# 软件行业读后感范文精选53篇

来源：网络 作者：独坐青楼 更新时间：2025-04-17

*软件行业读后感范文 第一篇>1、广联达预算软件操作流程工程的实际状况是：建面为6603平方，有两层地下室，负一层地下室层高为6米，负二层地下室层高为4米，第一层层高是4米，标准层的层高为3米，一共有10层。工程框架结构而言，主体部分是钢筋砼...*

**软件行业读后感范文 第一篇**

>1、广联达预算软件操作流程

工程的实际状况是：建面为6603平方，有两层地下室，负一层地下室层高为6米，负二层地下室层高为4米，第一层层高是4米，标准层的层高为3米，一共有10层。工程框架结构而言，主体部分是钢筋砼，其次还涉及到筏板、框架柱、剪力墙、梁板等。

>2、广联达预算软件操作心得

第一，首要工作就是对图纸进行深入了解，对钢筋工程进行分析与研究，从而产生以软件处理钢筋工程的思想，特别是不符合常理的结构与构造需要引起重视，如此一来才能够节省大量时间与精力，实现良好效果。第二，施工期间需要纵观全局，慎重考虑到各个方面的影响因素，尤其是图形、钢筋等方面，负责人要统筹全局，兼顾多个方面的工作，谨防出现重复劳动的情形，若难以顾及到多个方面的工作，则要做到两害相权取其轻，结合实际情况制定出科学合理的施工方案。综上所述，以最少投入获得最大效益。第三，施工期间要充分利用各类软件，了解各类软件所具备的功能，例如偏移、镜像、复制、块操作等。第四，构件建设过程中要对各方面因素进行考量，尤其是构件名称必须要准确、详细、便于查找。构件定义之后还要套取清单与定额，谨防出现错套或者是遗漏等情形。第五，频繁使用的构件主要以构件做法存档或者是构件存档等形式进行存储，便于下次继续使用。以图集为例，要依据图集编号进行存储，结合实际情况选择最合适的存储方式。第六，各类软件都只是辅助工具，构件输入也只能看做是一种手段，要充分利用这些器具与方式获得各类数据信息，这才是我们的最终目的。所以，实践工作中要灵活应用各类软件，从而获取所需信息。

总之，以往不少行业都采用人工编制预算的模式，但在工程量计算精度要求愈来愈大的趋势下，人工计算早已无法满足工作需求，广联达预算软件顺利成章成为一个高效的工具。但我们必须清楚知道，使用软件的目的是要提升工作效率，不能够完全对软件产生依赖，任何预算软件在运用中都有一个持续完善的过程，任何个人学习、操作软件也是一个循序渐进的过程，我们只有真正实现活学活用，针对现实情况操作软件，在实践中不断发掘软件功能，才可促使广联达软件功能不断延伸，效果不断提高。

**软件行业读后感范文 第二篇**

今天的实训结束了，今天做的是纸牌游戏软件和趣味打字游戏。今天的东西对我来说有点难度，最后没有能过完全做完。但是我还是觉得这是一个不错的实训，在这种集体的环境里和同学们一起学习，每天的生活过的也是非常的充实。

此次实践课我的收获很多。我和同学们这一次真正自己动手制作了一个小软件，虽然还存在很多的问题，而且我做的软件在使用起来还是很不可行的，但是我们从中受到了很多知识，不仅是专业的知识，更让我明白了一个软件从设计到实现的每一个环节真的很不容易，不仅需要扎实的专业知识，更需要一个团队的配合，这才是一个软件成功的关键。这就告诉我们，一个人的出色不算什么，一个团队的出色才是真正有用的。

刚开始拿到题目我们组员都不知如何下手，经过小组成员一起查找资料，并且开会讨论，我们确定了设计的设计目标以及具体实现方式，包括如何将Java的思想运用到实际系统的详细设计之中。

在实验课上，我学会了很多学习的方法。而这是日后最实用的。要面对社会的挑战，只有不断的学习、实践，再学习、再实践。这对于我的将来也有很大的帮助。以后，不管有多苦，我想我都能变苦为乐，找寻有趣的事情，发现其中珍贵的事情。就像中国提倡的艰苦奋斗一样，我都可以在实验结束之后变的更加成熟，会面对需要面对的事情，以及学会遇到问题，不急不慌，慢慢解决它。

虽然过程辛苦是不可避免，但收获还是令人感到尤其的欣慰。在这次的软件设计中不仅检验了我所学习的知识，也培养了我的实践能力，让我知道遇到一个问题，如何去寻找思路，如何去解决问题，最终完成整个事情。在设计过程中，与同学分工设计，和同学们相互探讨，相互学习，相互监督。学会了合作，学会了宽容，学会了理解，也学会了做人与处世。课程设计是我们专业课程知识综合应用的实践训练，是我们迈向社会，从事职业工作前一个必不少的过程。实验过程中，也十分感谢实验指导老师陈中育老师的指点与教导。这次软件设计不仅是对这学期所学知识的一种综合检验，而且也是对自己动手能力的一种提高，增强了自己实践能力。通过这次课程设计使我明白了自己知识还比较欠缺，只是学习书本知识还是远远不够的，自己不会的东西还有太多，学习需要自己长期的积累，在以后的学习、工作中都应该不断的学习，将课本的理论知识与生活中的实践知识相结合，不断提高自己文化知识和实践能力。

**软件行业读后感范文 第三篇**

我实习的时候对于金蝶K3有不少认识：首先。做会计需要理清思路，像在建账的时候，要对整个建账的轮廓有较清晰地了解，这样具体工作起来有方向，效率比较高，否则很容易在一个问题上卡住。而理清思路后就是具体的一些工作，这些工作或许会重复，比较枯燥，很需要耐心。最好在开始项目之前，完成前期数据准备。对于一些基础数据，例如各种基本资料的编码。我们有必要提前完成，免得拉长项目实施周期。金蝶K3使用前，必须设臵好基础资料，基础资料是运行K3系统的基石，有点象做房子前要准备好材料一样。

基础资料的准备顺序基本上就是系统默认的顺序。这个顺序没有什么特别的，怎么着都行，你在需要时自动提示你需要先处理哪一些。不过我还是要将一般处理顺序说明一下，最好是：币别、凭证字、计量单位、结算方式、仓库、仓位、部门、职员、科目、客户、供应商、物料。一些项目是有必录项与非必录项的，在前期我们可以只录入必录项，以后在需要时再录入非必录信息。我原以为仓库与仓位是一多对应的，后来在操作中体会到，两者是可以任意组合的。

在设立仓位时，建议管理员到演示帐套中看一看两者的区别与联系，仓库是在单据表头上的，仓位是在表体的，只有设了仓位管理的才需录入仓位。由于上述特点，可以将仓位管理虚拟出来进行一些其他的使用。在客户、供应商、部门、职员、物料等中，其代码可以随时更改，在更改代码后，其原有信息自动过到新代码之中。二是同一核算项目的明细项尽可能的在级次和长度上保持一致。

科目代码的更改有自身的特点，其他项目的代码随时都可以更改，科目的代码的更改就不一样了。在科目使用后，其一级代码可以改动，除此外都不能改动。设臵科目要了解哪些属性是用后可以更改的，哪些是不能更改的。如上述代码除一级外都不能更改，同样不能更改的有科目类别、数量金额核算、是否为现金科目和银行存款科目、是否出日记帐、余额方向、计量单位和所挂的核算项目。别的不易反复，最易反复的是科目所挂的核算项目，如原材料挂了物料这一核算项目，后来又不想挂了，只在存货核算中看明细就行了等等

**软件行业读后感范文 第四篇**

学习了这门课程, 还有老师们的多元化教课,不但让我从理论上掌握软件工程,还有从不同的实例,让理论和实践得到了很好的结合。整一个学期下来，总的来说还是学到了很多东西的，有很多地方是值得肯定的，其实在我看来，软件工程与其说是一门课程，不如说是一门思想。是一个如何去分析和处理问题的过程，应该说其范畴已经远远不止局限于该门课程，成为了一个综合的一个能够解决问题的思想集合。

整本书的内容逻辑很清晰明了，由浅入深循序渐进，首先我就大概描述下我们所学的内容，第一章是从整体分析软件工程这门学科的发展和所处的社会环境，接着后面的几章深入分析了软件开放过程和模式、软件项目管理、计算机工程、需求分析、结构化分析建模以及基于UML面向对象分析建模等。接着我就详细介绍下我对这门课程知识点的理解概括：

软件：软件是能够完成预定功能和性能的可执行的计算机程序和使程序正常执行所需要的数据，加上描述程序的操作和使用的文档。

>软件的特征：

①软件是一种逻辑实体，而不是具体的物理实体，因而它具有抽象性。

②软件是通过人们的智力活动，把知识与技术转化成信息的一种产品。

③软件成为产品后，其生产只是简单的拷贝，不同于硬件制造。

④维护过程比硬件复杂的多，甚至会引发新的错误。软件危机：指的是软件开发和维护过程中遇到的一系列严重问题。

>出现软件危机的原因：

①软件维护费用急剧上升，直接威胁计算机应用的扩大。

②软件生产技术进步缓慢。软件工程是指导计算机软件开发和维护的工程学科。 软件生存周期：一个软件从定义到开发、使用和维护，直到最终被弃用，要经历一个漫长的时期，通常把软件经历的这个漫长的时期称为生存周期。

>软件的生存周期可分为八个阶段：

①问题定义；

②可行性研究；

③需求分析；

④总体（概要）设计；

⑤详细设计；

⑥编码与单元测试；

⑦综合测试；

⑧软件维护；

瀑布模式：是传统的软件开发模式，其中的“瀑布”是对这个模式的形象表达，由山顶倾泻下来的水，自顶向下、逐渐细化。其特点是：线性化过程；分为分析、设计、编码、集成等几个阶段，并且各阶段逐级推进，不允许跨越。里程碑管理；阶段评审；文档驱动；简洁便于工程应用的线性化过程步骤，并可以通过里程碑管理机制而使项目进程量化。其明显的优点就是没个阶段结束前都要对所完成的阶段成果进行评审，这使得软件的错误能够在个阶段内尽早发现并尽早解决，总的来说瀑布模式具有良好的质量保证机制，有很强的生命力。

