淮阴工学院

**Python程序设计课程设计报告**

**选题名称:** 学生成绩管理系统的设计与实现

**系（院）:** 计算机与软件工程学院

**专 业:** 计算机科学与技术

**班 级:** 计算机1181

**姓 名:** 章磊 **学 号:**1181103114

**指导教师:** 刘作军 张永军

**学年学期:** 2020 ~ 2021 学年 第 1 学期

2020 年 11 月 30 日

**设计任务书**

|  |  |  |
| --- | --- | --- |
| **课题**  **名称** | 学生成绩管理系统的设计与实现 | |
| **设计**  **目的** | 1. 调研并熟悉课题的基本功能、数据流程与工作规程； 2. 学习与课题相关的参考资料和基于Python集成环境的编程技术； 3. 通过实际编程加深对基础知识的理解，提高实践能力； 4. 学习开发资料的收集与整理，学会撰写课程设计报告。 | |
| **实验**  **环境** | 1. 微型电子计算机（PC）； 2. 安装Windows 7.0以上操作系统，Python3.6或者Python3.7开发工具。 | |
| **任务**  **要求** | 1. 利用课余时间去图书馆或上网查阅课题相关资料，深入理解课题含义及设计要求，注意材料收集与整理； 2. 在周末之前完成预设计，并请指导教师审查，通过后方可进行下一步工作； 3. 本课题主要实现对学生成绩进行有效地管理，实现成绩的录入、删除、修改、成绩信息统计等方面的基本操作。 4. 结束后，及时提交设计报告（含纸质稿、电子稿），要求格式规范、内容完整、结论正确，正文字数不少于3000字（不含代码）。 | |
| **工作进度计划** | | |
| **序号** | **起止日期** | **工 作 内 容** |
| **1** | 2020.11.23~2020.11.29 | 在预设计的基础上，进一步查阅资料，完善设计方案，形成书面材料。 |
| **2** | 2020.11.30~2020.12.02 | 设计总体方案，构建、绘制流程框图，编写代码，上机调试。 |
| **3** | 2020.12.03~2020.12.03 | 测试程序，优化代码，增强功能，撰写设计报告。 |
| **4** | 2020.12.04~2020.12.04 | 提交软件代码、设计报告，参加答辩，根据教师反馈意见，修改、完善设计报告。 |
| **指导教师（签章）：** | | |
| **2020 年 11 月 20 日** | | |

摘要：

传统的高校学生成绩管理是靠教职工手工操作,工作效率不高,利用现代计算机信息管理系统管理学生成绩,可以减少工作量,提高工作效率。基于此,设计了一套学生成绩管理系统。首先分析了成绩管理的需求,然后根据分析结果设计系统概念模型,接着设计数据库逻辑结构和物理结构;在此基础上, 录入学生信息时，每个学生信息用一个元组来表示。用一个列表来将学生信息保存到内存中,再将将内存中的学生数据保存到一个csv格式的文件中。设计了学生成绩管理信息系统。该系统实现了课程管理、教师管理、学生管理、选课管理、成绩登记、查询和分析等功能,系统界面友好、简便实用。

关键词：学生成绩管理系统；Python开发；PyCharm；matplotlib库

**目 录**

[1 课题综述 1](#_Toc58444549)

[1.1 需求分析 1](#_Toc58444550)

[2 系统分析 1](#_Toc58444551)

[2.1功能分析 1](#_Toc58444552)

[2.2 总体方案 1](#_Toc58444553)

[2.3 功能模块框图 3](#_Toc58444554)

[3 系统设计： 3](#_Toc58444555)

[3.1 硬件连接（图） 3](#_Toc58444556)

[3.2 算法描述 3](#_Toc58444557)

[3.3 实现方法 4](#_Toc58444558)

[3.4 流程图 4](#_Toc58444559)

[4 代码编写 6](#_Toc58444560)

[5 运行与测试 11](#_Toc58444561)

