编号：

|  |  |
| --- | --- |
| 设计报告文档成绩 |  |
| 设计作品分数成绩 |  |
| 课程设计成绩 |  |

**程序设计课程设计文档**

**题 目 ： 《面向PTA实验成绩处理的建模与分析》**

**姓 名 ： 张城硕**

**学 号 ： 202178030243**

**专业班级 ： 计算机类4班**

**年 月 日**

1. **题目意义和设计思想**

|  |
| --- |
| **1、题目意义**  **此问题来源于我们在C语言学习过程中的一个现实的问题， C语言的实验作业都是由老师在PTA平台上发布，让学生在其上进行完成提交的。而每次实验都需要根据每位同学在这个平台上的完成情况来进行评分，并进行排名。**  **所以对于同学们的实验完成情况，通过数学手段给其设计比较合理的评分标准以及参考的指标，并运用数学的语言和方法，通过抽象、简化，建立能近似刻画并"解决"实际问题的数学模型，使用该数学模型来对同学们的实验成绩进行计算，对于每个同学的成绩公平与合理化以及减轻老师的评分压力与工作量都有极大的现实意义。**  **2、设计思想**  **针对PTA实验成绩的处理这一问题，我主要从数学建模这一层面来进行考虑，考虑的关键因素就是数学模型的选择以及建立的方法。**  **由于PTA实验成绩的要素较多，我主要从每次实验的每个同学的题目通过数，完成实验所花费的时间，罚时数和查重率四个方面来进行着手考虑。而每个要素的重要程度并不相同，所以每个要素的权重也并不相同，对于各个因素的权重我采用层次分析法来求解。**  **该方法将定量分析与定性分析结合起来，凭借经验判断各衡量目标能否实现的标准之间的相对重要程度，并合理地给出每个决策方案的每个标准的权数，利用权数求出各方案的优劣次序，比较有效地应用于那些难以用定量方法解决的课题。通过这一方法计算出的每个因素的权重，为之后的实验数据的处理打下了一定的基础。**  **在这之后，由于要考虑四种因素，从而来给每个同学的实验情况进行评分，并进行排名，我采用TOPSIS（优略解距离法）方法来处理数据。TOPSIS 法是一种常用的组内综合评价方法，能充分利用原始数据的信息，其结果能精确地反映各评价方案之间的差距。该方法对数据分布及样本含量没有严格限制，数据计算简单易行。**  **在确定了处理数据所要采用的模型，并且对这些模型的建立方法进行了学习之后，我便通过C语言代码编程来处理数据和实现上述功能。**  **首先进行的就是数据的读入与存储。我采用链表的结构来储存数据，创建了相应的结点，链表，并且对链表进行了初始化。在读入数据方面有两个选择，一个是从文件读入，另一个是手动输入。读取完数据后，便是对数据进行正向化，标准化，归一化，在进行完这些操作之后，便得到了每个同学的归一化后的成绩，之后我又将他们的成绩进行了百分制转换，这些操作都是由一个个独立的函数完成的。最后便是成绩的打印，将成绩按照降序排列，打印出来。**  **当然在后续优化中，我设立了一个菜单，增加了一些其他内容，比如插入数据、删除数据、打印PTA当前数据等，这使得我的系统功能更加的完备，可以满足更多用户的需求。**  **以上，就是我的整个课程设计的思路与所采取的方法。**  **2.1 系统设计流程** |

