HZ BOOKS
华章教育

计 算 机 科 学 丛 书

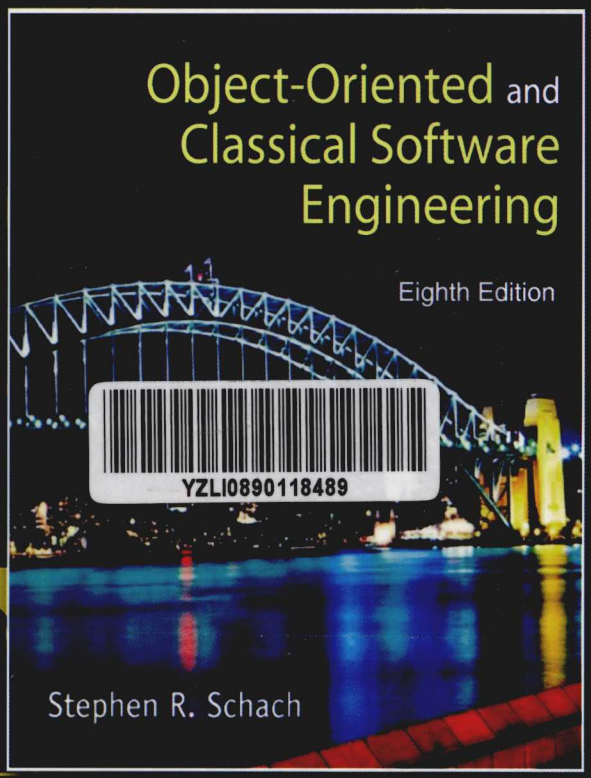
Mc Graw Hill

原书第8版

软件工程
面向对象和传统的方法

（美）Stephen R. Schach
范德比尔特大学 著 邓迎春 韩松 等译

Object-Oriented and Classical Software Engineering
Eighth Edition



机械工业出版社
China Machine Press

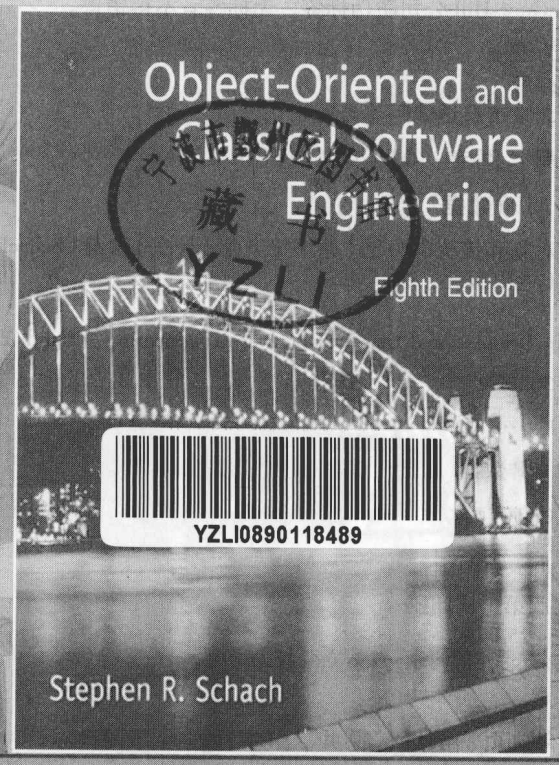
计 算 机 科 学 丛 书

原书第8版

软件工程
面向对象和传统的方法

(美) Stephen R. Schach 著 邓迎春 韩松 等译
范德比尔特大学

Object-Oriented and Classical Software Engineering
Eighth Edition



机械工业出版社
China Machine Press

本书是软件工程领域的经典著作，被加州大学伯克利分校等180多所美国高校选作教材。本书第8版继续保持了前七版的特色，采用传统方法与面向对象方法并重的方式，全面系统地介绍软件工程的理论与实践，并新增了第10章（第一部分的关键内容）和第18章（新兴技术）两章内容。全书分为两大部分，第一部分介绍软件工程概念，第二部分着重软件工程技术，教师可根据不同教学目的从任一部分开始讲授课程。

本书是高等院校软件工程课程的理想教材，同时也是专业软件开发人员和管理者的理想参考书。

Stephen R. Schach: Object-Oriented and Classical Software Engineering, Eighth Edition (ISBN 978-0-07-337618-9).