# 课题综述

## 1.1 需求分析

随着高校招生规模的扩大,学生成绩管理的工作量越来越大。传统的手工操作已经不能满足现代的实际工作需要,容易出现数据丢失、统计错误及效率不高等问题,查找和修改信息更加困难

与传统手工操作相比,利用计算机信息管理系统管理成绩有很大的优势,如查询简便、工作效率高、存储量大,能大幅度减少教职工的工作量,极大地提高工作效率,同时也为学生查询成绩提供了便利,实现了学生“足不出门”就可以查询相关成绩。本文采用Python语言编写,基于PyCharm平台开发了一种学生成绩管理系统,实现了学生信息管理，能够对学生成绩进行有效地管理，实现成绩的录入、删除、修改、成绩信息统计等方面的基本操作。等功能,实现了学生成绩信息管理的规范化、简便化。

# 2 系统分析

## 2.1功能分析

“学生成绩管理系统”包括九个模块：输入学生资料，输出学生资料，学生姓名按顺序排列，添加学生资料，按姓名查找，删除该学生资料，查找并显示学生资料，按姓名查找，修改该学生资料，从文件中读入数据，储存学生资料并退出系统。这九个模块既相互联系又相互独立。

本系统根据学生成绩管理的需要，而建立一个“学生成绩管理系统”，以方便对成绩的各项管理操作。本系统能对成绩进行输入和输出；能按姓名对学生进行排序，并显示学生资料、成绩等，不过得以系统输入学生资料、成绩为前提；能添加学生成绩资料；能根据学生的姓名来查询该学生的成绩资料，并修改或是删除该学生信息；能够从文件中读取学生信息，并且添加到系统中；能把对系统所进行的操作进行保存，以及时更新系统中的数据。

## 2.2 总体方案

管理内容（学生信息）包括：

录入学生信息：

录入学生信息时，每个学生信息用一个元组来表示。用一个列表来将学生信息保存到内存中。

保存学生信息：

可以将内存中的学生数据保存到一个csv格式的文件中，每个学生一行，学生的属性用逗号分隔。

载入学生信息：

从一个指定的csv格式的文件中将学生数据载入到内存中。

修改学生成绩：

对某个学生的某门专业课的成绩进行修改时，可以通过学生学号或学生姓名来查找此学生，再对其成绩进行修改。

删除学生信息：

对某个学生的信息进行删除时，可以通过学生学号或学生姓名来查找此学生，再对其信息进行删除。

计算每位学生的总分和平均分：

计算出每位学生的五门专业课的总分与平均分，以班级为单位，按照总分进行排序并将结果输出显示在屏幕上，每一行打印一个学生的信息，打印格式：学号 姓名 性别 班级 总分 平均分

查询学生信息：

查询某个学生信息时，可以通过学生学号或学生姓名来查找此学生，再将其信息显示在屏幕上，打印格式：学号 姓名 性别 班级 专业课1 专业课2 专业课3专业课4 专业课5

按班级查询学生信息：

按班级查询该班级中的所有学生，并将这些学生的信息显示在屏幕上，打印格式：学号 姓名 性别 专业课1 专业课2 专业课3 专业课4 专业课5

班级成绩的柱状图显示：（\*选做）

以班级为单位，对班级平均分进行分组统计，并采用柱状图的形式显示统计结果。

以班级为单位进行分组，将5门课的班级平均分和总平均分显示在一张柱状图中。

按班级统计学科不及格人数：

以班级为单位进行分组统计，统计班级各个专业课程的不及格人数，并将结果显示在屏幕上，每个班级一行，打印格式：专业课1不及格人数 专业课2不及格人数 专业课3不及格人数 专业课4不及格人数 专业课5不及格人数