原型进化模式：对软件进行直接模拟或仿真，只需要分析需求框架后进行原型创建，再对原型系统进行逐步细化与完善，通过版本更新逐步满足用户对于软件的多方面需要。

增量模式：开发过程有三个任务域，分别是设计结构、开发构件和集成系统，它既有完善的工程管理机制，又能适应用户需求变更，有利于质量的监控，并且各局部基于构件构造，有利于逐步构建与完善；由于先交付核心构件可利于降低项目的技术风险。

螺旋模式：是一种可较好的规避开发风险过程的模式，项目是基于任务的螺旋式推进，每个螺旋由内之外分别是需求分析、软件设计、系统集成、验证与交付。

>软件开发的整个过程：

①需要项目团队，组建优秀的团队可以开发出更搞质量的软件产品。任务开发团队要求小而精，成员大多在8人以内，主要成员有项目负责人、开发人员、资料管理员和软件测试员。

**软件行业读后感范文 第五篇**

对于学习软件工程这门课程，我认为有许多东西要学习。其实在我看来学习这门课程的精髓是学习一种方法。是一个如何去分析和处理问题的过程，应该说其范畴已经远远不止局限于该门课程，成为了一个综合的一个能够解决问题的思想集合。读完软件工程案例教程这本书，我觉得自己受益匪浅。

整本书的内容逻辑很清晰明了，由浅入深循序渐进，首先我就大概描述下我们所学的内容，第一章是从整体分析软件工程这门学科的发展和所处的社会环境，接着后面的几章深入分析了软件开放过程和模式、软件项目管理、计算机工程、需求分析、结构化分析建模以及基于UML面向对象分析建模和测试等。 对于这本书我主要对需求分析和测试比较感兴趣，在这我要着重的谈一些自己的心得体会以及自己的看法。

>1.需求分析

需求分析的重要性

一款成功的软件是建立在成功的需求分析之上的，而高质量的需求来源于用户与开发人员之间有效的沟通与合作。当用户有一个问题可以用计算机系统来解决，而开发人员开始帮助用户解决这个问题，沟通就开始了。由此我们可以看出需求分析的重要性。

需求获取可能是最困难、最关键、最易出错及最需要沟通交流的活动。对需求的获取往往有错误的认识：用户知道需求是什么，我们所要做的就是和他们交谈从他们那里得到需求，只要问用户系统的目标特征，什么是要完成的，什么样的系统能适合商业需要就可以了，但是实际上需求获取并不是想象的这样简单，这条沟通之路布满了荆棘。首先需求获取要定义问题范围，系统的边界往往是很难明确的，用户不了解技术实现的细节，这样造成了系统目标的混淆。

其次是对问题的理解，用户对计算机系统的能力和限制缺乏了解，任何一个系统都会有很多的用户或者不同类型的用户，每个用户只知道自己需要的系统，而不知道系统的整体情况，他们不知道系统作为一个整体怎么样工作效率更好，也不太清楚那些工作可以交给软件完成，他们不清楚需求是什么，或者说如何以一种精确的方式来描述需求，他们需要开发人员的协助和指导，但是用户与开发人员之间的交流很容易出现障碍，忽略了那些被认为是xxx很明显xxx的信息。最后是需求的确认，因为需求的不稳定性往往随着时间的推移产生变动，使之难以确认。为了克服以上的问题，必须有组织的执行需求的获取活动。

需求分析的原则

（1）需求分析必须能够表达和理解问题的数据域和功能域。数据域包括数据流、数据内容和数据结构，而功能域反映上述 3 方面的控制信息。

（2）需求分析要把一个复杂问题按功能进行分解并逐层细化。通常，软件系统要处理的问题如果太大、太复杂就很难理解，若划分成几部分，并确定各部分间的接口，就可完成整体的功能。在需求分析过程中，软件系统的用户需求中的数据、功能和行为都应细化。

（3）需求建模。模型可以帮助系统分析人员更好地理解软件系统的数据、功能和行为，这些模型是软件工程中下一阶段进行系统设计的基础。

需求分析的注意事项

（1）确定详细的需求，否则经费就算不准。经费估计错误的原因多为：用户需求频繁变动、遗漏重要需求、与用户交流不够、需求规格说明书质量低劣、需求分析不充分等。

（2）在编写需求规格说明书之前，应明确要解决的问题。在试图解决问题之前，要保证已考察了全部可替代的方案。要搞清哪地方有问题，真正的问题出在哪里。这样，在编写需求规格说明书时做到有的放矢，把存在的问题暴露出来。

（3）立即确定需求，并记录下该需求的背景。没有明确问题，就进行下一步的设计，想回避矛盾，可能会带来更大的问题。用户不确定需求，软件设计人员自己决定需求，将会带来严重的问题。为了避免将来可能出现的问题和软件工程项目能够尽快地进入到下一个阶段的系统设计中，要尽可能迅速地把用户需求确定下来。任何决定总比没有决定要好。

（4）一旦在需求规格说明书中发现问题，立即改正。如果把存在的问题拖延到系统设计阶段去改正，就可能要花数倍的时间和精力才能纠正同一错误。

（5）在众多用户需求中确定各个需求的优先顺序，并确定可能存在的子集，以便为软件设计、实施和项目管理等后续阶段提供有利条件。

（6）需求分析时，不要进行系统设计的工作。需求分析的主要目的是确定软件系统的外部特征，充分反映软件系统应有的面貌，便于让软件设计人员根据用户需求，去全面地考虑软件系统的体系结构、算法等。在需求分析阶段要集中精力解决用户需求存在的问题，尽可能避免产生遗留问题。

（7）对于复杂的软件系统，要从多种视角进行需求分析。根据软件系统的本质，切合实际地组织多种视角的需求。例如，可从根据用户的类型，或根据响应的类型，或根据对象的软件工程案例教程类型，或根据系统的模式等视角来组织用户需求。通过多个视角来研究用户需求问题，把可得到的不同的“投影”组合起来形成完整系统的描述。当试图从整体观点来描述软件系统发生困难，或者有可能发生错误，或者很有可能遗失软件系统的某些特性。而从不同的视角来 描述软件系统，因为每个视角限制了研究的范围并能够将注意力集中于此，所以很容易保证所研究的问题是真正完整的。

（8）重视形式化方法，但不放弃自然语言。为了用户需求表达的精确性和方便用户的可理解性，一个好方法是把自然语言的表达与形式化规格说明并立，互相对照，而且在一般情况下，先用自然语言写出，再给出它的形式模型。

（9）用户需求中不应存在“待确定”的条款。如若有这种需要，应同时说明：何时由谁来解决该问题。

用户需求的类型

需求分析是从用户最初的非形式化需求到满足用户要求的软件产品的映射过程。它实际上是一个对用户意图不断进行揭示和判断的过程，其目的在于细化、精化软件的作用范围，确定拟开发软件的功能和性能、约束、环境等。可将用户的需求分为两大类：功能性需求和非功能性需求。

（1）功能性需求。功能性需求主要说明了系统各功能部件与环境之间的相互作用的本质，即拟开发软件在职能上实际应做到什么。一般来说，它是用户最主要的需求，通常包括系统的输入、系统能完成的功能、系统的输出以及其他反应。在功能性需求中还应包括备选功能的定义识别。

（2）非功能性需求。非功能性要求主要从各个角度对所考虑的可能的解决方案起约束和限制作用。

需求分析的方法

在软件工程中，常用的需求分析方法有面向数据流的结构化分析方法（简称 SA）和面向对象的分析方法（简称 OOA）。此外，还有以用户为中心的需求分析方法。这些方法都采用图文结合的方式，可以直观地描述软件的逻辑模型。这里仅介绍结构化分析方法和以用户为中心的需求分析方法。

>2.软件测试

软件测试概述

软件本身无形态，它是复杂的知识高度密集的逻辑产品，其中不可能没有错误。软件实施工程过程中必须伴随着软件质量保证的活动，而软件测试是主要活动之一。在开发软件的过程中，人们使用了许多保证软件质量的方法分析、设计和实现软件，但难免还会在工作中犯错误。这样，在软件产品中就会隐藏许多错误和缺陷。对于规模大、复杂性高的软件更是如此。在这些错误中，有些是致命的错误，如果不排除，就会导致生命与财产的重大损失。

软件测试的目的

测试的目的是“说明程序能正确地执行应有的功能”，还是“表明程序没有错误”？基于不同的立场，存在着两种完全不同的测试目的。从用户的角度出发，普遍希望通过软件测试暴露软件中隐藏的错误和缺陷，以考虑是否可以接受该产品。而从软件开发者的角度出发，则希望测试成为表明软件产品中不存在错误的过程，验证该软件已正确地实现了用户的要求，确立人们对软件质量的信心。因此，他们会选择那些导致程效概率小的测试用例，回避那些易于暴露程序错误的测试用例。同时，也不会刻意去检测、排除程序中可能包含的副作用。显然，这样的测试对完善和提高软件质量毫无价值。因为在程序中往往存在着许多预料不到的问题，可能会被疏漏，许多隐藏的错误只有在特定的环境下才可能暴露出来。如果不把着眼点放在尽可能查找错误这样一个基础上，这些隐藏的错误和缺陷就查不出来，会遗留到运行阶段中去。如果站在用户的角度，替他们设想，就应当把测试活动的目标对准揭露程序中存在的错误。在选取测试用例时，考虑那些易于发现程序错误的数据。