**二、采用的主要技术、遇到的难点和解决方法**

|  |
| --- |
| **一、采取的主要技术**  **1.层次分析法**  **在计算每个因素的权重时采用了数学建模中的层次分析法。**  **他是通过建立递阶层次结构，把人类的判断转化到若干因素两两之间重要度的比较上，从而把难于量化的定性判断转化为可操作的重要度的比较上面，极大地提高了决策的有效性、可靠性和可行性，但其本质是一种思维方式，它把复杂问题分解成多个组成因素，又将这些因素按支配关系分别形成递阶层次结构，通过两两比较的方法确定决策方案相对重要度的总排序。**  **2.利用Matlab编程求解**  **在用层次分析法的特征值法计算权重的过程中，需要通过Matlab来进行一部分的求解，从而求得特征值计算出权重，并且对比较矩阵进行一致性检验。**  **3.TOPSIS（优略解距离法）**  **在对成绩进行计算的时候采用了数学建模中的TOPSIS方法。**  **TOPSIS 法是一种常用的综合评价方法，其能充分利用原始数据的信息，其结果能精确地反映各评价方案之间的差距。**  **4.数据结构——链表**  **采用链表这一数据结构来储存学生的成绩等信息。**  **5.冒泡排序**  **对于最后的成绩的排名中，运用链表的冒泡排序对成绩进行排序。**  **二、遇到的难点及解决方法**  **1.对于数据读入方面的困难**  **对于数据的读入，我有文件读入和手动输入两种方式供用户选择，由于文件操作不是很熟悉，加之一开始采用绝对路径读入文件，导致经常查找不到文件，所以从文件读入这一块是我整个课程设计的最大难点。对于文件操作，我去网上学习了相关的知识之后，将路径改为了相对路径，并且成功的写出了文件的读取函数。**  **2.将数据百分制的困惑**  **在经过一系列的数据处理之后，得到的最终结果数据十分小，都是0.02左右，所以我考虑将这些数据百分化，但是一开始我用当前成绩/数据中最大成绩\*100，但是这样计算出来的结果，第一名的成绩就总是100分，不符合常理。在一番思索之后，我决定设置一个满分的标准，将其插入链表之中，使其与其他数据一起计算，最终用当前成绩/满分标准成绩\*100，得到百分化的成绩。**  **3.数据存储的选择**  **对于数据的存储而言，由于平时经常用数组来进行数据的存储，加上了解到C语言动态数组的存在，于是一开始我想用数组来储存相关的数据，但最后在了解数据的特点之后，由于一开始数组的大小没有办法确定，所以在一番思考之后，我觉得用链表是处理这个问题的最优解。故最后采用了链表来储存数据。**  **4.对于数据处理的困惑**  **在一开始我对于实验数据的处理并没有十分清晰的思路和方法，十分的苦恼，在查找了相关的资料之后，了解到了数学建模这一便捷有利的方法，最后选择层次分析法和TOPSIS法来处理数据，并且也得到了较好的结果。** |

**三、实现的主要功能和系统结构**

|  |
| --- |
| **一、实现的主要功能**  **1.** **从文件录入学生PTA信息**  **从文件ScoreData.txt中读取各个学生的实验数据，并将它们均插入到链表之中。这是读入数据的第一种方式。**  **2.手动录入学生PTA信息**  **首先通过用户输入确定学生的人数，然后循环调用AddinFormation()函数来插入学生PTA数据信息，这是插入学生PTA数据的第二种方式。**  **3.打印当前学生PTA信息**  **通过定义cur结点，循环遍历链表，打印学生的PTA成绩。**  **4．删除学生PTA信息**  **通过学生学号定位学生信息，通过free()函数将其删除。**  **5.插入学生PTA信息**  **创建一个新的stu链表节点，然后依次从键盘上读取该学生的信息，将其赋给stu结点。最后调用Insert()函数来将这个结点通过头插法插入链表中。**  **6.** **对数据处理并输出成绩**  **首先定义一个满分结点，并且通过用户输入以及在链表中找到各项数据的最优最为满分标准，然后将满分结点插入链表，对数据进行正向化，标准化，然后计算最终得分，将最终得分百分化。最后对成绩进行排序，打印最终的成绩。**    **二、系统结构**  **1.我将代码分成四个部分**  **main.c //存放主函数**  **list.h //主要是链表创建及链表的相关函数以及互动菜单**  **data.h //存放数据处理的相关函数**  **也就是自己创建了两个头文件**  **#include"list.h"**  **#include"data.h"**  **2.数据结构模块：**  **（1）定义一个结构体，来规范存放学生成绩信息。**  **typedef struct Testscore**  **{**  **char ID[13]; //学号**  **char name[20]; //姓名**  **float NumofComplete; //通过数**  **float time; //完成题目所用时间**  **float NumofMistake; //罚时数**  **float RepetRate; //查重率**  **float score; //最终成绩**  **}Testscore;** |