图书在版编目(CIP)数据

软件工程：面向对象和传统的方法(原书第8版)/(美)沙赫(Schach, S. R.)著；邓迎春等译. —北京：机械工业出版社，2011.12

(计算机科学丛书)

书名原文：Object-Oriented and Classical Software Engineering, Eighth Edition

ISBN 978-7-111-36273-9

Ⅰ. 软…　Ⅱ. ①沙…　②邓…　Ⅲ. 软件工程　Ⅳ. TP311.5

中国版本图书馆 CIP 数据核字(2011)第220660号

机械工业出版社(北京市西城区百万庄大街22号　邮政编码　100037)

责任编辑：刘立卿

北京瑞德印刷有限公司印刷

2012年1月第1版第1次印刷

185mm×260mm · 25印张

标准书号：ISBN 978-7-111-36273-9

定价：65.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

现在几乎每门计算机科学和计算机工程课都包含一个要求团队完成的软件开发项目。有时这个项目只持续一个学期（半年或三个月），但持续一年时间的团队软件开发项目越来越成为标准。

理想情况下，每个学生在学完一门软件工程课程后才开始进入基于团队的项目（“两阶段课程”）。然而实际上许多学生在学习软件工程课的中途即启动项目，甚至在课程开始时就启动项目（“并行课程”）。

下面将描述本书的结构，从中可以看出本书适用于上述两种情况。

第 8 版的结构

本书包含两大部分：第二部分指导学生如何开发软件产品；第一部分为第二部分提供必要的理论支持。18 章按如下结构组织：

	第 1 章	软件工程简介
第一部分	第 2 章至第 9 章	软件工程概念
第二部分	第 10 章至第 17 章	软件工程技术
	第 18 章	新兴技术

第 10 章为新增章节，概要介绍了第一部分的关键内容。如果采用两阶段课程，可先讲授第一部分，然后讲授第二部分（可跳过第 10 章，因为第 10 章的内容在第一部分中已经深入讲解过）。对于并行课程，教师会先讲授第二部分（这样学生可以尽早启动项目），然后再讲授第一部分，第 10 章的内容可帮助学生在没有学习第一部分的情况下理解第二部分。

后面的方法看起来不合常规：理论应在实践前学习，但事实上许多使用过本书第 7 版教材的教师在第一部分之前讲授第二部分的内容，他们这样做也收到了很好的效果。他们反映学生们在进行项目工作的过程中能够更好地理解第一部分的理论内容，也就是说，基于团队的项目工作使学生们更容易接受和理解软件工程基础的理论概念。

具体而言，第 8 版的内容可按以下两种方式讲授：

1. 两阶段课程

	第 1 章（软件工程简介）
第一部分	第 2 章至第 9 章（软件工程概念）
第二部分	第 10 章至第 17 章（软件工程技术）
	第 18 章（新兴技术）
	然后学生在接下来的学期（半年或三个月）里开展基于团队的项目

2. 并行课程

	第 1 章（软件工程简介）
	第 10 章（第一部分的关键内容）
	学生现在开始进行基于团队的项目，与第二部分内容的学习并行
第二部分	第 11 章至第 17 章（软件工程技术）
第一部分	第 2 章至第 9 章（软件工程概念）
	第 18 章（新兴技术）

第 8 版的新特性

1）对本书全面进行了更新。

2）增加了两章。如前面所述，一个新增章是第 10 章——第一部分的关键内容概述，这样学生可与软件工程课程并行地开始基于团队的项目。另一个新增章是第 18 章，概述了 10 个新兴技术，包括：

- 面向层面技术；
- 模型驱动技术；
- 基于组件技术；
- 面向服务技术；
- 社交计算；
- Web 工程；
- 云技术；
- Web 3.0；
- 计算机安全；
- 模型检查。

