

最新程序设计心得体会(实用12篇)

通过写工作心得，我们可以发现不足之处，及时进行改进和提升。考虑到每个实习生的实习经历都不尽相同，下面这些实习心得范文提供了不同的视角和观点，希望能给您一些新的思考。

程序设计心得体会篇一

指针是C语言中非常重要的一种数据类型，同时也是非常难以理解和运用的。在进行指针程序设计实验的过程中，我深刻地感受到了这一点。下面，我将分享一下我在这一过程中的心得体会。

一、指针令人望而生畏

指针是C语言中的一种数据类型，它是对于内存中的某个位置进行操作的一种方式。由于指针的运用非常灵活，可以完成很多复杂的操作，所以在程序设计中，指针经常被使用到。但对于初学者来说，指针的概念非常抽象，让人望而生畏。在我的实验开始之前，我也感到非常迷惑和无从下手。

二、多加练习是重要的解决方法

在完成实验过程中，我发现只有不断地练习才能掌握指针的使用方法。在编写程序时，我不断地试错，通过多次尝试和修改，最终找到正确的实现方式。尤其是在指针运用较为复杂的情况下，需要更多的时间和精力去实践和体会。

三、指针应用举足轻重

在程序设计中，指针可谓是举足轻重的存在。通过指针的运用，我们可以实现结构体的嵌套、动态内存的分配和释放等

高级操作。在实验过程中，我尝试着使用指针解决了一些复杂问题，减少了代码的冗余和可读性的降低。

四、指针的运用需谨慎

指针虽然功能强大，但是在使用时需要十分小心谨慎。一些不当的操作可能会导致程序的崩溃、数据的丢失等问题。如使用未初始化的指针进行访问会导致程序崩溃，访问数组时下标越界也会导致程序异常。在实验过程中，我也曾因为指针的使用不当而导致程序出错，对此我深感需要加强自己的代码规范和严谨性。

五、指针让我更深入理解程序

通过指针程序设计实验，我更深入地理解了程序的本质和一些底层的知识。我了解了内存的分配、变量的存储、数据类型的本质等诸多概念。在实验过程中，我也不断学习和探索，让我认识到了不断学习和尝试的重要性。

总之，通过这次指针程序设计实验，我学到了很多新的知识和技能，并且更深入地理解了编程中的一些概念。希望在未来的学习中，我能够更好地运用指针来解决问题，提高自己的编程水平。

程序设计心得体会篇二

当得知要编一计算器的程序时，觉得不是很难，但后来仔细一想似乎要满足老师的要求并不是那么容易，越想越觉得复杂。当老师给我们上了一节指导课后，我看到了希望。再经过两个星期的仔细构想，我感到胜利在望。于是兴孜饬饬得坐进了机房，手指在键盘上不停的跳跃，不一会就一部分写好了，真的蛮兴奋的。于是又开始第二部分的编写，又好了，当我按下`alt+f4`+回车时，没得到我预想的结果，有点意外。于是移动光标上下扫描，好象没发现任何语法错误，我猜可

能是结构的顺序出了问题，再仔细的将程序看了遍，想了想。结果发现第二部分没被运行。接下来我对疹下药，果然有用，这人真是喜啊！在接下来的几节课我完成了余下的部分，当然不可能一切非常顺利，其中的曲曲折折是很难说清楚的。

当写完了这程序我深深地感到写程序是一件很有趣的事，给人喜悦，给人快感，给人满足。其中的感觉真是难以用言语来表达，即使是一个小小的程序运行成功那也要比打游戏打死一个大boss要爽快！总之这一切只有身临其境才知道。

其次还对同学之间的合作精神深感赞叹，同学之间的互相帮助精神在这时体现的淋漓浸渍。

程序设计心得体会篇三

在软件实验室经过2个星期jsp程序设计实训，从对软件开发没有太多经验的我掌握了更多的软件开发的方法和技能，收获颇多。

经过这次理论学习，了解到要做好软件开发，要求掌握的知识并不是只有编写代码方面的，网络、数据库、操作系统等的知识对做好项目开发是很有帮助的。这让我明确了以后学习的目标，在不断学习软件开发的同时，也应该继续其他相关知识的深入学习。

实训期间经历的许许多多，给我们带来了很多很多有实无形的东西，知识、友情以及团队开发的热情，快乐无一不包围着我们。在这我很想感谢几个人，王璐老师和班上一同参与这次实训的2位同学。王璐老师对我们备至关心，不论在学习前后，都给予我们关怀不少，并且王璐老师不愧是从事多年软件开发的专业人士，其深厚的专业技能知识和丰富的经验令我等钦佩无比。对于我们未出茅庐的大学生而言，遇到的问题多而复杂，询问有经验的人，他们或许一句话就能解决问题，而自己就得找资料或上网摸索可能要一两天，有时甚

至绞尽完脑汁都不得其解。用个词形容，老师的解答如醍醐灌顶，而对我们，则甘之如饴。

实训这段期间真的好累，尽管实验室给我们提供了一个相当不错的上课学习环境，但是整天木木地待在实验室面对毫无生气的荧屏，为code而烦恼，为bug而郁闷，被文档所包围，随程序侵蚀。然而，这就是学习的生活，既选之也则安之。忙碌的生活让我每天似乎没有太多的时间去想一些其他的事，耐心的去调试，专注的去编写更好的算法，沉浸充实和快乐之中。