**四、核心算法描述和相关技术说明**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **一、链表的创建**  （1）创建合适的节点来储存数据。我写了一个Testscore的节点。  **节点包含以下内容。**  **typedef struct Testscore**  **{**  **char ID[13]; //学号**  **char name[20]; //姓名**  **float NumofComplete; //通过数**  **float time; //完成题目所用时间**  **float NumofMistake; //罚时数**  **float RepetRate; //查重率**  **float score; //最终成绩**  **}Testscore;**  **该节点用于储存相关的数据。**   1. **接下来的一步是最重要的创建链表。**   **typedef struct StuInfo{**  **Testscore data; //头节点**  **StuInfo\* next; //指向下一个节点**  **int length; //链表的长度**  **}StuInfo;**   1. **建立链表之后需要将链表进行初始化，之后一个空链表便建立好了。**   **要注意第一个头节点要指向空。并且要判断初始化是否成功。**  **开始时链表的长度为0。**  **bool initlink(StuInfo\* &link){**  **link = (StuInfo\*)malloc(sizeof(StuInfo));**  **if (link == NULL)**  **return false;**  **link->next = NULL;**  **link->length = 0;**  **return true;**  **}**  **二、数学建模方法**  **1.层次分析法**   1. **分析系统中各因素之间的关系，建立系统的递阶层次结构。**   **1.1成绩组成的要素wps**   1. **对于同一层次的元素关于上一层次中某一准则的重要性两两比较，构造两两比较矩阵（判断矩阵）。**  |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **题目通过数** | **时间** | **罚时数** | **查重率** | | **题目通过数** | **1** | **6** | **8** | **4** | | **时间** | **1/6** | **1** | **2** | **1/5** | | **罚时数** | **1/8** | **1/2** | **1** | **1/6** | | **查重率** | **1/4** | **5** | **6** | **1** |   **1.2比较矩阵**   |  |  | | --- | --- | | **标度** | **含义** | | **1** | **同样重要性** | | **3** | **稍微重要** | | **5** | **明显重要** | | **7** | **强烈重要** | | **9** | **极端重要** | | **2，4，6，8** | **上述两相邻判断的中值** |   **1.3指标重要程度参考表**   1. **由判断矩阵计算被比较元素对于该准则的相对权重，并进行一致性检验（检验通过权重才能用）。**   **第一步：计算一致性指标CI**    **第二步：查找对应的平均随机一致性指标RI**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | | RI | 0 | 0 | 0.52 | 0.89 | 1.12 | 1.26 | 1.36 | 1.41 | 1.46 | 1.49 | 1.52 | 1.54 | 1.56 | 1.58 | 1.59 |   注：在实际运用中，n很少超过10，如果指标的个数大于10，则可考虑建立二级指标体系  **第三步：计算一致性比例CR**  ***CR*=**  **如果CR<0.1,则可认为判断矩阵的一致性可以接受；否则需要对判断矩阵进行修正。**  **上述表格中的数据，由Matlab计算可得CI=0.0657，CR=0.0739<0.1，故一致性检验通过。**  **（4） 填充权重矩阵，根据矩阵计算得分，得出结果。**  **计算权重的方法有三种：算数平均法求权重、几何平均法求权重以及特征值法求权重。**   |  |  |  |  | | --- | --- | --- | --- | |  | **算术平均法** | **几何平均法** | **特征值法** | | **通过数** | **0.5861** | **0.5999** | **0.6056** | | **完成所用时间** | **0.0858** | **0.0819** | **0.0794** | | **罚时数** | **0.0527** | **0.0515** | **0.0499** | | **查重率** | **0.2754** | **0.2667** | **0.2651** |   **大多数人使用该方法时，都是采用其中一种方法来求权重，而不同的计算方法可能会导致结果有所偏差。为了保证结果的稳健性，我采用了三种方法来分别求出了权重，再根据得到的权重矩阵计算各个方案的得分，并进行排序和综合分析，这样避免了采用单一方法所产生的偏差，得出的结论将更全面、更有效。**  **综上，权重结果采用三种方法的平均值：**   |  |  | | --- | --- | | **通过数** | **0.5972** | | **完成所用时间** | **0.0824** | | **罚时数** | **0.0513** | | **查重率** | **0.2691** |   **2.TOPSIS法**  **（1）将原始矩阵正向化**  **由于除了通过数外的其他三个都是极小型指标，故需要对其进行正向化。采用的方法是**  **Max-X，这样就将他们转化成了极大型指标。