软件测试的原则

（1）应当把“尽早地和不断地进行软件测试”作为软件开发者的座右铭。由于原始问题的复杂性、软件的复杂性和抽象性、软件开发各个阶段工作的多样性，以及参加开发各种层次人员之间工作的配合关系等因素，使得开发的每个环节都可能产生错误。所以不应把软件测试仅仅看成是软件开发的一个独立阶段，

而应当把它贯穿到软件开发的各个阶段中。在需求分析阶段就应该制订测试计划，以保证每个需求，每个设计单元都是可测试的，便于测试。坚持在软件开发的各个阶段的技术评审，这样才能在开发过程中尽早发现和预防错误，把出现的错误克服在早期，杜绝某些隐患，提高软件质量。

（2）测试用例应由测试输入数据和与之对应的预期输出结果这两部分组成。测试以前应当根据测试的要求，选择在测试过程中使用的测试用例（Test Case）。测试用例主要用来检验程序员编制的程序，因此不但需要测试的输入数据，而且需要针对这些输入数据的预期输出结果。如果对测试输入数据没有给出预期的程序输出结果，那么就缺少了检验实测结果的基准，就有可能把一个似是而非的错误结果当成正确结果。

（3）程序员应避免检查自己的程序。测试工作需要严格的作风、客观的态度和冷静的情绪。自己测试自己的软件不容易发现错误，程序员应避免测试自己的程序。测试是一种“挑剔性”的行为，人们常常由于各种原因具有一种不愿否定自己工作的心理，认为揭露自己程序中的问题总不是一件愉快的事，这一心理状态就成为测试自己程序的障碍。心理状态和思维定式是测试自己程序的两大障碍，应由别人或另外的机构来测试程序员编写的程序。另外，程序员对软件规格说明理解错误而引入的错误则更难发现。如果由别人来测试程序员编写的程序，可能会更客观、更有效，并更容易取得成功。要注意的是，这点不能与程序的调试（Debugging）互相混淆，调试由程序员自己来做可能更有效。

（4）在设计测试用例时，应当包括合理的输入条件和不合理的输入条件。合理的输入条件是指能验证程序正确的输入条件，而不合理的输入条件是指异常的、临界的、可能引起问题变异的输入条件。在测试程序时，人们常常倾向于过多地考虑合法的和期望的输入条件，以检查程序是否做了它应该做的事情，而忽视了不合法的和预想不到的输入条件。事实上，软件在投入运行以后，用户的使用往往不遵循事先的约定，使用了一些意外的输入，如用户软件工程案例教程 在键盘上按错了键或打入了非法的命令。如果开发的软件遇到这种情况时不能做出适当的反应，给出相应的信息，那么就容易产生故障，轻则给出错误的结果，重则导致软件失效。因此，软件系统处理非法命令的能力也必须在测试时受到检验。

**软件行业读后感范文 第六篇**

时间飞逝，不知不觉间《软件工程》的学习已经过了大半了。在这将近半学期的学习中，虽然我不能说我将《软件工程》学习的有多么的好，但是通过学习，我还是受益良多。

在以前，我一直对软件存在一些偏见或则是误解，认为软件就是程序，软件的开发就是编写程序，只要编完了程序，一切也就ok了，而且我还片面的认为只要我掌握了时下最新的语言和工具，那么我就能写程序了。一个人，只要会编程，就能写软件，就是程序员；一个公司，只要招聘一些程序员，就能开发好的软件产品。只要有几个有经验的程序员，再找些兼职的大学生，就能组成一个软件公司。

但是通过了《软件工程》这门课的学习，使我认识到了我以前的错误。软件其实不仅仅是程序，软件开发其实也不仅仅是编写程序，软件是思想在硬件上的载体和体现，处理的是逻辑和信息。唯有对软件和软件的开发过程，有充分的认识，才能更好的开发出，过程受控、质量受控的软件产品。

而且在以前，我一直以为软件的开发其实是一件很轻松快乐的事情，只要一天坐在电脑旁敲敲键盘，那么一切就可以了，但是现在我才发现，我以前的很多的思想是多么的肤浅可笑。编程其实是一种乐趣和苦恼共存的一项创造性活动。因为编程不仅能够满足我们内心深处进行创造的渴望，而且还能愉悦我们内在的情感。

而且通过学习《软件工程》，我还学到了很多其他的东西。比如通过学习《软件工程》，特别是老师每次用实际的软件现场的讲解，为我提供了一个尽早接触世界工作和真实项目的机会。让我知道如何在以最小的成本中，训练自己的基本工程素质和能力，如何激发自己的积极性等。而且通过学习《软件工程》，还让我认识和培养了我的团队协作能力，特别是对于我们这些在校的学生来说，这种学习更是能让我在以后工作中少走很多的弯路。

所以，通过《软件工程》的学习，我是真的学习到了很多有用的东西，让我明白了很多的道理。在此我对老师的辛勤教育表示感谢，因为是你让我学习到了这些，是我获益良多。

**软件行业读后感范文 第七篇**

>1、基础很重要

学习《Auto CAD》，需要一定的画法几何的知识和能力，需要一定的识图能力，尤其是几何作图能力，一般来说，手工绘图水平高的人，学起来较容易些，效果较好！

>2、循序渐进

整个学习过程应采用循序渐进的方式，先了解计算机绘图的基本知识，使自己能由浅入深，由简到繁地掌握CAD的使用技术。

>3、学以致用

在学习CAD命令时始终要与实际应用相结合，不要把主要精力花费在各个命令孤立地学习上；把学以致用的原则贯穿整个学习过程，使自己对绘图命令有深刻和形象的理解，有利于培养自己应用CAD独立完成绘图的能力。

>4、熟能生巧

要自己做几个综合实例，详细地进行图形的绘制，使自己可以从全局的角度掌握整个绘图过程。

>掌握技巧：

>1、常见问题要弄懂

（1） 同样画一张图，有的人画的大小适中，有的人画的图形就很小，甚至看不见，这是因为绘图区域界限的设定操作没有做，或虽用LIMITS命令进行了设定，但忘记了用ZOOM命令中的ALL选项对绘图区重新进行规整。绘图区域的设定是根据实际的绘图需要来进行的。

（2） 有人用线型名称为xxxHIDDENxxx的线型画线段，但发现画出的线段看上去像是实线，这是xxx线型比例xxx不合适引起的，也就是说xxx线型比例xxx太大，也可能是太小。结局问题的办法是将线型管理器对话框打开，修改其xxx全局比例因子xxx至合适的数值即可。

cad学习心得体会cad学习心得体会

（3） 在进行尺寸标注以后，有时发现不能看到所标注的尺寸文本，()这是因为尺寸标注的整体比例因子设置的太小，将尺寸标注方式对话框打开，修改其数值即可。

以上三个问题仅仅是我上机过程中遇到的最典型的三个问题和困难。实际问题不胜枚举，作为初学者彻底弄懂这些问题，很有必要，对提高绘图质量和效率很有帮助。

>2、有比较，才有鉴别

容易混淆的命令，要注意使自己弄清它们之间的区别。如ZOOM和SCAIE,PAN和MOVE,DIVIDE和MEASURE等等。

>3、 层次要分明

图层就像是透明的覆盖图，运用它可以很好地组织不同类型的图形信息。学习过程中，有的人图省事，直接从对象特性工具栏的下拉列表框中选取颜色，线型和线宽等实体信息，这很不好，使得处理图形中的信息不那么容易，要特别注意纠正自己的这一不好习惯。严格做到层次分明，规范作图。我的体会是：养成良好习惯，受益匪浅。

>4、 粗线要清楚

能够显示实体的线宽是AutoCAD20\_的新工程。使用线宽，可是用粗线和细线清楚地展现出部件的截面，标高的深度，尺寸线以及不同的对象厚度。作为初学者，一定要通过图层指定线宽，显示线宽。提高自己的图纸质量和表达水平。

>5、 内外有别

利用AutoCAD20\_的xxx块xxx以及属性功能，可以大大提高绘图效率。xxx块xxx有内部块与外部图块之分。内部图块是在一个文件内定义的图块，可以在该文件内部自由作用，内部图块一旦被定义，它就和文件同时被存储和打开。外部图块将xxx块xxx以文件的形式写入磁盘，其他图形文件也可以使用它，要注意这时候外部图块和内部图块的一个重要区别。

>6、 滴水不漏

图案填充要特别注意的地方是构成阴影区域边界的实体必须在它们的端点处相交，也就是说要封闭，要做到xxx滴水不漏xxx;否则会产生错误的填充。初学者一定要学会如何查找xxx漏洞xxx,修复错误。

>7、 写文字要规范

文字是工程图中不可缺少的一部分，比如：尺寸标注文字、图纸说明，注释、标题等，文字和图形一起表达完整的设计思想。尽管AutoCAD20\_提供了很强的文字处理功能，但符合工程制图规范的文字，并没有直接提供。因此要学会设置xxx长仿宋体这一规范文字。具体操作的简要步骤是，打开xxx文字样式xxx对话框，新建一个样式，可取名为xxx长仿宋体xxx,对话框中字体名改为选用xxx仿宋体GB-2312xxx,宽度比例也要改为.尺寸标注的文字可改为xxxxxx代替xxx仿宋体GB-2312xxx.