更重要的是，两个星期的实训使我们明白我们所欠缺的不仅仅是技术知识，更重要的是有一种处理事情的方法、面对问题的心态和动手能力。面对完全陌生的新知识、新技术、新项目以及整个it行业，我们不能畏惧，要以一种积极的心态去面对，分析并抓住关键所在。因为我们所即将应对的每一个项目都是既需要实际操作，又需要详细规划的。作为组长，协调组员、激励其他组员和积极参与项目开发是我每天必做的工作。我认为每个人都应该在团队中做好自己应尽的职责，再优秀的个人也可能完成一个即庞大又复杂的项目工作，我们必需紧密的联合在一起，以一个团队的角色来面对。

总之，这次实训为我们以后从事软件开发工作打下了良好的专业基础，为我们的进一步学习提高打下了扎实的理论基础。

程序设计心得体会篇四

_____计算机软件_____

1

一、实训目的

java程序设计是计算机软件的专业核心课程，而且java也是当

前软件市场上较流行和功能很好的一种软件开发语言。经过这个学期的学习后，能不能把学习的知识综合串联起来[]java的实训项目的制作可以有效的检验这一点。进行综合项目实训可以练习java软件开发的环境和配置，能够把java的面向对象技术、数据结构、图形界面、线程、网络知识、小应用程序等知识运用到实践中去，从而更好地理解java的各项知识，达到能够应用自如的地步。进行实训的同时还需要查阅大量的文档，可以锻炼同学的自学能力。

二、实训内容

每个同学完成一个小项目，小项目的选题可以从规定题目中，也可以自己定题目。建议选做的题目如下：

- 1、画图软件。要求：能够使用菜单功能完成基本图形的绘制(画圆、直线等)，能够画较复杂的图形(如房子)，能够有橡皮擦、改变画笔粗细等功能。
- 2、记事本。仿照windows下的记事本，要求：能够读写文件，文件内容的复制、剪切和粘贴，能够设置字体的大小和颜色。
- 3、计算器。仿照windows下的计算器能够完成基本的计算任务，能够完成平方、开方、求余等一些特殊的运算。
- 4、万年历。要求：能够查阅农历、阳历和星期。
- 5、拼图游戏。能够完成拼图游戏，能够设置游戏的级别和时间，记录游戏用户的成绩。
- 6、扫雷游戏。模仿windows下的扫雷游戏。
- 7、英汉词典。要求：能够进行英汉单词的互译，进行英语单词的记忆训练以及一些简单句子的翻译。
- 8、设计一个俄罗斯方块的游戏。

9、设计一个简单的聊天软件。要求：注册和登陆功能，客户端能够进行通信。10、动态时钟的制作。要求：仿照手表制作一个时钟。

每个同学也可以自己选择一个题目，要求不能抄袭他人作品，独立完成任务。

三、课程设计项目内容和简介(在此填入项目名称和进行项目设计的背景及简单介绍)

四、系统环境(开发、运行的硬件环境和软件环境)

操作系统[]windows;内存[]256m;

cpu[]interceleron1g;jdk版本： 1.5;

五、进度计划及执行情况

日期-1-7任务安排了解记事本的基本功能，并实现基本的框架功能实现方法调试结果错误原因分析改正情况日期-1-8功能实现方法调试结果错误原因分析改正情况对文件可以进行正确的读写。刚开始进行文件读写操作时，用的是字节流，读出来的都是乱码。数据的流使用错了。使用了字符流解决了问题任务安排实现文件的打开和保存功能文件的打开和保存功能执行情况对文件进行正确的读写经过查api函数。

在jscrollpane中添加文本域控件。在将jscrollpane解决了问题。最后程序能正常运行遇到的问题是：在开始时，文本域控件jtext没有滚动条进行了整个窗口的布局执行情况基本框架搭建好了日期2008-1-9功能实现方法调试结果错误原因分析改正情况日期2008-1-10功能实现方法调试结果错误原因分析改正情况日期2008-1-11任务安排实现字体风格的保存，保证打开后，能保持原来的风格。功能实现方法调试结果错误原因分析将字体风格保存到另一个文件中，打开文本时，也同

时读取该文件，重新进行字体的设置。

遇到过的一次错误是，颜色不能正常显示，改进过，后来程序可以实现文本打开后字体维持原状。错误的使用颜色的函数执行情况功能已经实现有待改进可以进行查找和替换还有点功能上的欠缺查找到到最后一个时。不能返回重新查找任务安排实现查找和替换功能应用的是string常用的几个方法执行情况可以正确进行查找和替换在新建后可以解决该问题可以对字体的字体，字号，还有字体颜色进行相关设置，新建文件后，原来的字体颜色还保留着在新建时没有进行颜色的设置任务安排实现字体风格和小的设置执行情况可以对字体进行正确设置字体名是调用系统的，字号用的是8-99号字，颜色调用java的颜色面板改正情况已经没有问题，可以正常显示颜色。

六、程序清单

加程序压缩包

七、综合实训总结文章

java实训感想

通过一周的实训，虽然实训的时间不长，但是总体上收获是很大的，我们的java课程学到的只是java和皮毛，一开始我还没有信心完成。后来看到书到有一个关于记事本的小例子，我仔细看了之后。对记事本有了一个大概的了解，在初训的实候知道自己应该做一个什么样的记事本了，在实训的过程中也遇到了不少困难，通过查api文档和看书，解决了问题，在写到字符的查找的替换时后，用到了上课时学到的知识，对字体的风格的保存里面的颜色分离和还原方法，是通过查api文档获得的,现在我所写的记事本只实现在一些基本功能。简单的复制粘贴，字体风格的设置，还有就是查找和替换功能以及一些字体风格的保存.对文件的操作基本都用到了。