**  **正向化代码：**  **void PostChange(StuInfo\*& link, float a[])**  **{**  **StuInfo\* p = (StuInfo\*)malloc(sizeof(StuInfo));**  **p = link->next;**  **//对于不是正向标准的数据，用这类数据的最大值减去各个数据，即可将其正向化**  **for (p; p != NULL; p = p->next)**  **{**  **p->data.time = a[2] - p->data.time;**  **p->data.NumofMistake = a[3] - p->data.NumofMistake;**  **p->data.RepetRate = a[4] - p->data.RepetRate;**  **}**  **}**  **（2）将正向化矩阵标准化**  **标准化的目的：消除不同指标量纲的影响**  **假设有n个要评价的对象，m个评价指标（已经正向过）构成的正向化矩阵如下：**    **那么，对其标准化的矩阵记为Z，Z中的每一个元素：**    **（每个元素/）**  **（3）计算得分并归一化**  **定义第*i*（*i*=1，2，…，*n*）个评价对象与最大值的距离**  **定义第*i*（*i*=1，2，…，*n*）个评价对象与最小值的距离**  **定义最大值：**  **=**  **定义最小值：**  **=**  **在考虑权重的情况下：**  **那么，我们可以计算出第个评价对象为归一化的得分：很明显且越大小，即越接近最大值。**  **我们可以将得分归一化：这样的话（注意：得分归一化不影响排序）归一化之后，再将成绩转化为百分制。得到最终的成绩，打印出来。**  **三、数据的排序**  **在数据的排序这块，我采用的是改编版的冒泡排序。**  **冒泡排序是升序的，我将它改编成为了降序排列。**  **这里的排序牵扯到链表的数据，所以有一定难度。**  **这里我选择用一个sort函数来将这部分功能模块化，更加的便捷形象。**  **在数据比较这方面，我主要比较的就是最终成绩，根据最终成绩的大小来排序。**  **运用了两个for循环。**  **当条件满足时，就交换指针和数据，实现要求。**  **void sort(StuInfo\* &link)**  **{**  **Testscore temp;**  **StuInfo\* p = new StuInfo;**  **p = link->next;**  **StuInfo\* q = new StuInfo;**  **while(p!=NULL){**  **q = p->next;**  **while(q!=NULL){**  **if (p->data.score < q->data.score){**  **//交换数据**  **temp = p->data;**  **p->data = q->data;**  **q->data = temp;**  **}**  **q = q->next;**  **}**  **p = p->next;**  **}**  **}**  **四、数据读入**  **（1）读取数据时，可以从文件中读取。**  **void readStuInfo(StuInfo\* &link){**  **FILE\* fp = fopen("ScoreData.txt", "r");**  **if (fp == NULL){**  **printf("无法打开该文件\n");**  **return;**  **}**  **char ID[13];**  **char name[20];**  **float NumofComplete;**  **float time;**  **float NumofMistake = 0;**  **float RepetRate;**  **float mistake;**  **while (fscanf(fp, "%s %s %f %f", ID, name, &NumofComplete, &time) != EOF){**  **Testscore stu;**  **for (int i = 0; i < NumofComplete; i++){**  **fscanf(fp, "%f", &mistake);**  **NumofMistake += mistake;**  **}**  **fscanf(fp, "%f", &RepetRate);**  **strcpy(stu.ID, ID);**  **strcpy(stu.name, name);**  **stu.NumofComplete = NumofComplete;**  **stu.NumofMistake = NumofMistake;**  **stu.RepetRate = RepetRate;**  **stu.time = time;**  **insert(link, stu);**  **NumofMistake = 0;**  **}**  **fclose(fp);**  **}**   1. **也可以采用手动输入的方法**   **void InputInformation(StuInfo\*& link)**  **{**  **StuInfo\* p, \* r; //p指向新结点，r指向当前链表的尾结点**  **int i, n; //n 为需要输入的学生人数**  **r = link;**  **printf("请输入学生的人数：");**  **scanf("%d", &n);**  **for (i = 1; i <= n; i++) { //通过循环依次插入**  **AddinFormation(link);**  **printf("---------------------------\n");**  **}**  **printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");**  **}**  **五、学生PTA信息删除**  **通过学生学号定位学生信息，以此将其删除。**  **void DeleteInfo(StuInfo\*& link)**  **{**  **StuInfo\* p, \* r;**  **char s[15];**  **r = link;**  **printf("请输入你要删除学生的学号：\n");**  **scanf("%s", s);**  **while ((r->next != NULL) && strcmp(s, r->next->data.ID))**  **r = r->next;**  **if (r == NULL)**  **{**  **printf("删除失败！\n");**  **}**  **p = r->next;**  **r->next = p->next;**  **free(p);**  **printf("删除成功！\n");**  **}** |