另一种规范文字更简单的方法是，直接使用AutoCAD20\_样板文件提供的xxx工程字xxx样式；注意，使用前要用xxx使用模板xxx方式启动AutoCAD20\_,选择国标标题（如：GBA3）进入绘图状态。再将xxx工程字xxx样式置为当前工作样式。这种方法，大多数教科书中没有提及，初学者要注意补充一下这一训练。

>8、 特殊字符，特殊处理

**软件行业读后感范文 第八篇**

曾经看过一本书叫《道法自然》，内容略记得一二，但我最欣赏的是它的书名。软件设计没什么太神秘有东西，只要用心体会，其实一切都很自然。软件的设计之“道”，也不在于设计有多么的华丽、精巧，而在于其朴实、自然，最终达到“以无招胜有招”，进入一个全新的境界。

>一、软件设计理论的层次

以我的拙见，软件设计领域中的各种概念，可以分为以下几个层次来进行理解：

1、软件设计的目的：重用性、扩展性。

这是最高的层次，是应对软件危机的需要。

2、设计原则：低耦合、高聚合。

各种软件设计的原则，如依赖倒置原则、单一职则原则、面向接口等，以及各种设计模式，其根本的目的其实只是为了降低耦合这么简单。因为只有低耦合才能更好的适应变化，更好的重用和扩展。

3、实现方法：运用设计模式封装变化、降低耦合。

设计模式只是用来“封装变化、降低耦合”的工具而已。它是面向对象设计时代的产物，其本质就是充分运用面向对象的三个特性，即：封装、继承和多态，进行灵活的组合运用。

>二、关于耦合

1、耦合的粒度

耦合无论如何也是不可避免的。当我们实现接口、继承父类的时候，就会不可避免的产生耦合。耦合是有不同粒度的，我们解耦到什么粒度为止，我认为应以模块的重用粒度为准。尽量解除重用模块或对象之间的耦合。而重用模块之内的耦合，应属于聚合的范畴，所以不要盲目的去解耦，否则就陷入了误区。

2、解耦的原理

怎样才能解耦呢，或者说为什么各种设计模式能达到解耦的目的呢？我觉得有以下几个思路：

（1）将具体的东西抽象处理

（2）将分散的东西集中处理

而面向对象中的接口、继承正为我们提供了这样的一种机制。通过访问接口或基类或抽象类，而不是具体的实现类，从而与具体的实现类达到了解耦的目的。我们还可以设计一些控制类，像润滑剂一样，协调各实现类之间的访问，也可以达到耦的目的。

事实上，各种设计模式的基本思想也就是这样。创建型模式是为了解除创建对象时产生的耦合，实际上是解除对类称名的依赖，而结构型和行为型是为了解除对象属性或方法的直接调用。不管什么设计模式，都是将对具体实现类的访问提升为对接口、基类或用于协调的控制类的访问。

>三、关于接口

这一节更具体，谈一谈接口，因为使用接口是软件设计的重要手段，但已经不属于“道”了~

1、接口与继承

接口描述的是对象某一个方面行为特征。使用接口与使用继承关系各有优缺点，使用子类继承可以继承父类的功能，体现了重用的精神。而接品更加灵活，因为它解除了子类与父类之间的高度耦合，它体现在灵活扩展的精神。

2、接口与纯虚类

理论上接口可以由纯虚基类实现类似的功能，那为什么还我们不去掉接口的概念，而直接使用虚类呢？

接口存在的理由就是它更加灵活，关系简单，易于理解。比如一个类可以实现十几个甚至几十个接口，但一般开发工具只支持单继承（由于多继承太容易导致混乱和冲突），如果要继承十几层，系统结构想必会无法理解了，我以为这是接口存在的最重要的原因。

如果接口和虚类继承结合使用，可以产生强大的威力，这也是许多设计模式的“杀手锏”。

以上算是总结一下自己的心得。肯定有不少片面之处，请各位指教。

**软件行业读后感范文 第九篇**

曾经一度认为软件测试就是使用工具测试bug，现在看来不是这么一回事情，因为还是有手工测试（执行测试），工具只是一个辅助，用工具你先要去了解测试的一些基本的东西（如：测试用例，预期结果等），不是那按两下按钮就行了，就算是录制脚本，也需要看懂脚本的代码，工具不是万能的。

一开始接触软件测试觉得很枯燥乏味，全都是一些理论的东西，还不如回到小学学习语文呢，都是一些名词的解释，比如：黑盒测试，百合测试，系统测试。测试基础等等这些，老师都会去告诉你这些名词什么意思，很无聊，到后来慢慢由语文变成了数学，开始练习测试用列的编写，这个还有点意思，因为这个更多时候能够体现个人的逻辑思维能力，再然后数学就转变成了英语，因为要使用到一些测试的工具，比如：winrunner工具，录制脚本它会产生一些代码，不过代码比较好理解，虽然是英文的但是还是很好看懂的。

学习软件测试一学期，其实我觉得最重要的是兴趣，有了兴趣还是不行的，还需要具备一些语言的基础，例如：c，java，c#等一些语言，这些语言你不需要去深入的学习，只需要了解，最重要的是了解数据库（例如：sql，mysql，oracle）的知识，想要成为一个好的测试工程师，应该要全面的发展，读懂需求分析文档（注：客户的要求），还有要学会写文档，语言的组织能力决定你这份文档的价值，这也是一种沟通能力的体现，比如写缺陷报告时：有一项是描述缺陷，这就能看出你的表达能力，给程序员能不能看懂就能体现沟通，最后就是整理文档和撰写测试总结报告，越是到最后越是要细心，因为软件永远都是有缺陷的，我们的细心可以让软件减少一些bug，不求最好，只求更好。

**软件行业读后感范文 第十篇**

至少自己做的东西，是真正交付到了客户手上，到也稍微有些成就感。

>1、浅谈测试

一直以来，我都认为测试是脱离了软件工程范围的工作，不以为屑。但在实际情况中，测试是既重要且难以精湛的.其真正的压力，在于找不到bug，责任在你，而不在于编码人员。一般的测试人员不懂编码，他们靠的是日以累计的经验总结和想象力。而要做到高级测试工程师，则一定要懂编码，因为这是你完全掌握整个系统的方方面面具体运作的前提。但占主导地位的，还是大型系统的集成测试经验。实际项目中，编码时间一般只占30%左右，真正耗费时间的是IT阶段的找 bug与对应bug，此阶段基本评定了coder的编码质量。

>2、程序员的困惑

有些人，以为教学视频和代码看多，自己就懂的多，实际做起来，却不知从何下手，

问题在那?如何定位?如何解决?通通跟一样能力有关，debug追踪能力，也称调试。在项目组工作不愁源码资源，但问题是蛋糕摆在面前，你如何去消化?

有位同事告诉我：代码看几遍都没用，要去抄，例如一个查询模块，在此基础上去做具体记录的历史记录查询模块，你可能会觉得很简单，但实际情况却往往报一堆异常，配置问题涉及到方方面面，以及数据库字段，传值问题等等，一大堆对于新人来说很郁闷的问题。但不用怕，只要学会调试，一个个问题去追踪，一个个去解决，自然而然，那段“源码”才真正属于你。

>3、如何调试追踪

>4、一些习惯

遇到问题先自己思考，无从下手再找高手帮忙看看，注意他帮你看的思路，别在一旁闲着，看多了自己也会了，不然你一辈子都停留在那种水平，从人身上学到的东西远远比书多的多。

解决了一个问题后，要去究根问底去找到问题产生的起因，以防你下次遇到类似的问题再浪费同样的时间。

把代码写的漂亮，注释、空行、规范一样不能少，可读性是放在第一位。曾经看过一个高手写的代码，真的一看就是不同水平的人写的，几乎很完美，读起来很流畅，方便自己也方便别人。

任务完后不要呆着，去要求经理给你更有挑战性的任务，只要你肯去尝试，他们就会对你另言相看，把三天的任务一天加班搞定，效率和忠诚都有了，路也比较好走了。

**软件行业读后感范文 第十一篇**

PS是一款很实用的处理图形图像的软件。PS只是软件简称，其全名为Photoshop。它和其它计算机图形图像处理程序一样，通过操作工具，在菜单、调板和对话框中做出各种选择来使用。在对图片进行修改以前，必须告诉PS要修改图中的哪些部分，可以选择一个图层或它的一个蒙版，或在一个图层内确定选区。如果不做选择，PS就会假定不限制修改，将把修改应用到正在工作的图层或蒙版的所有地方。

Photoshop通过自身不断完善，已经出现了很多版本。目前较为流行的有photoshop CS2、photoshop CS3、photoshop CS4。它们在很多部分都是共同的，只是新款比老款新增了一些功能。PS功能包罗万象，包括了不少其它类似的软件的很多功能，当然这也是这类图形图像处理软件的共通之处，所以，学习图形图像处理时，没太多的必要学许多其他乱七八糟的软件，如果你实在很感兴趣学别的，又有充足的时间，充沛的精力，我当然也不反对。就我个人而言，还是强烈建议研习透某款Photoshop每项功能，熟透每个工具的使用方法，自然攻下PS就不在话下，说不定，你一不小心，就成了PS高手或其它类似软件的高手。

>对那些正在学习PS或正准备学习PS的人,我有以下几点建议：

首先，不能有畏难情绪，千万不要被PS的表象——繁多的工具吓到。要知道千里之行始于足下，只要你坚持不懈，循序渐进。采用逐个击破，有计划有步骤地学会每个工具的功能与使用方法。在学的过程中一定要思考如何应用，注意触类旁通。其实，学着学着，你就会发现一些工具的用法有很多相同的地方，例如仿制图章工具和修复画笔工具的用法基本相同。但如果满足于停留在知道怎么用它的水平还远远不够，我们得其次，得想方设法提高自己的兴趣。大多数人都喜欢欣赏漂亮的照片，何不自己动手美化自己的照片，让自己摇身一变，比那些明星还要美丽。这样既可以强化自己原有的知识结构体系，又能激发自己追求新的知识的欲望，何乐而不为呢?