退出系统：

程序运行结束，退出系统。

## 2.3 功能模块框图

主

菜

单

输入学生资料模块

输出学生资料模块

按学生姓名进行排列模块

添加学生 资料模块

删除学生资料模块

查找并显示学生资料模块

按姓名查找，修改学生资料模块

从文件中读入数据模块

退出系统模块

图2-1功能模块框图

# 3 系统设计：

## 3.1 算法描述

通过switch语句的执行流程是:首先计算switch后面圆括号中表达式的值,然后用此值依次与各个case的常量表达式比较,若圆括号中表达式的值与某个case后面的常量表达式的值相等,就执行此case后面的语句,执行后遇break语句就退出switch语句。

管理内容（学生信息）包括：

录入学生信息：

录入学生信息时，每个学生信息用一个元组来表示。用一个列表来将学生信息保存到内存中。

保存学生信息：

可以将内存中的学生数据保存到一个csv格式的文件中，每个学生一行，学生的属性用逗号分隔。

载入学生信息：

从一个指定的csv格式的文件中将学生数据载入到内存中。

修改学生成绩：

对某个学生的某门专业课的成绩进行修改时，可以通过学生学号或学生姓名来查找此学生，再对其成绩进行修改。

删除学生信息：

对某个学生的信息进行删除时，可以通过学生学号或学生姓名来查找此学生，再对其信息进行删除。

计算每位学生的总分和平均分：

计算出每位学生的五门专业课的总分与平均分，以班级为单位，按照总分进行排序并将结果输出显示在屏幕上，每一行打印一个学生的信息，打印格式：学号 姓名 性别 班级 总分 平均分

查询学生信息：

查询某个学生信息时，可以通过学生学号或学生姓名来查找此学生，再将其信息显示在屏幕上，打印格式：学号 姓名 性别 班级 专业课1 专业课2 专业课3专业课4 专业课5

## 3.2 流程图

case=2

case=3

case=4

case=5

case=6

case=1

case=7

case=8

case=0

输入学生资料

添加学生资料

按姓名查找，删除该学生资料

查找并显示学生资料

从文件中读入数据

按姓名查找，修改学生资料

储存学生资料并退出系统

按学生姓名进行排列

输出学生资料

yes

no

yes

yes

yes

yes

yes

yes

yes

yes

no

no

no

no

no

no

图3-1系统流程图

# 4 代码编写

# 添加  
def add\_stu():  
 print('\*\*\*\*\*\*\*\*\*添加学员\*\*\*\*\*\*\*\*\*\*')  
 student\_id = input('请输入学员学号：')  
 name = input('请输入学员姓名：')  
 age = input('请输入学员的年龄：')  
 sex = input('请输入学员性别（男/女）：')  
 banji = input('请输入班级：')  
 zhuanyeke1 = input('专业课1：')  
 zhuanyeke2 = input('专业课2：')  
 zhuanyeke3 = input('专业课3：')  
 zhuanyeke4 = input('专业课4：')  
 zhuanyeke5 = input('专业课5：')  
 student = [student\_id, name, age, sex, banji, zhuanyeke1, zhuanyeke2, zhuanyeke3, zhuanyeke4, zhuanyeke5]  
 student\_list.append(student)  
  