3）第 8 章扩充了设计模式方面的内容，包括新扩充了一个小型实例研究。

4）第 5 章增加了两个理论工具：分治和关注分离。

5）第 13 章中电梯问题的面向对象分析体现了现代分布式的分散结构。

6）围绕当前研究的重点，广泛更新了参考文献。

7）新增了 100 多道习题。

8）新增了一些“如果你想知道”内容。

继承第 7 版的特性

- 统一过程仍是面向对象软件开发方法的首选。因此，贯穿全书的仍是统一过程的理论和实践。
- 第 1 章深入分析面向对象范型的优势。
- 在第 2 章，尽可能早地引入迭代 - 递增生命周期模型。进一步地，与前面的所有版本一样，对其他的生命周期模型进行描述、比较和对比，并特别关注敏捷过程。
- 在第 3 章（“软件过程”）中，介绍工作流（活动）和统一过程的各个阶段，并解释二维生命周期模型的需求。
- 第 4 章（“软件小组”）讨论组织软件小组的多种方式，包括开发敏捷过程的小组和开发开源软件的小组。
- 第 5 章（“软件工程工具”）包含一些 CASE 工具中重要的类的信息。
- 第 6 章（“测试”）着重讨论连续测试的重要性。
- 对象仍旧是第 7 章（“从模块到对象”）关注的焦点。
- 设计模式仍是第 8 章（“可重用性和可移植性”）的核心。
- 软件项目管理计划的 IEEE 标准在第 9 章（“计划和估算”）中再次提供。
- 第 11 章（“需求”）、第 13 章（“面向对象分析”）和第 14 章（“设计”）大都致力于阐述统

一过程的工作流（活动）。基于显然的理由，第12章（“传统的分析”）大体没有进行修改。

- 第15章（“实现”）中的内容明确区分实现和集成。
- 第16章重点描述交付后维护的重要性。
- 第17章为准备在软件业中就业的学生提供了UML方面的额外材料。这一章特别适用于采用本书作为两学期软件工程课程系列教材的指导教师们。在第二个学期，除了开发基于小组的学期项目或顶石（capstone）项目以外，学生还可以获得UML的额外知识。
- 与以前一样，有两个运行实例研究，即使用统一过程开发的MSG基金实例研究和电梯问题实例研究。同往常一样，Java和C ++实现在www. mhhe. com/schach在线可用。
- 除了使用这两个运行实例研究阐述完整的生命周期外，还通过8个小实例研究来突出专门的主题，例如移动目标问题、逐步求精、设计模式和交付后维护。
- 在先前的所有版本中，我强调文档、维护、重用、可移植性、测试和CASE工具的重要性。在本版中，所有这些概念都无疑受到同等程度的重视。如果学生不理解这些软件工程基础知识的重要性，教授学生们最新的思想就没有什么用处了。
- 如同第7版，对以下几方面给予特殊的重视：面向对象生命周期模型、面向对象分析、面向对象设计、面向对象范型的管理含义、面向对象软件的测试和维护，其中还包括面向对象范型的度量。此外，对对象作了许多更简明的注解，有的是一个段落，有的只是一句话。这样做的原因是，面向对象范型不仅与各种阶段如何执行有关，也影响着我们思考软件工程的方式。对象技术再次贯穿本书始终。
- 软件过程仍然是整体贯穿本书的一个概念。为了控制这个过程，我们必须能够测量项目中发生了什么。相应地，继续保留对度量的强调。关于过程改进，保留有关能力成熟度模型（CMM）、ISO/IEC 15504（SPICE）以及ISO/IEC 12207内容。
- 本书仍然与计算机语言无关，少量代码实例用C ++或Java表示，而且我尽量减少与语言有关的细节，确保代码实例对于C ++和Java用户同样清晰。例如，不使用`cout`表示C ++的输出，也没有使用`System. out. println`表示Java的输出，而使用伪码指令*print*。（一种例外情况是新的实例研究，其完整的实现细节用C ++和Java同时给出。）
- 像在第7版中一样，本书包含600多个参考文献。我选择了当前的研究文章以及一些仍保持有关最新信息的经典的文章和书籍。毫无疑问，软件工程是一个快速发展的领域，学生需要知道最新的成果以及在哪些文献里可以找到它们。与此同时，今天的前沿研究是在昨天的事实基础上进行的，没有理由将一篇较早的参考文献排除在外，如果它的思想在今天如最初一样仍在应用着。
- 本书假设读者对诸如C、C#、C ++或Java的一种高级编程语言很熟悉，另外，读者应学习过数据结构。

为什么仍然包括经典范型

现在几乎一致认为面向对象范型比经典范型优越。相应地，许多选用了本书第7版的教师只选择该书中面向对象方面的内容进行讲授。然而，当问起这些教师们的意见时，教师们指出，他们更倾向于选择包含有经典范型内容的教材。