在实现记事本的功能过程中，也不是很顺利，基本的文件读写和保存文档的复制、粘贴、删除等功能还比较容易实现，但如何改变字体、字体颜色和背景颜色，就比较麻烦了，要实现字体颜色和背景颜色，首先要调用系统的调色板，再进行设置。但如何调用系统调色板？我是通过查阅帮助文档来找到其调用方法的。

像自动换行、设置字体等都是通过方法来调用，给相关事件加上监听，在进行响应事件。整个程序中方法才是最关键的，也是最困难的。所以像查找、替换、转到几个功能我都不知道用什么方法，也由于时间有限，没有完成其功能。总体的感觉是对java有了一个全新的认识，现在感觉用java做小的图形用户程序有些缺陷，但是在其的应用方面。尤其是有网络方面的应用，在整个实训我也尝试过做一个小的画图软件。但是由于时间关系还有一个是自身的知识的欠缺，最终这个事情被搁浅下来了。最后还是进行记事本的编写。由于在做画图软件时花了不了时间，所以现在的记事本的功能不是很全，如果有更多的时间，我会把这个记事本再完善一下，再把画图软件也做出来。实训的时间不长，但是我会把在课外把它们的功能完善。然后再学习一下java和网络的编程，体验java的前沿技术。

在学习的过程中也体会到，遇到问题也要请教。不然的话自己一个人去做的话会花很多的时间，我在做这个记事本的时间，其中也问到过许多问题，通过同学的帮助下解决了，提高了程序的开发进程。其他的同学有问题的话我也是给他们尽量解决。我认为学习知识的过程当中就应该是相互学习相互进步的。java是现在的主流技术我会努力将它学好，现在我缺乏的是坚持不懈的精神，周贤军给了我很大的影响，我会向他学习，虽然学习java的路还很找，但我会继续走下去。

程序设计心得体会篇五

数据结构是计算机存储、组织数据的方式。数据结构是指相互之间存在一种或多种特定关系的数据元素的集合。通常情况下，精心选择的'数据结构可以带来更高的运行或者存储效率。数据结构往往同高效的检索算法和索引技术有关。

一般认为，一个数据结构是由数据元素依据某种逻辑联系组织起来的。对数据元素间逻辑关系的描述称为数据的逻辑结构；数据必须在计算机中存储，数据的存储结构是数据结构的实现形式，是其在计算机内的表示；此外讨论一个数据结构必须同时讨论在该类数据结构上执行的运算才有意义。一个逻辑数据结构可以有多种存储结构，且各种存储结构影响数据处理的效率。

在许多类型的程序的设计中，数据结构的选择是一个基本的设计考虑因素。许多大型系统的构造经验表明，系统实现的困难程度和系统构造的质量都严重的依赖于是否选择了最优的数据结构。许多时候，确定了数据结构后，算法就容易得到了。有些时候事情也会反过来，我们根据特定的算法来选择数据结构与之适应。不论哪种情况，选择合适的数据结构都是非常重要的。

从上面我们了解到了数据结构的重要性，它是一个程序的关键因素。拿到“阴风习习的大楼”这个题目时，一开始不知道从哪入手，以为可以把每个位置看成一个图的一个节点，从而可以用邻接表这种结构，通过遍历邻接表，每次选择权值小的路径走就可以得到最短的路径了。我错了，完全的错了。通过认真的分析后，我认识到，这能简单的看成一个图，因为每个位置都只能到达它邻接的位置而不是和其他各位置都有联系。而且如果每个位置转化为图的一个节点，这样要表示每个节点的关系就需要 $n*n$ 节点和 n^4 大小的数组来存储各个点的关系。还有这个大楼结构想转化为一个图也是不容易的。根据书上有些类似的“老鼠迷宫问题”，让我想打，

不如就用 $n*n$ 大小的数组直接存储大楼结构。通过类似走迷宫的方法来遍历，当遍历完所有路径后就能得到一个最小的路径。那接下来怎样遍历呢？我采用了深度优先遍历的方法，这样可以用递归的方法，简化代码。虽然理解上有一定困难，而且递归条件的控制要注意。

通过这次数据结构程序设计，不仅让我对c语言的一些知识得到了回顾，加深了对c语言的掌握。同时也让我对数据结构知识得到了一定的掌握。知道了怎么去分析一个题目，怎样选择比较好的数据结构。让我从怎么样实现一个程序功能，转变到怎么用更少的空间，更短的时间完成程序设计。空间和时间上对程序的优化是评价一个程序好坏的关键标准。

通过这次数据结构的程序设计也让我懂得了怎么样去设计一个程序。从问题分析中找出程序所要解决的关键问题和数据结构的选择；在概要设计中完成程序的大体轮廓；在详细设计中解决关键问题的算法和设计；在调试分析中完成程序最终的修补。这样一个比较好的程序就设计出来了。

程序设计心得体会篇六

作为计算机专业的一名学生，指针程序设计一直是我们的学习过程中的难点之一。但是，在指针程序设计实验中，我深切体会到了指针的强大，也从中受益匪浅。下面，我将从实验准备、实验过程、实验成果三个方面，分享我对指针程序设计实验的心得体会。