  
# 查询学生信息：  
def query\_stu(type):  
 num=input('1.输入学生姓名查询\n2.输入学生学号查询')  
 num=int(num)  
 if num == 1:  
 for x in range(0, len(student\_list)):  
 student = student\_list[x]  
 student\_id = student[0]  
 name = student[1]  
 age = student[2]  
 sex = student[3]  
 banji = student[4]  
 zhuanyeke1 = student[5]  
 zhuanyeke2 = student[6]  
 zhuanyeke3 = student[7]  
 zhuanyeke4 = student[8]  
 zhuanyeke5 = student[9]  
 name\_1 = input('请输入学员姓名：')  
 while 1:  
 rs = False  
 for student in student\_list:  
 if student[1] == name\_1:  
 print('学号：%s 姓名：%s 年龄：%s 性别：%s 班级：%s 专业课1：%s 专业课2：%s 专业课3：%s 专业课4：%s 专业课5：%s ' % (student\_id, name, age, sex, banji, zhuanyeke1, zhuanyeke2, zhuanyeke3, zhuanyeke4, zhuanyeke5))  
 rs = True  
 if rs == False:  
 name\_1 = input('未找到学生，请重输：')  
 else:  
 break  
 else:  
 for x in range(0, len(student\_list)):  
 student= student\_list[x]  
 student\_id = student[0]  
 name = student[1]  
 age = student[2]  
 sex = student[3]  
 banji = student[4]  
 zhuanyeke1 = student[5]  
 zhuanyeke2 = student[6]  
 zhuanyeke3 = student[7]  
 zhuanyeke4 = student[8]  
 zhuanyeke5 = student[9]  
 student\_id\_1=input('请输入学生学号：')  
 while 1:  
 rs = False  
 for student in student\_list:  
 if student[0] == student\_id\_1:  
 print('学号：%s 姓名：%s 年龄：%s 性别：%s 班级：%s 专业课1：%s 专业课2：%s 专业课3：%s 专业课4：%s 专业课5：%s ' % (student\_id, name, age, sex, banji, zhuanyeke1, zhuanyeke2, zhuanyeke3, zhuanyeke4, zhuanyeke5))  
 rs = True  
 if rs == False:  
 student\_id\_1 = input('未找到学员，请重输：')  
 else:  
 break  
 return student  
# 封装判断选择序号是否符合范围的函数  
# 带返回值的函数，函数执行完会返回一个结果，可以声明变量接受这个结果  
def select\_num(type):  
 index = input('请选择要%s的学员序号：' % type)  
 index = int(index)  
 while index not in range(0, len(student\_list)):  
 index = input('选择的学员不存在，请重选：')  
 index = int(index)  
 # 返回选择的序号  
 return index  
  
#λ 按班级查询学生信息  
def all\_stu():  
 banji\_1 = input('请输入想查询的班级：')  
 for x in range(0, len(student\_list)):  
 student = student\_list[x]  
 student\_id = student[0]  
 name = student[1]  
 age = student[2]  
 sex = student[3]  
 banji = student[4]  
 zhuanyeke1 = student[5]  
 zhuanyeke2 = student[6]  
 zhuanyeke3 = student[7]  
 zhuanyeke4 = student[8]  
 zhuanyeke5 = student[9]  
 if student[4] == banji\_1:  
 print('学号：%s 姓名：%s 年龄：%s 性别：%s 班级：%s 专业课1：%s 专业课2：%s 专业课3：%s 专业课4：%s 专业课5：%s ' %(student\_id, name, age, sex, banji, zhuanyeke1, zhuanyeke2, zhuanyeke3, zhuanyeke4, zhuanyeke5))  
  
  
  
# 修改  
def alter\_stu():  
 if len(student\_list) == 0:  
 print('没有学员信息，无法进行修改操作！')  
 # 强制结束函数的执行，return下面的代码都不会再执行  
 return  
 query\_stu('修改')  
 index = int(select\_num('修改'))  
 student = student\_list[index]  
 new\_name = input('请输入修改后的姓名（%s）：' % student[0])  
 new\_age = input('请输入修改后的年龄（%s）:' % student[1])  
 new\_sex = input('请输入修改后的性别（%s）:' % student[2])  
 new\_phone = input('请输入修改后的手机号（%s）:' % student[3])  
 student[0] = new\_name  
 student[1] = new\_age  
 student[2] = new\_sex  
 student[3] = new\_phone  
 print('修改学员成功')  
  