原因在于，尽管越来越多的教师只“讲授”面向对象范型，但他们仍旧愿意在课堂上“提到”经典范型；许多面向对象技术难以理解，除非学生们对演化出这些面向对象技术的经典技术有一些认识。例如，如果学生对实体关系建模有所了解，即使是肤浅的了解，也会更容易理解实体类建模。类似地，对有穷状态机的简要介绍会使教师更容易讲授状态图。因此，我在第8版中保留了经典方面的内容，这样教师们在教学中就有可用的经典材料。

习题集

与第7版一样，本书有5种类型的习题。首先，在第11、13和14章结束时有运行着的面向对象

分析和设计项目。之所以包含这些项目，是因为学习如何执行需求、分析和设计工作流的唯一途径来自于广泛的实践。

第二，每一章结尾包含一些意在突出重点的练习。这些练习是独立的，全部练习的技术信息都可以在本书中找到。

第三，有一个软件学期设计项目。该设计需要由最少三人组成的小组协作完成，而不是通过常规的电话协商完成。学期设计项目由15个独立的组件组成，每个组件都附在相应的章后。例如，“设计”是第14章的主题，因此在该章中学期设计的组件与软件设计有关。通过将一个大的项目分解为小的、明确定义的几个部分，教师能够更密切地掌控班上的学习进度。学期设计项目是这样一种结构：指导教师能够自由地将这15个组件应用于任何其他项目。

本书是为研究生和高年级本科生而编写的，第4种类型的习题是根据软件工程文献中的研究报告拟制的。在每一章都选择了一篇重要的文章，尽可能选择一篇与面向对象软件工程有关的文章。要求学生阅读该篇文章并回答与其内容有关的一个问题。当然，教师可以自由安排任何其他研究文献，在每一章后的“进一步阅读指导”中包含各种相关论文。

第5种类型的习题与实例研究有关。这类习题首先在第3版中引入，是应许多教师的要求加入的。教师们感觉：学生们通过修改一个现成的产品而不是从头开发一个新的产品可以学到更多的东西。业界的许多高级软件工程师同意这个观点。基于此，给出实例研究的每一章有需要学生在某种程度上修改该实例的问题。例如，在某一章中，要求学生使用一项与该实例研究中使用的不同的设计技术重新设计该实例。在另一章中，要求学生回答以不同的顺序执行面向对象分析的步骤会产生什么不同的效果。为了使学生易于修改实例研究的源代码，在万维网上提供这些源代码，网址是：www. mhhe. com/ schach。

该网站还提供给教师一个完整的电子课件和包括学期项目在内的所有习题的详解。

有关 UML 的材料

本书实际使用统一建模语言（UML）。如果学生没有UML的前期知识，可以用两种方式教授这部分。我倾向于在需要时才教授UML，也就是说，每个UML概念只在需要它之前讲解。下表描述了本书使用的UML结构所在的章节。

结　构	介绍对应的UML图的章节
类图、注解、继承（泛化）、聚合、关联、导航三角形	7.7节
用例	11.4.3节
用例图、用例描述	11.7节
构造型	13.1节
状态图	13.6节
交互图（顺序图、通信图）	13.15节

另一方面，第17章介绍UML，包括本书当前及以后所需的材料。第17章可以在任何时间讲授，它不需要依赖前16章。第17章涵盖的主题如下表所示。

结　构	介绍对应的UML图的章节
类图、聚合、多重性、组合、泛化、关联	17.2节
注解	17.3节
用例图	17.4节
构造型	17.5节
交互图	17.6节
状态图	17.7节
活动图	17.8节
包	17.9节
组件图	17.10节
部署图	17.11节

在线资源

本教材的支持网站：www. mhhe. com/schach 上有供学生使用的 MSG 实例研究的 Java 和 C ++ 实现和源代码，以及供教师使用的课件、所有练习和学期项目的详解及图库。有关具体的细节请联络销售代表[⊖]。

致谢

我非常感谢前面 7 版书的审阅者，他们给予了建设性意见和许多有帮助的建议。特别感谢本版书的审阅者，他们是：

Ramzi Bualuan
University of Notre Dame

Ruth Dameron
University of Colorado, Boulder

Werner Krandick
Drexel University

Taehyung Wang
California State University, Northridge

Jie Wei
City University of New York—City College

Mike McCracken
Georgia Institute of Technology

Nenad Medvidovic
University of Southern California

Saeed Monemi
California Polytechnic University, Pomona

Xiaojun Qi
Utah State University

关于出版商 McGraw-Hill，我特别感谢样本编辑 Kevin Campbell 和设计师 Brenda Rolwes，我还特别感谢 Montage 工作室的 Melissa Welch，她将悉尼港大桥夜景图片转化为极美的封面。