一、实验准备

在进行指针程序设计实验前，我首先需要了解并掌握指针这种数据类型，比如指针变量的定义、指针变量的初始化、指针与数组的关系等等。除此之外，还需要掌握指针的运算方式，如指针的自增、自减、指针的加减等。在理论基础掌握之后，我还需要编写相关的程序验证和巩固自己的知识。

二、实验过程

在实验中，我发现指针和数组的关系密切。通过指针，我们可以方便地对数组进行访问和操作，提高了程序的效率。在实验过程中，我们需要通过指针找到数组中特定元素的地址，进而进行修改或输出操作。此外，还需要熟练掌握指针变量的使用方法，如通过指针变量间接访问数据、通过指针变量作为参数传递、返回值等。

三、实验成果

在完成指针程序设计实验后，我深刻认识到指针的重要性和实用性。通过这次实验，我不仅熟练掌握了指针的定义和运算方法，还学会了如何利用指针提高程序的效率和可读性。在实验过程中，我也发现了自身的不足之处，如对指针的理解不够深入、程序调试能力有限等。这让我更加注重提升自身的编程能力和理论水平。

四、实验收获与启示

通过指针程序设计实验，我对指针的应用有了更加深入的理解和掌握。同时，也发现了自身的不足之处，从而更加注重学习和提升自身编程能力。此外，还认识到简洁高效的代码对于编程的重要性，这将对我的日后编程工作产生深远的影响。

五、总结

指针程序设计实验是非常重要的部分，通过实践的方式来提高我们的理论知识，并且检验我们的能力与技术水平。在实验过程中，我们应该注重掌握各种知识与技巧，适当学会优化代码，提高程序效率。同时对于自身的不足之处也可以进行探讨与改进，走上编程大师之路。在以后的学习中，我将继续加强对指针的学习和掌握，熟悉其常见应用场景，以

便更好地应用到实际编程中。

程序设计心得体会篇七

近年来，计算机控制系统在各个领域中的应用越来越广泛。而作为控制系统的核心部分，控制器的微程序设计是保证系统高效运行的重要环节。在我参与控制器微程序设计的实践中，我深刻认识到了其重要性和特点，并得到了一些宝贵的经验与体会。

首先，在控制器微程序设计中，正确的指令调度是至关重要的。在一个复杂的控制系统中，有各种各样的任务需要完成，其中一些任务之间有时是相互循环的。合理地调度这些任务和指令的顺序，可以极大地提高系统的性能和效率。而在调度过程中，我们需要考虑指令的前后依赖关系、任务的优先级以及系统的硬件资源等因素。只有在考虑到这些因素的前提下，才能使控制系统平稳运行，充分发挥其潜力。

其次，在控制器微程序设计中，对硬件资源的合理利用也是至关重要的。由于控制器的微程序设计主要是针对特定的硬件而进行的，因此需要密切关注硬件资源的利用情况。合理分配和利用硬件资源，可以使控制系统在给定资源的情况下取得最佳的系统响应时间和处理速度，提高整个系统的效率。为了实现这一目标，我们可以结合实际情况，灵活选择合适的控制器和硬件架构，并进行相应的优化和调整。

再次，在控制器微程序设计中，充分发挥硬件和软件的协同作用是非常重要的。控制器的微程序设计需要依赖特定的硬件设备进行实施，而在具体的实践中，准确理解和掌握硬件设备的特点和限制，并灵活地运用软件技术对其进行协同，可以实现更高效的控制系统。例如，我们可以通过选择合适的硬件控制模块，合理设计硬件和软件的接口，实现控制器和外部设备之间良好通信和协作。只有在硬件和软件的紧密合作下，才能使控制系统发挥最大的潜力。

另外，在控制器微程序设计中，良好的系统仿真和测试也是非常重要的。系统仿真和测试可以有效评估控制器的性能，验证微程序设计的正确性和稳定性，并为后续的优化和调整提供反馈。通过系统仿真，我们可以模拟实际控制场景，检查控制器的各种指令和算法的运行情况，及时发现和解决问题。同时，测试也是一个必不可少的环节，通过对实际系统进行测试，我们可以验证控制器的微程序设计是否符合实际需求，并根据测试结果进行针对性的调整和完善。

总之，控制器微程序设计是控制系统实现高效运行的重要环节。在这个过程中，我们需要合理调度指令、充分发挥硬件资源、优化硬件和软件的协同作用、进行系统仿真和测试等。通过不断的尝试和实践，我深刻认识到了这些要素的重要性，并获得了一定的经验和体会。相信在不断地积累与学习中，我可以在控制器微程序设计这一领域中不断进步，为实现更高效的控制系统做出自己的贡献。

程序设计心得体会篇八

实训总结

短短的一个月很快就过去了，在这短短的一个月里，我学到了很多，了解了很多。在这一个月里我学到了有关java等方面的知识，了解了关于软件开发的流程。了解自己的不足，知道了自己努力的方向。

回顾这次的实训，这次或许是最后的实训机会。我是一名大二的学生，明年或许就要出去实习了，所以我很珍惜这次的实训机会，因为它能够让我了解自己的不足以及以后自己努力的方向，同时也能让我了解软件开发的流程，增加一点软件开发经验和社会经验。让我在以后的实习中会更好的融入到其中，增加自己的就业机会，因为纵观现在的就业形势很不让人乐观，由于之前的经济危机，就业机会越来越少，就业也越来越难，这就给了我们很大的压力，所以要是没有真