再次，学而时习之，要每隔一段时间，练习一下，比较一下，看自己是否有进步。最好独立地做一些创造性的作品。坚持这样，在不知不觉中，就会很轻松地把所学知识纳入长时记忆系统。

学习PS其实并不难，下定决心学好它吧!我期待大家都能学好它，都能做出漂亮的作品来。

**软件行业读后感范文 第十二篇**

这个暑假惠普派人到我们学校来开展软件测试培训。老师说机会难得所以我就参加了，说实话每天在教师从早晨坐到下午，中间只有一个半小时休息时间，这样还是相当累人的。我们第一天开始就觉得这个简直比平常上课还累啊。

不过 看到老师讲得如此认真，看到惠普如此强大，我看在座的学员都听得非常认真。所以向我这种上课从来不听讲的这回都听得认真得不得了，呵呵。

前两天确实还是有点累，讲的也是理论课，而且以前我们从来没有接触过测试这个行业，所以听得也嘿吃力。但是老师给我们讲了不少他们的工作经验和惠普这种世界五百强美国十强的企业文化，鄙人是深受教育啊。

后两天我们每个人带一个笔记本进行上机操作了。我们的第一个任务就是安装软件，那个软件好大啊 ，整整2个g。我们考啊考啊考了好久才考完。软件叫qtp，就是惠普的快速测试专业版。确实是一个强大的软件，呵呵 大家用了就晓得了!

我记得老师说得最有感情的一句话就是“社会是黑暗的啊”。我们对这句话都是深信不疑!所以以后呢，要好好努力啊，不管社会有 好黑暗你都能找到光明，生活就是如此，时间本就平凡。好好干好好干!

**软件行业读后感范文 第十三篇**

一、总结：

1.自身定位：在过去一年，是我进公司的第一年，也是我工作的第一年，刚开始在我对工作竞争和自身都不甚了解的情况下，在领导和同事的指导下，我感觉自己已经慢慢对人与人的竞争和自身定位有了深刻的了解，因为有了自我目标，才能感受到自己的压力有多大!我的目标也不只是完成目前所要做的工作而已，要向其它方面拓展学习。

2.定下心来，踏踏实实：我学的是计算机专业，我的工作也是计算机方面的，以前有什么优势，但是踏入工作岗位后才发现，自己学的只是一个基础，只是有些方面或许比别人走的快一步，所以一切都要靠自己.自己要定得心下来学习.成功需要耐得住寂寞，不求最快，但求.

3.团队合作：以前在学校或许你可以靠一个取得好成绩，在工作上你必须要有一个团队，在一个部门之中，团队合作精神显得尤为重要.以前我做有些事都是一意孤行，但现在已经对自己改变了，多听听他人意见，会犯更少错误，会更长见识，所以要学会与同事之间的合作，做事才更有效。

4.工作情况：在公司一年，对mes大型系统有了个大概了解，对我们所要学习的mes已经可以说差不多都掌握，条码打印机的维修和设置掌握，a4打印机大多数情况可以维护，pda、条码枪已掌握，电脑的系统重装和维护已掌握，其它基本设置可以维护，对新出来的程序掌握和了解也比较快。

二、自身缺点

1.沟通问题：自己的沟通能力只能算一般，因为对于某些事的阐释还是不怎么好，语言表达能力有点差，希望通过平时的交流和沟通来加强。

2.心态问题：自己对于做某些事过于着急，一心想急切完成，确反而误时，这个问题一开始就一直出现，现在虽然已经基本克服，但也要列入缺点方面，希望以后时刻注意!

**软件行业读后感范文 第十四篇**

我应该是对maya榜首阶段的学习有一个初步的知道了，在接maya之前一向觉得这个软件很神秘，当栩栩如生的三维动画即是用它制造出来的，我又对这个软件产生了几分“爱意”。带着这份“爱意”我到清美报名了，招待的教师很热心，教育教师也很认真担任，这儿不负担阐述。下面就我榜首阶段的学习作一个简略的学习心得报告：

maya软件的主要有四个模块：maya建模、maya动画人物建模、maya原料灯光、maya人物动画、maya后期特效。我首要接触到的即是建模，想学好三维动画，当然要把这个maya人物建模学好啦。

三维动画的建模主要有：

a、nurbs曲面建模技能。

b、polygons多边形建模技能。

c、subp细分建模技能。

当李教师给咱们上完榜首堂课后，我对maya的“爱意”就全消了，因为软件根本都是英文，而我又是一个英语盲，除了简略“thank you”以外估量就说不出完好的第二句，加上翻了一下教材，难度之大彻底触及了我的底线。

当然“爱意”登时衰退，但是我仍是很认真地去学，鉴于我自个的工作发展方向的需求和动漫工业的朝阳性，我的学习动力依旧，教师也很担任地在教授。由于写字速度慢，字又写得欠好，所以上课做的龙飞凤舞的笔记也只要我自个看的懂，杂乱无章的圈圈叉叉中蕴涵着大致内容和操作过程，全部上午所要学的东西许多很繁琐，着手操作时脑袋经常会混杂一些东西，致使过程紊乱，所以做我做起东西来会很慢。好在咱们上课的时分都人手一机，能够及时消化，在不至于打乱仗。

通过教师多天来的孜孜不倦的教训，以及自个的勤学苦练。一个月后自个总算能够着手做出完好的模型了，嘿嘿……尽管做出的东西还比照粗糙和天真。通过几天的激战，最终做出了机器猫，那个得瑟劲不知道咱们能否领会，让我也有了无比的成就感。当然现实是严酷的，在我得瑟完毕静下心来，把自个做的模型和参阅的图像进行比照时，我才知道大相径庭是什么意思。这种差距源自于两者的创造目的，源自于时间的磨炼，源自于阅历，文明和技能的沉积，快餐和速成文明对咱们这代人的影响等等等等。这些都是我从两幅著作的比照中领会到的。著作自身不会说话，却能够感染我这么去想。

所以我此时此刻坚信一句话“学习是没有捷径可言的，仅有的捷径即是不断的学习和操练，只要这么才能学到真功夫”。我学maya还只迈出了榜首步，路漫漫其修远兮，吾将愈加尽力。我也相信在李教师的教训下，我在清美一定能学到更多的东西。今后的学习心得我会渐渐更新，期望有相同阅历的兄弟能够多多沟通!

**软件行业读后感范文 第十五篇**

转眼间，到昆山已经两个多月了。不知不觉中我已经从一个在校生变成了一个职员。这跟在我们学校是完全不一样的。除此之外，安博还制定了严格的制度，这些使我们在安博的培训像职工在公司工作一样，让我们提早接触到公司的氛围。

来安博最重要的目的还是学技术，那就说说这里的教育情况吧。安博实行的是上午授课，下午上机练习的制度。我觉得我们这个班上午的授课经理非常好，他对java的理解非常透彻。我在大学学了半年的java，仅仅停留在表面上，对实质的内容根本都不了解。比如说==与equals的区别，方法的覆盖，变量的隐藏等等。老师通过图的方式，给我们讲解它们在内存中的情况，使我们从本质上了解了这些东西。他的这种授课方式，既生动又形象，彻底地将问题讲明白，我们接受起来轻松容易，也不容易遗忘。

除了他的讲课方式以外，他还是一个非常幽默的人，坐在凳子上听四个小时的课，会很乏味的，他时不时的给我们说一些搞笑的事，或者开玩笑的话，使课堂气氛非常活跃。他每讲完一个新知识点，都给我们留一点时间练习，加深对新知识的理解。我们有什么问题，他都会很耐心的跟我们讲解，不管程度是好是坏，他都一样对待。总之，听他的课就是一种享受。他还把跟知识相关的材料发给我们，让我们有研究的空间。有时还给我们一些面试题，让我们提早看，只有准备好了去应聘才有机会。

昆山还给我们开了一些素质课，讲解一些职场素质，如何为人处事，如何同面试官讲话，还要求我们在日常生活中也尽量做到。在我们就业之前开这样的课，对我们这些即将踏入社会的大学生来说是非常重要的。公司招聘员工，一看实力，二看素质，两者缺一不可。

总之在昆山的这段时间中，我学到了很多。时间虽短，但所学到的和知识的实用性很强。所有的老师们都教给了我们很多工作习惯、工作技巧、日常礼仪、职业素养和心态方面的东西。使我们对今后的工作有了新的认识，增添的极大的信心。

**软件行业读后感范文 第十六篇**

在走秀期间，我们测试部总监还会对我们不定时的培训。教会我们测试的工作流程和每个阶段应该展开的工作范畴。作为测试，必要会使用的缺陷管理工具bugzilla和测试用例管理工具testlink，还给我们培训了，如何使用自动化工具ruby+watir来对一些测试点进行自动化脚本的编写。慢慢的，在对公司的业务了解的比较透的`时候，老大就开始让我们自己对一些小需求进行测试，测试的过程中，不仅仅是对页面和表面功能进行测试，还要根据需求文档和页面的显示对数据库表进行查询操作，查看页面的显示和功能是否和数据表里面的一致，还要在后台日志中查看是否有报错。所以，测试并不是像我想象中的那么简单，不是在页面上点来点去就可以测的好的。

实习可以使每一个学生有更多的机会尝试不同的工作，扮演不同的社会角色，逐步完成职业化角色的转化，发现自己真实的潜力和兴趣，以奠定良好的事业基础，也为自我成长丰富了阅历，促进整个社会人才资源的优化配置。作为一名学生，我想学习的目的不在于通过毕业考试，而是为了获取知识，获取工作技能，换句话说，在学校学习是为了能够适应社会的需要，通过学习保证能够完成将来的工作，为社会做出贡献。然而步出象牙塔步入社会是有很大落差的，能够以进入公司实习作为缓冲，对我而言是一件幸事，通过实习工作了解到工作的实际需要，使得学习的目的性更明确，得到的效果也相应的更好。

人要想成功及获得好的业绩，必须牢记一个规则：我们永远不能将个人利益凌驾于团队利益之上，在团队工作中，会出现在自己的协助下同时也从中受益的情况，反过来看，自己本身受益其中，这是保证自己成功的最重要的因素之一。