#计算每位学生的总分和平均分  
def allandavg():  
 for x in range(0, len(student\_list)):  
 student = student\_list[x]  
 student\_id = student[0]  
 name = student[1]  
 age = student[2]  
 sex = student[3]  
 banji = student[4]  
 zhuanyeke1 = student[5]  
 zhuanyeke2 = student[6]  
 zhuanyeke3 = student[7]  
 zhuanyeke4 = student[8]  
 zhuanyeke5 = student[9]  
sum=int(zhuanyeke1)+int(zhuanyeke2)+int(zhuanyeke3)+int(zhuanyeke4)+int(zhuanyeke4)+int(zhuanyeke5)  
 pingjun=int(sum)/5  
 print('学号：%s 姓名：%s 年龄：%s 性别：%s 班级：%s 总分: %s 平均分: %s ' % (student\_id, name, age, sex, banji, sum, pingjun))  
 return student  
# 删除  
def dele\_stu():  
 query\_stu('删除')  
 index = select\_num('删除')  
 rs = input('是否真的删除（y/n）:')  
 if rs == 'y':  
 student\_list.pop(index)  
 print('删除成功')  
 else:  
 print('删除数据操作已取消！')  
# 存储至本地文件  
def save\_data():  
 file\_handle = open('2.csv', 'w')  
 for student in student\_list:  
 # 把列表中的数据用 , 分开拼接为一个字符串  
 s = ','.join(student)  
 file\_handle.write(s)  
 file\_handle.write('\n')  
 file\_handle.close()  
  
def bujige():  
 bujige1 = 0  
 bujige2 = 0  
 bujige3 = 0  
 bujige4 = 0  
 bujige5 = 0  
 for x in range(0, len(student\_list)):  
  
 student = student\_list[x]  
 student\_id = student[0]  
 name = student[1]  
 age = student[2]  
 sex = student[3]  
 banji = student[4]  
 if banji=='1181':  
 zhuanyeke1 = student[5]  
 if int(zhuanyeke1)<60:  
 bujige1=bujige1+1  
 zhuanyeke2 = student[6]  
 if int(zhuanyeke2)<60:  
 bujige2=bujige2+2  
 zhuanyeke3 = student[7]  
 if int(zhuanyeke3)<60:  
 bujige3=bujige3+1  
 zhuanyeke4 = student[8]  
 if int(zhuanyeke4)<60:  
 bujige4=bujige4+1  
 zhuanyeke5 = student[9]  
 if int(zhuanyeke5)<60:  
 bujige5=bujige5+1  
 print('专业课1：%s 专业课2：%s 专业课3：%s 专业课4：%s 专业课5：%s' % (bujige1, bujige2, bujige3, bujige4, bujige5))  
 bujige1 = 0  
 bujige2 = 0  
 bujige3 = 0  
 bujige4 = 0  
 bujige5 = 0  
 for x in range(0, len(student\_list)):  
  
 student = student\_list[x]  
 student\_id = student[0]  
 name = student[1]  
 age = student[2]  
 sex = student[3]  
 banji = student[4]  
 if banji == '1182':  
 zhuanyeke1 = student[5]  
 if int(zhuanyeke1) < 60:  
 bujige1 = bujige1 + 1  
 zhuanyeke2 = student[6]  
 if int(zhuanyeke2) < 60:  
 bujige2 = bujige2 + 2  
 zhuanyeke3 = student[7]  
 if int(zhuanyeke3) < 60:  
 bujige3 = bujige3 + 1  
 zhuanyeke4 = student[8]  
 if int(zhuanyeke4) < 60:  
 bujige4 = bujige4 + 1  
 zhuanyeke5 = student[9]  
 if int(zhuanyeke5) < 60:  
 bujige5 = bujige5 + 1  
 print('专业课1：%s 专业课2：%s 专业课3：%s 专业课4：%s 专业课5：%s' % (bujige1, bujige2, bujige3, bujige4, bujige5))  
# 引入python内置函数os  
import os  
# 读取文件内容  
def read\_data():  
 # 判断文件是否存在，如果存在，在做打开文件的操作  
 # 如果文件存在返回true，否则返回False  
 rs = os.path.exists('1.csv')  
 if rs == True:  
 # 1，打开文件  
 file\_handle = open('1.csv', mode='r')  
 # 2.读取所有行  
 contents = file\_handle.readlines()  
 # 3.取出每一个姓名  
 for msg in contents:  
 # 去除\n  
 msg = msg.strip('\n')  
 # 使用 空格分割字符串  
 student = msg.split(',')  
 # 将小列表添加到大列表中  
 student\_list.append(student)  
 #print(student\_list)  
 file\_handle.close()  
 #查询班级信息