本事，就业岗位就无从谈起，因此，在以后的学习中，我会更加好好努力。

通过这次的实训，我学到了很多：

首先，对java识比以前有了更深的了解。在这之前由于种种原因我java学的很不好，编程我几乎写不出来。但经过这次的实训，我对java的理解，虽然还有很多都不懂，但我会在今后的实训和学习中加以学习了解，力求弄懂，增强自己对java的理解。

其次，在这次的实训中我的动手操作能力有了一点的提高，刚开始的时候jdk的配置，数据库的安装都出现了一定的问题。java可实话开发工具的不熟悉，所以开始的时候进程很慢，时间都用在了jdk的配置，数据库的安装以及熟悉java可视化开发工具上，但付出有了回报，成功的配置了jdk，安装了数据库，熟悉了java可视化开发工具，总的说来，自己还是有一定的收获的。因为自己的动手操纵能力得到了提高。

最后是团队协作。在整个项目完成过程中团队协作有着不可替代的作用。从在刚拿到项目时对项目的分析到最后的项目完结的都有一定的体现。刚拿到项目时，我们团队进行了分析，并分配了各自的任务。当我们其中一人遇到问题的时候，我们其他人都会去帮忙，效率提升了很多。但可能由于不是一个宿舍的，大家彼此间都不是很了解，所以交流中还是存在了一定的问题。

在这次的实训当中我收获颇丰。但仅仅靠这一个月的学习还是远远不够的。所以在以后的学习中我会更加努力，提高自己的能力，让自己在以后的社会道路上打下坚实的基础。

姚新

1月4号

扩展阅读□java个人实习总结

4、在做项目的时候不应该有参考代码

长话短说就是以上几点，如果你想要更多的了解，可以继续往后看。

一：项目的地位

因为参加就业培训的学员很多都是有一定的计算机基础，大部分都具备一定的编程基础，尤其是在校或者是刚毕业的学生，多少都有一些基础。他们欠缺的主要是两点：

(1)不能全面系统的、深入的掌握某种技术，也就是会的挺多，但都是皮毛，不能满足就业的需要。

(2)没有任何实际的开发经验，完全是想象中学习，考试还行，一到实际开发和应用就歇菜了。

学以致用，学完的知识点需要到应用中使用，才能够真正理解和掌握，再说了，软件开发是一个动手能力要求很高的行业，什么算会了，那就是能够做出来，写出代码来，把问题解决了，你就算会了。

4：最后综合项目一定要是实际应用系统

学员经过这个项目的练习，就要走上实际的工作岗位了，如果这个系统还达不到实际应用系统的标准，学员练习过后也还是达不到企业实际的需要，那么这个培训应该说质量就不高了。理想的状况是这个项目就是实际项目，到时候学员就业到另外一个公司，不过是换个地方干活而已，完全没有技能上的问题。

三：java怎么选项目

这个不是靠想象，而是根据实际情况一步一步分析出来的(呵呵要卖弄一下：这也算是逻辑思维)，当然这里只是讨论方法，不涉及具体的项目案例。我们可以采用倒推的方式来分析：

(3) 然后开始选择分阶段的项目，分阶段的项目应该比刚才分析出来的分阶段任务稍稍复杂点，这样才能达到训练的目标。定下分阶段项目后，同样去分解，定出为了完成他所需要的各部分知识点和深度。

(5) 最终把要求掌握的知识点和要求掌握的深度，分散到日常教学和练习中。好了，把上面的分析用正向描述就是：

日常教学和练习中学习和掌握的知识==> 演示项目==> 分阶段项目==> 综合项目==> 胜任企业实际开发需要。

附注：有了好的项目还要有好的训练方法(这个另外立文讨论)，最最基本和重要的一条就是：绝对要指导学生亲自动手做，而不是看和听，所以尽量不要选择有现成代码的项目，演示用的项目也最好不配发代码，让学生能够跟着做出来才是真的掌握了。

上面阐述了java对项目的态度和基本选择方法，这里再把其中几个重要的、与众不同的特点阐述如下：

1：真项目

项目一定要真实，要是企业实际应用的，不能是教学性的项目，否则会脱离实达不到项目实训的效果。还有一个一定要是最新的项目，企业的要求也是在不断变化的，应用技术的方向和层次也在不断变化，这些都体现在最新项目的要求上，旧项目所要求的技术和层次很可能已经过时了，根本达不到

训练的目的。

开发的流程也要跟企业的实际开发保持一致。从项目立项开始，到需求分析、概要设计、详细设计、编码、测试的各个环节，都要完全按照真实的开发流程来做。

开发的环境也要跟企业一样，包括常用的开发工具、开发平台、应用服务器、常用插件、测试工具、项目管理工具、项目管理文档等等。

这是java最与众不同的一点□java深知：软件是做出来的，而不是听出来或者看出来的，企业需要能实际开发的人员，而不是只听过、看过但不能做的人。所以java十分强调：项目一定要让学生动手写出来。

java的做法是：老师做为项目经理，带领着大家一起去需求分析、概要设计、数据结构设计、接口设计、重要业务流程的分析设计等，然后由学生们分团队进行开发，实现整个项目。这里有几个与众不同：

(1)：现场带领学生一起做分析和设计，而不是预先做好了来讲讲。因为需要学生学习的是分析设计的过程和方法，而不是已经做好的结果□(2)□java选用的项目基本都是企业最新的项目，很多都是java的学生和企业同期开发，所以是不配发源代码的。这样也断绝了学生的依赖思想，认认真真去开发。可能有同学会问：为什么不选择有源代码的项目呢，那样还可以参考学习啊！