**软件行业读后感范文 第十七篇**

在本学期的软件工程课程的学习中，我们学习了十一章的内容。第一章软件与软件工程的概念，这一章主要讲解的是一些概念性和基础性的内容，例如软件的概念、特性，软件危机的主要表现，软件工程的概念以及软件生存期、典型生存期模型等等。第二章软件工程方法与工具，这一章主要对软件工程方法进行介绍，包括三种方法：传统方法、面向对象方法、形式化方法。还引出了工具UML。第三章软件需求获取与结构化分析方法，本章详细介绍了需求获取与需求分析阶段的任务以及结构化分析方法，画分层的数据流图、E—R图以及状态图式本节的重点。第四章结构化分析方法，这一章重点讲解了使用变换型映射方法和事务型映射方法生成初始的模块结构以及模块结构的改进。第五章编码，这一章重点讲解了编码的风格及规范，还告诉我们编码规范说带来的好处，并告诫我们将来一点要形成好的编码风格。第六章软件测试方法，本章讲解了软件测试相关的概念及重要性，软件测试与开发各个阶段的关系；还介绍了白盒测试技术以及黑河测试技术。第七章统一建模语言UML概述，本章详细介绍了UML的基本模式、事物、关系及建模时用到的各种图进行了介绍。第八章面向对象分析，这一章主要讲解了面向对象分析的3种模型，包括功能模型、静态模型和动态模型。第九章软件体系结构与设计模式，本章对软件体系结构的基本概念、典型风格等进行了讲解。第十章面向对象设计，本章的重点是对面向对象分析时建立的对象模型进行调整和细化。第十一章软件维护，本章主要介绍软件维护的任务、软件维护活动以及软件维护方法进行了介绍。

要学习软件工程，学会如何系统的思考，以及养成良好的编码习惯，想学好软件工程，就必须知道软件工程的目标、过程和原则：软件工程目标：生产具有正确性、可用性以及开销合宜的产品。正确性指软件产品达到预期功能的程度。可用性指软件基本结构、实现及文档为用户可用的程度。开销合宜是指软件开发、运行的整个开销满足用户要求的程度。这些目标的实现不论在理论上还是在实践中均存在很多待解决的问题，它们形成了对过程、过程模型及工程方法选取的约束。

软件工程过程:生产一个最终能满足需求且达到工程目标的软件产品所需要的步骤。软件工程过程主要包括开发过程、运作过程、维护过程。它们覆盖了需求、设计、实现、确认以及维护等活动。需求活动包括问题分析和需求分析。问题分析获取需求定义，又称软件需求规约。需求分析生成功能规约。设计活动一般包括概要设计和详细设计。概要设计建立整个软件系统结构，包括子系统、模块以及相关层次的说明、每一模块的接口定义。详细设计产生程序员可用的模块说明，包括每一模块中数据结构说明及加工描述。实现活动把设计结果转换为可执行的程序代码。确认活动贯穿于整个开发过程，实现完成后的确认，保证最终产品满足用户的要求。维护活动包括使用过程中的扩充、修改与完善。伴随以上过程，还有管理过程、支持过程、培训过程等。

软件工程的原则是指围绕工程设计、工程支持以及工程管理在软件开发过程中必须遵循的原则。

我们学习了详细设计的方法，其原则是过程描述是否易于理解、复审和维护，进而过程描述能够自然地转换成代码，并保证详细设计与代码完全一致。包括程序流程图、N—S图、PAD图、HIPO图

程序流程图：程序流程图又称之为程序框图，它是软件开发者最熟悉的一种算法表达工具。它独立于任何一种程序设计语言，比较直观和清晰地描述过程的控制流程，易于学习掌握。在流程图中只能使用下述的五种基本控制结构：顺序型；选择型；while型循环；until型循环；多情况型选择。

N—S图：一种符合结构化程序设计原则的图形描述工具，称为盒图，又称为N—S图。在N—S图中，为了表示五种基本控制结构，规定了五种图形构件。顺序型；选择型；WHILE重复型；UNTIL重复型；多分支选择型。

PAD图：它是用结构化程序设计思想表现程序逻辑结构的图形工具。PAD也设置了五种基本控制结构的图示，并允许递归使用。

HIPO图：HIPO图是由一组IPO图加一张HC图组成。它是美国IBM公司在软件设计中使用的主要表达工具。

HC图既是层次图，用于表示软件的分层结构。HC图中的每一个模块，均可用一张IPO图来描述。IPO图由输入、处理和输出三个框组成，需要时还可以增加一个数据文件框，这种图形的优点，是能够直观地显示输入—处理—输出三者之间的联系。

还有测试方法：按照测试过程是否在实际应用环境中来分，有静态分析与动态测试。测试方法有分析方法（包括静态分析法与白盒法）与非分析方法（称黑盒法）。

静态分析技术：不执行被测软件，可对需求分析说明书、软件设计说明书、源程序做结构检查、流程分析、符号执行来找出软件错误。

动态测试技术：当把程序作为一个函数，输入的全体称为函数的定义域，输出的全体称为函数的值域，函数则描述了输入的定义域与输出值域的关系。

还学习了其他很多工具、语言、方法等，虽然不是都学得很透彻，但我相信在今后的学习中一定会慢慢的完善的。

软件工程对于初学者来说，知识基础较薄弱，对一些应用操作、概念、工具方法等理解起来较为困难，要能从整体概念上较好地理解和把握、学好软件工程，不是仅仅把几本专业书籍细致地看几遍，然后上机练习几次就可以成功，学习过程中要注意多看多练要注意结合实际，更要多思考，面对错误不要一范就问，要尝试自己去解决。但是还要注意什么都学，肯定是什么都学不透的，要集中精力打攻坚战，学习软件工程首先要明白自己的学习目标究竟是什么，根据自己的实际工作出发，有针对性的在相应的学习方向上进行提高，制定出详细的学习规划。还要注意与其他科目的相辅相成，就像我们在学习面向对象分析的时候要结合大一学习的面向对象及其方法学这一专业科目进行研究拓展；在学习语言时，要看看与C语言的联系，多思多想，把从各个科目学到的知识通汇贯通。

在软件工程的学习中，我了解到了软件并非是一些代码这么简单，在开发软件的过程中，编写代码的工作量其实只占不到所有工程量的30%，而后期的管理和维护更是占了60%到80%之多。一个完整的项目规划须包括，软件的定义，可行性分析报告，项目开发计划，软件需求说明书，概要设计说明书，详细设计说明书，用户操作手册，测试计划，测试分析报告，开发进度报告，项目开发总结报告，软件维护手册，软件问题报告，软件修改报告，等多个文档，每个文档都要上级验收审查，而文档数量众多，要做好这点真的不是很容易，而恰恰写好文档正能保证完成软件工程其中一个目的的关键，既研究如何用最小的开销做出生存期较长的软件，再加上各个阶段都要进行周密的策划、详细的分工部署和人员安排，且各阶段要据具体情况不断的反复才能达成，所以代码只是开发软件这个浩大的工程的一个小小的过程。

而编码的学习中，我更了解到形成自己独特的规范的编码风格是非常重要的事。因为这影响到了软件后期繁重的维护，大家都要阅读你的程序，如果你写的程序毫无规范可言，那么别人怎么能读懂你的程序？读不懂程序，维护又从何谈起呢？所以，我们在今后的学习中，一定要注意这方面的培养，在写程序的过程中，要逐步的在规范的基础上形成属于自己的风格，即方便自己的修改，也方便日后他人的阅读。

在学习中，我们还要注意比较三种方法的优缺点，例如：传统方法虽然使软件摆脱了混乱和无序，但其在适应需求变化的方面不够灵活，而且传统方法要么面向行为，要么面向数据，缺乏两者的有机结合。而面向对象方法的程序设计和问题求解更符合人们日常自然的思维习惯，适合大型、复杂及交互性比较强的系统。形式化方法则是一中基于形式化数学变换的软件开发方法，它可将系统的规格说明转换为可执行的程序。

在今后的学习中要注意多读书、多思考、多练习、多讨论，不断熟悉书本的基础，并以此为基础将其扩散开来，应用于今后的实践。不断锻炼自己，向一名合格的程序设计师迈进。

**软件行业读后感范文 第十八篇**

经过两个多月的ERP软件相关知识的学习，我掌握了ERP软件的基本操作，从中收获甚多。我感觉到学习了ERP软件将对我毕业以后的社会就业有很大的帮助。就如何进行有效和针对性的学习，我得出了自己在学习ERP软件期间的一些感想与体会。

作为一个大四的学生，我还是首次比较系统地了解ERP，当老师在学习之初给我们介绍这一技术时，我突然感觉到我已经很落伍了。是的，到了大四，学校已经对管理类学生开设了好多专业课程。但是但老师对我们提到ERP时，我们没有几个人能够真地说出它是什么来，甚至是根本没听说过。作为新时代的大学生我没有做到对当今流行的生产生活中的新技术予以于时俱进的了解，这是我的一大损失。原来ERP不只是让人觉得深奥的几个字母，它对我们生产生活能产生好大的影响。

ERP代表着新时代的企业管理模式，它的出现再次证明科学知识在信息社会的重要性。一个企业再也别想仅仅靠激情，靠勇气，靠机遇就能运营的很好流畅，更需要的是科学的管理方式。一直一来我们财经管理类的学生在学习中很少有机会接触到企业的真实运作，而ERP课程的开设正是学习锻炼的最好时机。ERP系统的会计子系统与ERP系统的其他子系统融合在一起，会计子系统又集财务会计、管理会计、成本会计于一体。这种系统整合，及其系统的信息供给，有利于财务做前瞻性分析与预测。综上所述，将ERP系统中按西方管理会计理念及其方法设计的会计信息与我国现有的会计信息系统融合为我国现行的财务会计核算体系，实现ERP系统中的会计信息融合具有现实的意义。