还要感谢 Jean Naudé（Vaal University of Technology，Secunda Campus）与我合作完成教师用的题解手册，特别是 Jean 提供了学期项目的详细解答，包括它的 Java 和 C ++ 实现。在编写题解手册的过程中，Jean 提出了大量建设性建议来完善本书，对此我深表谢意。

最后，我感谢我的妻子 Sharon 一如既往地支持和鼓励我。与我以前所有书的情况一样，我尽力使尽家庭义务优先于写作，然而当最后期限临近时，这总是不可能的。在这种时候，Sharon 总是理解我，为此我特别感激她。

我愿把我的这第 15 本专著献给我的爱孙 Jackson 和 Mikaela。

Stephen R. Schach

⊖ 麦格劳－希尔教育出版公司服务热线：800-810-1936。E-mail：instructorchina@ mcgrawhill. com。——编辑注

目 录

Object-Oriented and Classical Software Engineering, 8E

第二部分 软件生命周期的工作流

附 录

软件工程的范畴

学习目标

- 明确软件工程意味着什么；
- 描述传统软件工程生命周期模型；
- 解释为什么面向对象范型现在被广泛接受；
- 讨论软件工程的各个方面的含义；
- 区分关于维护的传统和现代的观点；
- 讨论连续的计划、测试和文档编制的重要性；
- 感受遵守道德规范的重要性。

一个有名的故事讲的是一个主管有一天收到了一份计算机生成的账单，账单的金额为0.00美元。这人与朋友一起大大地嘲笑了一下“愚蠢的计算机”，然后将账单扔掉了。一个月之后，收到了一份同样的账单，上面标着已过了30天。然后，第三张账单来了。又一个月之后，第四张账单也来了，同时带来一个通知，暗示说如果不能及时付清这个0.00美元的账单将可能会采取法律行动。

第五张账单来的时候标着时间已过了120天，这张账单没有任何暗示，它粗鲁、直白地威胁道：这张账单如不能立刻付清将要采取所有法律手段。这个主管担心自己公司的贷款（信用）利率会受这个疯狂的机器的影响，于是找了一个软件工程师朋友，跟他讲了这件恼人的事情。这位工程师克制住不让自己发笑，他让主管邮了一张0.00美元的支票。该做法取得了预期的效果，几天后一张0.00美元的收据寄到了，这个主管小心翼翼地收好了这张收据以防将来计算机声称那张0.00美元的账单还没有支付。

这个有名的故事有一个不太为人知晓的结局。几天后，这个主管被他的财务经理叫去了，这个财务经理拿着一张0.00美元的支票问：“这是你的支票吗?”

这个主管回答说：“是。”

“那你能告诉我为什么要写一张0.00美元的支票吗?”财务经理问。

这样，整个故事又被重复一遍。当这个主管说完的时候，经理转向他，平静到地问“你能说说你付0.00美元的账单对我们的计算机系统有什么影响吗?”

计算机专业人士虽然会感到一点点窘迫，但还是会对此故事感到可笑。毕竟在我们任何一个人设计或完成一个产品的初样阶段，是可能出现类似催讨0.00美元账单的情况的。到目前为止，我们在测试中总是能够发现此类错误。但是我们的笑声中有一种恐惧感，因为我们内心深处担心有一天，在我们已经将产品交付给顾客的时候还没有发现这些问题。

1979年11月9日检测到一个绝对称不上幽默的软件错误。美国战略防空司令部收到由全球军事指挥控制系统（WWMCCS）计算机网络发出的警报，引起了混乱，警报报告苏联已经向美国发射导弹[Neumann，1980]。实际上把模拟演习当成了真的，就像5年后电影《War Games》里所演的那样。虽然美国国防部可以理解地没有详细说明把实验数据当成真实数据的确切理由，但看起来很可能是软件错误导致的；或者是系统作为一个整体在设计上没有区分模拟和真实；或者是用户界面没有设置必要的检查来确保系统的终端用户从虚假中分辨出事实。换句话说，软件错误，如果该问题确实是由软件引起的，可能会给我们的文明社会带来不愉快和灾难性的结局（有关由软件错误所导致的灾难方面的信