这是一个典型的误区：觉得通过看人家的代码能够加快自己的学习。这个方法对有经验的人来讲是对的，但是对于还在学习期间，没有经验的学员来讲，就是一个认识的误区了。因为你也许能看懂代码的语法，但根本看不懂代码背后的含义，看不懂代码的设计，看不懂为什么要这么实现，看不懂代码所体现的思想，其实，看了跟没看没有多大区别。事实

上java界最不缺的就是代码了，那么多开源软件，小到一个工具，到企业级erp都有开源的，而且很多都是大师级的作品，说实话比你参考的那些代码质量高多了，但是又有多少人能真正看明白呢，原因就在于达不到那个水平，看不懂代码背后所体现的设计和思想以及为什么要这么做，效果也就不好了。

当然不排除有这样的高人是能够完全能理解和掌握这些大师级作品的，但估计不会是刚开始学习java开发的学生，所以java根据多年的培训经验，友情提示：在初学阶段，看一千遍，听一千遍，都不如自己动手写一遍所达到的学习效果。自己能写出来的功能才是你真正掌握的功能，而不是你看得代码或者是听老师讲的，就算讲过看过，但是自己做不出来，那都不算会java的主力老师都是实战出身，在java开发方面都至少有七年以上的开发经验，同时具备多年的项目管理经验，所以能够完全按照企业开发的流程来训练学生。

java实训收获总结

尊敬的各位领导、各位老师、各位同学：

大家好！

在学院领导老师的带领和安排下，我们在上个学期末，到北京进行了为期10天的专业实习。在实习过程中，我们在专编程技能以及软件开发的总体架构思想上都收获颇丰。

一、理论知识的学习

在实训进行的前五天，我们上的是java编程的理论课。负责上课的老师言语风趣幽默而不失严谨，在老师的引导和启发下，我们巩固了之前在学校所学的编程知识，并加以深化，澄清之前对编程技术认识的一些错误或模糊的概念。

我们还在课上以java里gui编程常用的swing框架和java的容

器体系作为切入点，框架作为这期间除了学习一般的编程的知识技巧。在全面的系统的认识了框架同时补充学习了泛型技术，以及框架中架构思想的知识。在企业的实际编码过程中，一个个框架构成了软件的基石，只有掌握了几门框架知识，才能在实践中高效开发，让自己的产品在市场上立于不败之地。

在结束了理论知识的学习之后，后五天我们在实训基地进行了项目开发的实战练习。

项目的名称是supervcd[]是由海辉曾经所接的一个远程音乐唱片管理软件项目的精简版。虽然有所精简，但是其内容对我们学习仍具有相当的价值，对我们了解真实的软件的设计规范，编码规范，文档规范以及客户对软件的需求很有参考价值。该内容囊括了gui设计和编码，容器与泛型的知识，规范文档编写，测试及日志系统，甚至要求我们实现一个专用的轻量级数据库系统，这无疑是对我们很好的一次历练机会，同时可以积累丰富的知识。

平时我们很少有机会能够完成一个如此规模的项目，所以在这次项目实战训练中，我的收获也有很多。

第一，我学会了如何规范的编写代码和文档。文档和代码是团队各成员在项目开发交流合作中的纽带，风格随意混乱的代码和缺失或语义不明的文档会让团队中的其他成员不能理解和使用自己写的程序，我在平时编写代码比较随意，并不注重这一点，经历这次实战演练之后，我一定会严格的遵守编码规范。

第二，我学会了如何理解和分析客户的需求。软件为客户而生，只有理解了客户的需求才有写出优秀的软件的可能，在这次实战演练中，老师教会了我们如何组织团队正确的理解和分析客户的需求，并写出完善的需求分析，并用于项目开发。

第三，我学会了对按照需求对软件架构进行设计，这点我想是我们许多同学之前都非常欠缺的，因为我们在学校的时编写代码篇幅短小，功能单一，所以设计思路通常只留于脑内，非常模糊。而在这次项目开发演练中，老师手把手的教会我如何从需求中抽象出问题，并将之模块化，然后逐步求精，就如同为高楼大厦搭建栋梁。

第四，我学会了如何团队合作，共同开发。作为项目开发小组的组长，我的体会更加深刻。独木难支，光靠一个人的力量是远远不够的。只有管理组织好一个团队，才能够最终完成项目的开发，或许在编程技术水平上团队成员有高有低，但是，每个人都有自己的闪光点，总可以在团队的找到适合自己的位置，每个人在自己的团队中做的贡献，创造出不可磨灭的重大价值。

第五，我学会了如何与客户沟通并展示自己的项目，软件项目的开发最终需要的是客户的认可，让客户了解并认可自己的产品最好的办法就是用言语将自己的项目的最好的一面展示出来，在最后一天里，实训的老师组织了项目的答辩，给我们提供了锻炼自己的表达能力，展示自己的项目的舞台。

在实战的过程中，我对项目开发流程有了切实的体会：学会了如何编写规范合格的代码和文档；学会了了正确理解分析客户需求并设计项目的方法；体会到了团队合作精神的重要性，明白了组织团队进行项目开发的具体流程，还了解到了向客户展示产品以及与客户进行沟通的技巧。

经历过这次实习实训，我们不仅在编程的理论知识上有所巩固和深化，更重要的是有了实战进行项目开发锻炼动手能力的机会，积累了一笔宝贵的经验。实训虽已结束，但我们的不断学习的征途才刚刚迈出第一步。今后的路还很长，还有很多机会去实习提高，我们要以更加积极的姿态去充实自己，学习，不断地迎接挑战。