在这两个月的实训课程中，我深深体会到ERP到给企业的帮助和作用不是一点点，ERR适应企业，企业适应ERP！我们的实训课程分为基础设置、采购管理、销售管理、财务会计与mps的计划执行等几大模块。通过学习，我了解到了企业业务的运行，可能很多小企业由于资金和技术的问题没有使用ERP对企业进行管理，导致很多问题的发生，如：不能按时发货、需求与预测不适应等。而使用ERP对企业进行管理的企业能够把这几大模块通过管理人员的授权分工地完成工作，这有利于企业提高生产效率与管理水平。下面我详细地分享一下我对这几大模块的体会。

>一、基础设置合理化

企业的运行离不开人员的配置，人事管理的核心是做到人岗匹配，任人为贤。录入基础数据时必须仔细，避免因错误导致后期工作的失误。

>二、采购管理与mps相结合

物料的采购对于生产制造型企业尤为重要。很多企业都实行“零库存”管理与JIT生产方式，丰田汽车就是典型的企业。但是，要想真正的实现“零库存”是很难的，也存在很大风险。ERP就开辟一条新的道路，采用采购管理与mps相结合的方法，利用mps计划执行，对库存与销售、采购进行控制，能够保证企业有安全的库存，又不至于挤压过多的原材料在仓库，又能按时发货给客户。

>三、销售管理与mps相结合

企业在接销售订单前都必须对市场进行预测，为了预测更加准确，可以使用与mps相结合的方法。填制销售订单后，可以输入产品需求表，通过供需平衡预算可以了解到企业未来的需求，这有利于企业作采购和生产计划。

>四、会计电算化优化企业记账程序

随着电子信息化的不断发展，企业运用计算机系统软件进行管理。会计作为企业财务的重要环节，运用ERP可以实现电子记账，不用手工记账，大大降低了由于人为疏忽导致错误的发生，而且提高了企业的财务核算的效率。

这个学期经过了一段时间与ERP的接触，已经对ERP有了一定的了解，但是感觉上自己还只是停留在刚刚认识ERP这个层面上，还没能真正的熟练掌握而且灵活的运用在实务中了，虽然还是有点不足，但是毕竟开始了对ERP有一定的了解，也算是有点安慰了，我会继续努力完善的。

总结两小点：

1、学好和掌握ERP软件要多花时间和精力上机做实验与多做上机练习。

2、要学会总结总结，无论在什么阶段，在处理什么事情都起到很重要的作用最后要感谢老师的悉心指导，在短短两个多月的实训中薄老师都能充分将自己的知识传授给我们，并且有耐心的给我们讲解所有问题，帮助每位同学了解企业的事务。我认为通过实践去验证理论比直接从书本上获得理论要深刻得多。

**软件行业读后感范文 第十九篇**

11月下旬，我离开了学校，进入了最后一学期的顶岗实习阶段。从离开学校的那天起，我就当自己是一个正式的员工，做好了经历重重困难的准备，我始终坚信，我的努力与坚持不懈不会白费，我的未来不是梦。

一、实习单位介绍：

二、实习岗位、主要过程、主要内容、主要工作、收获

我在该公司任职软件开发工程师，刚实习的第一个星期，主要的工作是熟悉公司的开发框架及开发流程，及时完成项目经理派发的任务。从第二个星期开始，我就开始接触公司的开发项目。令我印象最深、体会也最深的一个项目是《南京大学110周年校庆系统》。该项目不仅开发时间短(只有一个月的时间)，最主要的是需求不清。在客户与公司的双重压力下，我们这组人(共5人)在与客户只沟通了两次的需求下，我们就开始动工。该项目的用户主要分为校领导和嘉宾。针对校领导和嘉宾进行相应的车辆、餐厅、宾馆、会场、时间、志愿者的安排，要有冲突提示功能。如：在一个重叠的时间段内，一个人不能同时参加两项及以上活动;一辆车安排的人数不能超过他的最大承载数等等。在做这些功能的时候，我们有过以人为基础，分配车辆、餐厅等资源，也有考虑过以资源为主，分配人员等种种意见。在每一次意见有了分歧以后，我们都要讨论，各抒己见。这种方式使我乐于表达自己的想法，当我的想法得到了其他同事的认可时，那种感觉无法用言语来表达。即使是自己的想法错了，但听到同事的另一番看法，使我茅塞顿开，也有了收获的喜悦。由于项目太紧，我们经常加班，每晚都要坐末班车回去，好几次回来后倒头就睡，周末也没得休息。正是有了这些无以言表的感觉才使我对这个项目，对这个阶段很难忘。在项目的功能方面，我负责所有基础信息的添加与列表显示。在做这些的时候，与我在学校做的项目的最大不同是分页都是用的存储过程，验证最好要做到客户端和服务端同时验证。还涉及到站内信互发、打印、导出、发邮件等这些我以前看似很神奇的东西，现在我已把神奇变为了现实。

在这半年多的工作实践中，我学到了很多，这些都是作为学生学不到的，如：

1、我参与的都是团队合作的项目，无论是在自己公司，还是在合作公司，我们都讲究团队精神。在信任自己和他人的基础上，思想统一，行动一致，这样的团队一定会攻无不克、战无不胜。我们的很多工作是一起完成的，在这个过程中，大家互相提醒和补充，大大提高了工作效率，所有的工作中沟通是最重要的，一定要把信息处理的及时、有效和清晰。

2、每一个项目在开始着手的第一步，一定要和客户把需求沟通清楚，只有了解了项目的需求，才有可能真正做好一个项目。我们需要向客户提出要求客户提供所有涉及该项目的资料，每次与客户见面都需要熟悉业务与程序的最少一至两个程序员随时记录需求。

3、工作中，有一个无论是技术，还是经验都比较让人肯定的前辈带领，将任务详细化，详细到每个页面、甚至是一个页面中的图片什么时候做好，做到什么程度，这样把工作进度有计划有方向的暂定下来，做事很有效率。

4、每天的工作小结真的很重要，这让我们每天都有计划的知道自己干了什么，不是漫无目的的工作。

5、工作并不是一成不变的，也许有一天你要去其他岗位帮忙，所以同事之间的技术要互相学习，也许有一天，公司需要你发挥其他的技能帮忙，所以互相学习也是很重要的，我在公司，深有感触，除编程之外，切图、界面设计等工作都干过。

三、体会

转眼，出来社会都已大半年，已是半个社会人了。不能再向学生那样，某些时候可以随心随意。顶岗实习，为我们提供了一个很好的实践机会，可以让我们更好的把理论应用于实践，在实践中领悟理论，更可以学习到很多书本上学习不到的、甚至比理论知识更实用的业务知识。而且，这些实习经验，无疑是我们毕业后就业的一大筹码。作为一个成年人，作为一个社会职业人，任何时候都要守规矩，做好自己的本分，承担起自己所需要承担的责任。经历了2家公司的工作，我渐渐的认识到，每一份工作或每一个工作环境都无法尽善尽美，但每一份工作中都有许多宝贵的经验和资源，如失败的沮丧、自我成长的喜悦、温馨的工作伙伴、值得感谢的客户等等，这些都是工作成功者必须体验的感受和必备的财富。如果每天怀着感恩的心情去工作，在工作中始终牢记“拥有一份工作，就要懂得感恩”的道理，你一定会收获很多很多。在你收获很多很多的同时，你会发现自己已经在锻炼中变得勇敢，坚强，乐观，阔达。这样的你，是不断前进的走在成功的路上的。

软件工程实习心得感想

基于本次暑期实习的情况，我收获很大。

从实训目的来看，本次实训主要是在vs开发平台上实现对c#语言、sql数据库的应用，就我所在小组的项目工作内容来看，还需要掌握水晶报表和mdi多窗口应用技术。除了要充分熟悉开发平台及应用设计之外，还要掌握并实践软件项目设计规范及其开发流程即需求阶段、分析和设计阶段、编码阶段和测试阶段，以便能按时交付作业。

从实训的内容来看，我小组的项目是“人事工资管理系统”，主要实现公司企业对内部员工档案、工资、奖罚考勤等一系列的管理，这在当今不断壮大的企业用人单位是亟待需要的，总的来说，项目的难度系数不是很高，但也是相当困难。

在校外实习之前，老师先在校内培训了5天。5天实训中，着重讲解了c#语言应用，并布置编写“通讯录”工作台程序，windows应用程序和wpf程序，数据先是用文本保存，然后练习用数据库保存，期间每天需上交当日作业。5天的时间里，通过不断地学习，我从完全不懂c#语言成长到初步掌握了这门语言，当然在具体应用中还是会遇到些问题，但我还在学习。其实在实习的前一个月里，老师就上传了一些关于c#语言的资料并通知我们需要配置哪些软件环境，对于这一点，我是相当感谢何老师的，当然在校外的实习里，公司也全都把开发项目用到的一系列软件上传到公共文件夹里，从而免去了我们下载软件的麻烦，在此，感谢众老师。

校内实习要上交的作业我是在校外实习中补充完成的，因为之前上交的作业由于时间仓促大都有错误，所以借着校外实习中对c#语言的进一步应用和讲师的详细讲解后，我重新改正了错误，希望何老师能谅解采纳。

校外实习是从7月12号开始，历时15天，15天的最后一天要回校评审。我对本次校外实习的各个方面都十分满意。相比上次实习，首先这里的住宿学习条件非常好，住宿和实习地都安装了空调，实习地方环境清幽宽敞明亮，伙食也非常好。然后就是实习讲师非常专业和负责，他事先把所有有关实习的资料软件存放在公共文件夹，且规划入微。一开始先是对我们进行c#测试分好组安排好每组项目任务，又指导我们每个小组安装svn软件，方便小组文件上传与统一，整体一气呵成，不似之前实习的凌乱，这一点我十分敬佩。之后每天都先安排好任务工作，认真且非常详细的给我们操作示范并细心地制作成word文档供小组之后查看。每隔几天都会有小组评审，主要针对小组所编文档的规范程度和不足提出修改建议。作为小组的pm，我必须承认一开始工作任务安排不当，没有完全发挥各成员的优势，以致文档编写的有些瑕疵，不过在小组评审过后，我便及时更改了项目工作的分工，而之后小组项目工作进行的非常顺利，每个人的能力都得到了最大的发挥和升华，当然这完全得力于小组成员的高度配合。

