样的一般性思想，STL 也就成了顺理成章的自然结果。虽然人们往往不能理解他的这种做法，不过幸运的是，STL 成为了意外中的必然，人们欣然接受了它。毋庸讳言，这是广大程序员的福气。

3. 最大公测度：上下 2500 年

斯特潘诺夫在世界各地做过很多精彩的演讲[2]，其中有一个是“最大公测度：上下

[2] A. Stepanov, Greatest Common Measure: the Last 2500 Years, http://www.stepanovpapers.com/gcd.pdf.

2500 年”（Greatest Common Measure: the Last 2500 Years）。下面我们来共同体味一下他在演讲中的思想脉络。

我们都知道，两个整数 n 和 m 的最大公约数是指这两个整数的公共因数中最大的一个，记作 gcd(n, m)。“gcd”这个记号是最大公约数的英文名称“greatest common divisor”中各个单词的首字母的组合。比如，$n = 54$ 和 $m = 24$，我们先对它们进行分解，n 的因子为：1, 2, 3, 6, 9, 18, 27, 54；m 的因子为：1, 2, 3, 4, 6，8, 12, 24。它们的公共因子为：1, 2, 3, 6。所以，gcd(54, 24) = 6。因此，如果我们要度量这两个数的话，最大的公约数是 6。从这个意义上说，人类对这个问题的研究已经有了 2500 年的悠久历史。

图 3 古巴比伦泥版文书“普林顿 322”/ 维基百科

现存于美国哥伦比亚大学图书馆的古巴比伦泥版文书“普林顿 322”（Plimpton 322，如图 3），其形成年代大约在公元前 1600 年之前，因其最初的收藏者为普林顿且收藏编号为 322 而得此名号。它记录了 15 个勾股数组，涉及的最大的一个勾股数组是（18541, 12709, 13500）。相信即便哪个数学神人也不能随意写出这些数，因此，可以推断，古巴比伦数学一定已经有了某种算法。

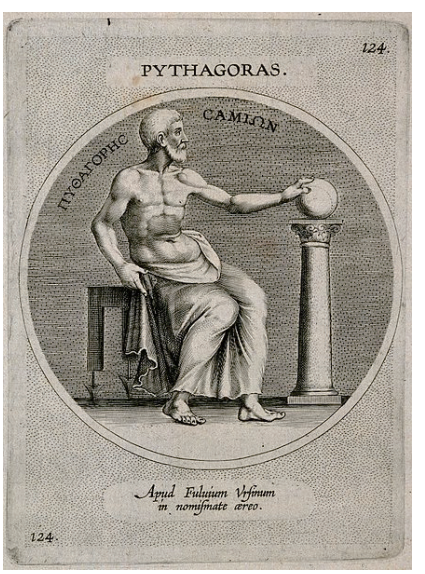

图 4 毕达哥拉斯 / 维基百科