程序设计心得体会篇九

近年来，随着计算机技术的飞速发展，控制器微程序设计在计算机领域中扮演着愈发重要的角色。作为一名计算机专业的学生，在学习和实践中，我深刻体会到了控制器微程序设计的重要性和挑战。下面我将从理论学习、实际操作、团队协作以及总结反思等方面来谈谈自己的心得体会。

首先，理论学习是提高控制器微程序设计能力的基础。在学习的过程中，我发现掌握计算机原理和微程序语言是至关重要的。对于微程序设计来说，理论知识包括硬件结构、指令操作和控制流程等都需要我们理解和掌握。只有通过深入学习这些基本理论，我们才能更好地在实践中运用。因此，我会利用课余时间阅读相关的书籍和论文，通过不断学习来提高自己的理论水平。

其次，实际操作是提高控制器微程序设计能力的关键。纸上得来终觉浅，绝知此事要躬行。在学习过程中，我通过实验平台进行了大量的实践操作。通过实际操作，我不仅加深了对控制器微程序设计的理解，更能够从中感受到设计和调试的乐趣。尤其是在面对各种问题和困难时，通过实践操作，我逐渐培养了解决问题和分析bug的能力，这对于拓宽实践经验和提高技术水平起到了重要的作用。

此外，团队协作是进一步提高控制器微程序设计能力的必要途径。控制器微程序设计往往需要多人合作完成。在实践中，我发现团队合作能够充分发挥每个人的优势，避免重复劳动，提高工作效率。而在团队中，我也学会了与人合作、协调沟通的技巧，进一步培养了团队合作精神。在团队合作中，有时候会碰到分歧和冲突，但通过沟通、妥协和互相理解，我们能够凝聚共识，最终取得良好的团队效果。

最后，总结反思是提高控制器微程序设计能力的重要方法。无论是理论学习过程还是实践经验总结，都需要我们进行反思和总结。通过总结经验，我们能够更好地发现自身的不足和薄弱环节。在我的学习过程中，我会反思自己在设计思路、代码规范和团队协作方面的问题，并及时调整和改进。总结经验对我个人的提高和成长起到了关键的作用。

总之，控制器微程序设计是一门具有挑战性的知识和技能。通过理论学习、实际操作、团队协作以及总结反思，我深刻认识到提高自身能力的重要性，并不断努力去实践和提高自己的。在未来的学习和实践中，我将更加注重理论的深入学习，继续进行实际操作，积极与团队合作，持续总结反思，以提升自己在控制器微程序设计领域的能力。我相信，经过持续努力，我的控制器微程序设计能力一定会得到提高，并为计算机技术的发展做出自己的贡献。

程序设计心得体会篇十

控制器微程序设计是计算机科学中的一个重要领域，它涉及到计算机硬件的控制逻辑编程。在我学习和实践控制器微程序设计的过程中，我深刻体会到了它的重要性和艰巨性。下面我将分享一些我在这方面的体会和心得。

第一段：初始认识

在开始学习之初，我对控制器微程序设计的概念还不是很清晰。我以为它只是简单地将一系列指令存储到控制器中，按顺序执行而已。然而，通过学习，我发现控制器微程序设计涉及到更深层次的逻辑设计和硬件架构。它需要将复杂的指令分解为微操作，并对它们进行编码和存储。这样，控制器才能准确地控制计算机硬件的工作流程。

第二段：挑战与突破

控制器微程序设计的困难之处在于要精确理解并处理各种硬件指令。这些指令通常是复杂而庞大的，涉及到数百乃至数千个微操作。为了应对这一挑战，我深入学习了指令集架构和寄存器的工作原理。我发现，通过合理划分指令和微操作，并充分利用寄存器的功能，能够极大地提高微程序的执行效率。

第三段：优化与验证

在开发控制器微程序的过程中，为了提高执行效率，我尝试了多种优化策略。例如，我使用了流水线技术来并行执行指令，减少了执行时间。另外，我还采用了部分微程序的预取技术，提前将可能用到的微程序载入寄存器，从而避免了取指令的延迟。为了确保微程序的正确性，我还使用了模拟器进行了验证。这样，我能够在开发过程中及时发现和解决问题，确保最终的微程序能够顺利地运行。

第四段：经验总结

通过不断地学习和实践，我逐渐积累了一些关于控制器微程序设计的经验。首先，我认识到了指令设计的重要性。正确和合理地设计指令可以提高微程序的效率和可扩展性。其次，我发现了代码复用的价值。将一些常用的微操作封装成子程序，能够简化微程序的设计和维护。并且，我还体验到了调试的挑战。由于微程序通常较长且复杂，一旦出现错误，找到问题所在并进行修复就变得非常困难。因此，我意识到了编写高质量代码和详细注释的重要性。

第五段：收获与展望

通过学习和实践控制器微程序设计，我获得了很多宝贵的知识和经验。我深入了解了计算机硬件的控制逻辑，并学会了如何编写高效的微程序。这不仅提高了我的编程能力，也让我对计算机体系结构的理解更加深入。在未来，我希望能够

运用这些知识和经验，参与更深入的硬件设计和优化工作中。

总结起来，控制器微程序设计是一项极具挑战性的工作。通过学习和实践，我深入理解了其背后的理论和技术，也积累了一些解决问题的方法和经验。这让我逐渐提升了自己的技术水平，并为未来的发展打下了坚实的基础。控制器微程序设计是计算机科学中值得深入研究和探索的领域，它为我们理解计算机的工作原理和提高硬件性能提供了重要的支持。