**五、总结和体会**

|  |
| --- |
| **通过这次课程设计，我在巩固了许多计算机应用基础知识的同时也学到了很多新的知识，在具体操作中对这学期所学的C语言的理论知识得到巩固加强，达到实训的目的，也发现自己的不足之处，为我们以后的C语言以及其他语言的学习奠定了基础，同时体会到C语言具有的语句简洁，使用灵活，执行效率高等特点。发现上机操作的重要作用，特别是对数组和链表有了深刻的理解。通过实际操作，学会 C语言程序编程的基本步骤、基本方法，开发了自己的逻辑思维能力，培养了分析问题、解决问题的能力。**  **除此之外，由于本课题中的许多知识点都没有学过都要靠自己到课外的资料中去查找，我也学习到了许多课外的知识，比如数学建模的一些知识和技巧，这不仅仅提升了我的思维能力，也为我以后参加相关的比赛奠定了一定的基础。我也在制作课程中从迷茫，不知道该如何下手，到自己慢慢在网上学习相关的知识，同时不断完善代码及其功能，最终完成了该次课程设计。**  **在设计的时候难免出现这样那样的错误。如开始时对于数据的储存并不是很熟悉，并且选择用动态数组来进行，但由于人数的不确定，最后选择更加便捷的链表来储存。在处理数据的过程中也有因为数据类型的问题使得自己的结果全为0的情况，除此之外，对于debug也有了更深层次的认识，对于许多未曾见过的报错，在运用网络查找之后也得到了解决，同时也获得了一些经验。**  **总而言之，我在这次课程设计中受益匪浅，当然，我这次课程设计仍然有一定的缺陷，比如成绩精准度不是很高等的问题，我会在之后的学习生活中不断学习新的知识，争取在以后的实践与运用中解决这些问题。** |