我小组是c#软件开发第一组，组内有6名成员，两个主要负责项目核心代码的编写，其余负责简单模块代码的编写，我主要负责填写与修改相关项目文档并督促小组进度和团结组内成员、合理分配组员任务并及时调节组内安排的各种不适与问题。校外实习的过程中，小组成员高度团结，协调合作，对所有问题都能及时探讨并高效解决，有时遇到分歧，不过在全体讨论后，都能找到合理的统一方案，如在界面设计环节中，就登陆界面和模块界面的设计中，组员有很多不同建议，在询问老师和权衡实现困难程度后，我们达成了一致意见，统一后的界面相比之前更加规整明晰，且功能更加健全，当然这是我们小组齐心合力的结果。在开发的过程中，出现过项目分工不明确的问题，其原因源于我考虑到组员能力的不同，所以就把大部分的工作安排给能力高的人身上，其他能力低的只暂让他们自由学习希望能尽早跟上小组步伐，不过事实证明我错了。让部分组员自由学习而不参与项目的工作中本身就拖累了小组的整体速度，而且在短暂的时间里，他们也实在学不到任何实际的东西，所以后来我更改了战略。我先让能力低的组员尝试参照模板编写一些文档的简单部分，然后让能力稍高点的组员继续补充文档的核心部分，最后由我修改格式和内容，最后让老师审阅。这个方案实施过后，小组的效率整体上明显提高，每个人各司其职，全组团结一致，共同为项目的完成发挥自己的聪明才智。而之后，在项目编码阶段，我们小组的tl负责对项目整个框架的构建，即设计好主要的窗口原型，然后指导小组成员在自己所负责的模块上添加具体事件代码，最后再进行修改，这样做不仅将组员团结在一起提高组内效率，更直接提高了组员的编程能力。

需求分析阶段和设计阶段很快结束了，我们小组齐心协力按时完成了相关文档的编写，包括需求规约、实现规约、界面设计word图形文档、数据库设计说明书、详细设计说明书和概要说明书，在老师评审过后，发现问题还是有很多，作为项目经理的我本该对这些文档进行及时修改和更正，但由于时间仓促且我们是第一组需第一个评审，所以就马马虎虎的拿未精确修改的文档进行审评了，老师着重对系统模块图和模块功能描述那块提出了些许建议，并让演示了小组编写的代码程序，对其中错误问题提出了解决方案。之后进行一番修改过后，我小组正式进入了编码阶段，编码只有短暂的3天，后两天老师分小组评审并提出改进意见，小组修正和改进代码结构语句，完善项目要求，紧接着进入测试阶段，老师又着重介绍了一些测试工具方便小组测试，填写并修改好测试文档后，整个实习就将要进入尾声了。再回校进行最后评审项目ppt和提交作业后，暑期实习就正式结束。

短短半个月的实习中，我深深的感受到自己在实际应用中专业知识的匮乏和短时间学习掌握一门新知识和新技术的困难，这是我非常畏惧的两大难题。就自身来说，我喜欢有所准备，不喜欢临阵磨枪，尽管之前对实习要求的知识技能有些了解，但到真正实施时，还是不免有些不知所措，而之后能慢慢克服畏惧慌张并团结小组成员按时且高效的完成作业，这都得力于实习老师和组员对我的巨大帮助，在此，由衷感谢他们，也感谢我自己。

**软件行业读后感范文 第二十篇**

>引言

>一、对用友财务软件的初步认识

用友ERP-U8为企业提供一套企业基础信息管理平台解决方案，满足各级管理者对不同信息的需求：为高层经营管理者提供决策信息，以衡量收益与风险的关系，制定企业长远发展战略；为中层管理人员提供详细的管理信息，以实现投入与产出的最优配比；为基层管理人员提供及时准确的成本费用信息，以实现预算管理、控制成本费用。用友ERP-U8，根据业务范围和应用对象的不同，划分为财务管理、供应链、生产制造、人力资源、决策支持、集团财务、企业门户、行业插件等系列产品，由40多个系统构成，各系统之间信息高度共享。

>（一）用友财务软件系统管理

用友ERP-U8有多个子系统组成，各个子系统都是为同一个主体的不同方面服务的。用友ERP-U8为各个系统提供了一个公共平台，用于对整个系统的公共任务进行统一管理，系统管理是对用友ERP-U8管理系统的各个产品进行统一的操作管理和数据维护，具体包括账套管理、年度账管理、系统用户及操作权限的集中管理、设置统一的安全机制等几个方面。我们重点学习了账套管理、用户及权限管理和年度帐管理。

账套管理包括账套的建立、修改、引入和输出。其中，系统管理员有权进行账套的建立、引入和输出操作，而账套信息的修改则由账套主管负责。初次使用该系统，第一次必须以系统管理员的身份注册系统管理，建立账套和制定相应的账套主管之后，才能以账套主管的身份注册系统管理。由系统管理员建立账套并设置了账套主管后，在未使用相关信息的前提下，账套主管可以对某些信息进行调整，以使信息更真实准确地反映企业的相关内容。

用户及权限的管理主要包括设置用户、设置角色和设置用户的权限等内容。在系统设置了角色后，就可以定义角色的权限，当用户归属某一角色后，就相应地拥有了该角色的权限。角色管理主要包括角色的增加、删除、修改等维护工作。在系统中，用户不仅可以建立多个账套，而且每个账套可以存放不同年度的会计数据。对不同的核算单位、不同时期数据的操作只需通过设置相应的系统路径即可进行，而且系统

可以自动保存不同会计历史数据，方便了对历史数据的查询和比较分析。年度账管理主要包括建立年账、清空年度数据、引入和输出年度账及结转上年数据等内容。

>（二）用友财务软件总账管理

**软件行业读后感范文 第二十一篇**

>一、需求分析和概要设计。

>1）需求分析

按照软件工程的软件过程来说：

1需求分析产生了软件功能规格说明书,需要确定用户对软件的需求,要作到明确、无歧义。不涉及具体实现方法。用户能看得明白，开发人员也可据此进行下面的工作（概要设计）。

2.概要设计产生了软件概要设计说明书，说明系统模块划分、选择的技术路线等，整体说明软件的实现思路。并且需要指出关键技术难点等。

1.本系统在高校应用后高校工资管理方面的教职工将减少至目前的50%左右；

2.本系统在高校应用后将在高校各方面的成本将会有所降低；

3.本系统在高校应用后将教职工的工资达到完全透明，计算更加精确教职工因纠纷事件减少到1%。 根据分析将系统的功能从一般教职工与系统管理者两个角度将功能划分为7个模块，当然介于我们的知识有限，有的功能没有实现：员工工资与考勤直接挂钩，但本系统无法与员工考勤系统挂钩相连，由于涉及此系统时该高校并没有员工考勤系统，而且我们在最初进行商量的时候也没有提出该要求。

>2）概要设计

从概要阶段开发正式进入软件的实际开发阶段，本阶段完成系统的大致设计并明确系统的数据结构与软件结构。在软件设计阶段主要是把一个软件需求转化为软件表示的过程，这种表示只是描绘出软件的总的概貌。由概要设计说产生大的概要说明书的目的就是进一步细化软件设计阶段得出的软件总体概貌，把它加工成在程序细节上非常接近于源程序的软件表示。

在本阶段主要涉及处理流程的设计、总体结构和模块外部设计、功能分配。在接口设计上有用户接口、外部接口、内部接口；数据结构设计有逻辑结构设计、物理结构设计等等。在接口设计时参考了大量的资料。

最后就是编写文档——软件需求说明书、概要分析说明书。

而文档的作用在于：一是可以帮助整理思路。把要完成的目标，系统的结构，每一个模块的功能等整理一下，然后分门别类地写下来，这样在开发的过程中，就有据可依，在需要回过头来修改设计的时候，也有证可考。二是便于交流。三是可以作为以后维护时的参考资料。

>三、软件工程课程设计——心得体会

我们进行了为期一周的课程设计。通过这次课程设计，我拓宽了知识面，锻炼了能力，综合素质得到较大提高。安排课程设计的基本目的，在于通过理论与实际的结合、人与人的沟通，进一步提高思想觉悟。尤其是观察、分析和解决问题的实际工作能力，以便培养成为能够主动适应社会主义现代化建设需要的高素质的复合型人才。作为整个学习体系的有机组成部分，课程设计虽然安排在一周进行，但并不具有绝对独立的意义。它的一个重要功能，在于运用学习成果，检验学习成果。运用学习成果，把课堂上学到的系统化的理论知识，尝试性地应用于实际设计工作，并从理论的高度对设计工作的现代化提出一些有针对性的建议和设想。检验学习成果，看一看课堂学习与实际工作到底有多大距离，并通过综合分析，找出学习中存在的不足，以便为完善学习计划，改变学习内容与方法提供实践依据。对我们信息管理与信息系统专业的学生来说，实际能力的培养至关重要，而这种实际能力的培养单靠课堂教学是远远不够的，必须从课堂走向实践。这也是一次预演和准备毕业设计工作。通过课程设计，让我们找出自身状况与实际需要的差距，并在以后的学习期间及时补充相关知识，为求职与正式工作做好充分的知识、能力准备，从而缩短从校园走向社会的心理转型期。课程设计促进了我系人才培养计划的完善和课程设置的

本文档由范文网【dddot.com】收集整理，更多优质范文文档请移步dddot.com站内查找