程序设计心得体会篇十一

对于我们初次接触项目的人，来说本项目，是我们自学能力的一次测试(有些东西自己从来没有接触过)，也是对我们所学的基础知识的测试。在本程序中我学到了不少的知识，但也发现了很多不足之处。我学到了ssh框架的基本工作流程，并且将自己在学校学到的知识进行了整理。了解了在做项目的时候最重要的并不是写代码，而是去了解我们要做什么，怎么做，对工作流程的理解及需求的分析。自己的不足之处，如：

- 1、自己的知识仅限于在书本上，对项目有一点了解。
- 2、自己的动手能力太差，了解一定的原理但自己写不出来。
- 3、自己的知识有限，还需加强。
- 4、自己的思路不清晰，盲目的去完成项目。
- 5、与团体的合作能力很弱。
- 6、团体的不足之处：

团队的不足如下：

1、在刚做项目的时候都是自己写自己的，没有做到互补(学的好的带着学的不好的人)。

2、在遇到问题的时候没有坐在一起讨论，只管自己解决自己的问题。

以上是我的总结，在本项目中还存在这很多的小问题，还需要进一步的加强自己。

程序设计心得体会篇十二

在这个星期里，我们专业的学生在专业老师的带领下进行了c语言程序实习学习。在这之前，我们已经对c语言这门课程学习了一个学期，对其有了一定的了解，但是也仅仅是停留在了解的范围，对里面的好多东西还是很陌生，更多的在运用起来的时候还是感到很棘手，毕竟，万事开头难嘛。

由于时间的关系，我们的这次实习课程老师并没有给我们详细的介绍，只是给我们简单的介绍了几个比较重要的实际操作。包括了程序模块处理。简单界面程序。高级界面程序。程序的添加修改。用程序做一元线性回归处理以及用c语言程序来画粒度分布图等这几样比较重要的时间操作。

上机实验是学习程序设计语言必不可少的实习环节，特别是c语言灵活、简洁，更需要通过编程的实习来真正掌握它。对于程序设计语言的学习目的，可以概括为学习语法规定、掌握程序设计方法、提高程序开发能力，这些都必须通过充分的实际上机操作才能完成。

学习c程序设计语言除了课堂讲授以外，必须保证有不少于课堂讲授学时的上机时间。因为学时所限，课程不能安排过多的统一上机实验，所以希望学生有效地利用课程上机实验的机会，尽快掌握用c语言开发程序的能力，为今后的继续学习

打下一个良好的基础。为此，我们结合课堂讲授的内容和进度，安排了12次上机实验。课程上机实验的目的，不仅仅是验证教材和讲课的内容、检查自己所编的程序是否正确，课程安排的上机实验的目的可以概括为如下几个方面：

1、熟悉程序开发环境、学习计算机系统的操作方法

一个c语言程序从编辑、编译、连接到运行，都要在一定的外部操作环境下才能进行。所谓“环境”就是所用的计算机系统硬件、软件条件，只有学会使用这些环境，才能进行程序开发工作。通过上机实验，熟练地掌握c语言开发环境，为以后真正编写计算机程序解决实际问题打下基础。同时，在今后遇到其它开发环境时就会触类旁通，很快掌握新系统的使用。

2、学习上机调试程序

完成程序的编写，决不意味着万事大吉。你认为万无一失的程序，实际上机运行时可能不断出现麻烦。如编译程序检测出一大堆错误。有时程序本身不存在语法错误，也能够顺利运行，但是运行结果显然是错误的。开发环境所提供的编译系统无法发现这种程序逻辑错误，只能靠自己的上机经验分析判断错误所在。程序的调试是一个技巧性很强的工作，对于初学者来说，尽快掌握程序调试方法是非常重要的。有时候一个消耗你几个小时时间的小小错误，调试高手一眼就看出错误所在。

通过这次为数不多的几天计算机实习学习，我们了解了一些关于c语言的知识，理解巩固了我们c语言的理论知识，着对我们将来到社会工作将会有莫大的帮助。同时它让我知道，只要你努力，任何东西都不会太难。

3、加深对课堂讲授内容的理解

课堂上要讲授许多关于c语言的语法规则，听起来十分枯燥无

味，也不容易记住，死记硬背是不可取的。然而要使用c语言这个工具解决实际问题，又必须掌握它。通过多次上机练习，对于语法知识有了感性的认识，加深对它的理解，在理解的基础上就会自然而然地掌握c语言的语法规则。对于一些内容自己认为在课堂上听懂了，但上机实习中会发现原来理解的偏差，这是由于大部分学生是初次接触程序设计，缺乏程序设计的实习所致。

学习c语言不能停留在学习它的语法规则，而是利用学到的知识编写c语言程序，解决实际问题。即把c语言作为工具，描述解决实际问题的步骤，由计算机帮助我们解题。只有通过上机才能检验自己是否掌握c语言、自己编写的程序是否能够正确地解题。

通过上机实验来验证自己编制的程序是否正确，恐怕是大多数同学在完成老师作业时的心态。但是在程序设计领域里这是一定要克服的传统的、错误的想法。因为在这种思想支配下，可能你会想办法去“掩盖”程序中的错误，而不是尽可能多地发现程序中存在的问题。自己编好程序上机调试运行时，可能有很多你想不到的情况发生，通过解决这些问题，可以逐步提高自己对c语言的理解和程序开发能力。

程序设计实训心得体会篇3