息，请见“如果你想知道［1-1］”部分）。

无论我们正在处理的是账单还是防空任务，我们的许多软件都推迟交付时间、超出预算、带有残存的错误，并且不满足用户要求。软件工程试图解决这些问题，换言之，**软件工程**是一门学科，目的是生产出没有错误的软件，按时并且在预算内交付，满足用户的需求。更进一步，当用户的需求改变时，软件必须易于修改。

软件工程的范畴非常广。软件工程的某些方面可以归入数学或计算机科学；其他方面可以落入经济学、管理学或心理学的范畴。为了展示软件工程所触及的宽广领域，我们将从五个方面来进行考查。

如果你想知道［1-1］

在WWMCCS网络的情形下，灾难在最后一分钟内得以避免。然而，有些软件错误却会导致悲剧。例如，在1985年和1987年之间，至少有两个病人死于由Therac-25医用线性加速器产生的严重过量辐射［Leveson and Turner，1993］，原因是控制软件中的一个错误。

在1991年的海湾战争中，一枚飞毛腿导弹穿越了爱国者反导弹防御措施，击中了沙特阿拉伯的Dhahran附近的一个兵营，导致28名美国人死亡，98人受伤。爱国者导弹的软件中包含了一个累积的定时错误，爱国者导弹设计成每次只能工作几个小时，过了这个时间之后，时钟就要复位。结果是，这个错误从未有过明显的影响，从而没有被检测到。然而，在海湾战争中，爱国者导弹的电池在Dhahran连续工作了100小时以上，这引起的累积时间误差大得足以导致系统的不精确。

在海湾战争中，美国用船运送爱国者导弹到以色列，以保护其免受飞毛腿导弹的攻击。以色列的军队仅在8个小时后就察觉到了这个定时问题，并立刻向美国的制造商报告了这个问题，制造商尽快地纠正了这个问题。然而悲惨的是，新软件在飞毛腿导弹直接击中兵营的第二天才到达［Mellor，1994］。

幸运的是，由软件错误造成的死亡或严重伤害非常少。然而，一个错误可能会给成千上万的人带来重大影响。例如，在2003年2月，一个软件错误使美国财政部发出50 000张没有受益者名字的社会保险支票，因此这些支票无法储蓄或兑现［St. Petersburg Times Online，2003］。在2003年4月，借款者被SLM公司（通常称为Sallie Mae公司）告知，他们的助学贷款由于计算机软件问题从1992年起就被计算错了，该问题直至2002年底才被检测出来。大约100万借款人被告知他们需要支付更多的还款，从而在形式上，或者每月支付更多的还款额，或者在当初10年的还款期限之外，再继续花时间支付多出来的还款利息［GJSentinel.com，2003］。这两个错误都很快纠正了，但是它们对大约100万人产生了很大的经济上的影响。

比利时政府高估了2007年的预算，该预算达到8.83亿欧元（时值超过11亿美元）。这个错误由一个软件缺陷以及手动跳过一个检错机制共同引起［La Libre Online，2007a；2007b］。比利时税务专家使用扫描仪和光学特征识别软件来处理纳税申报单。如果该软件遇到一个不可读的申报单，它将纳税人的收入记录为99 999 999.99欧元（超过1.25亿美元）。大概是因为99 999 999.99欧元这个“神奇的数”可以被数据处理部门的雇员很快地检测到，这样这种有问题的申报单就可以手动处理。当为了评估税额而分析这些纳税申报单时软件工作正常，但当为了预算的目的再次分析这些纳税申报单时软件就出问题了。颇为讽刺的是，该软件确实有过滤器可以检测到这类问题，但为了加快处理速度，这些过滤器被人为地屏蔽了。

该软件至少有两个缺陷。首先，软件工程师假设在对数据进行进一步处理前总会有精确的人工详细审查。其次，软件允许过滤器被人工屏蔽。

1.1 历史方面

发电机会出现问题，这是事实，但是，它比工资报表软件产品出问题的几率小得多；桥梁有时候会倒塌，但是，比操作系统崩溃的可能性小得多。在软件设计、实现和维护应当与传统的工程学科具有同等地位的观念驱使下，一个NATO研究小组在1967年创造了软件工程的概念。软件的开发应当同