# 5 运行与测试

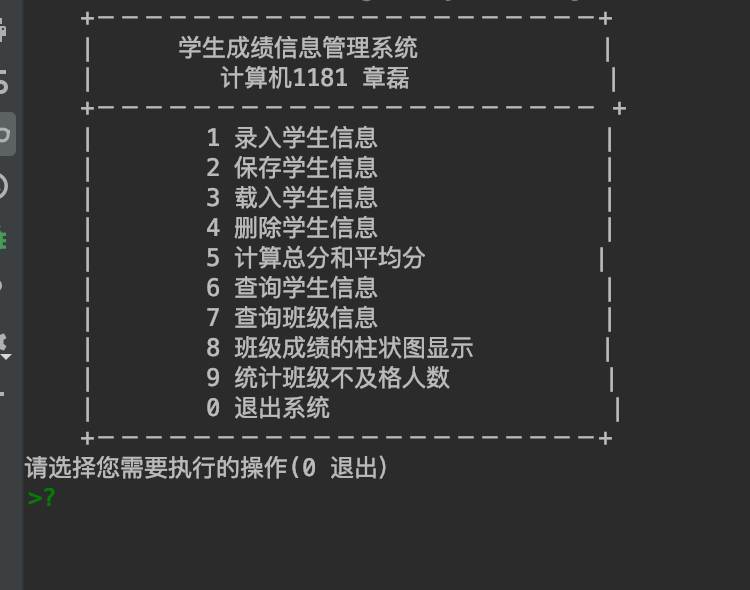


图5-1 系统主界面图

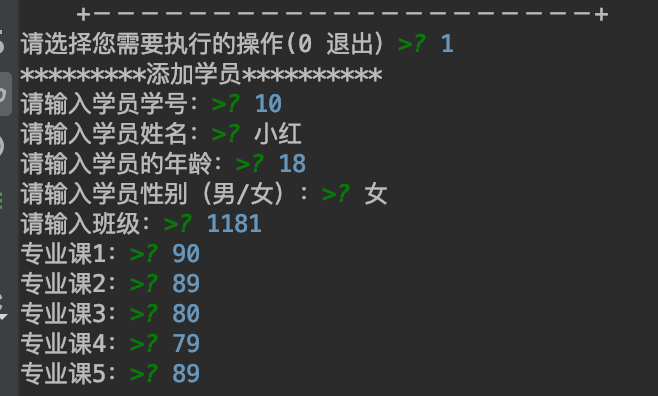


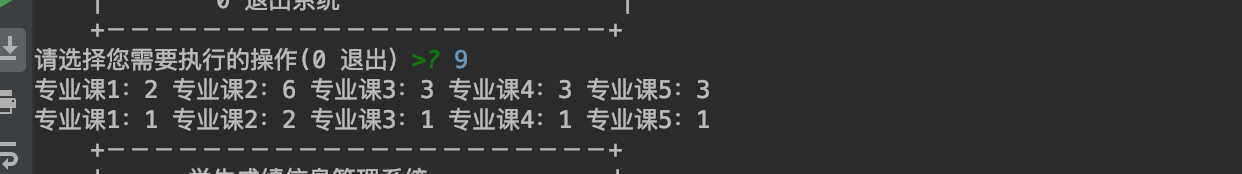
图5-2 添加学生图

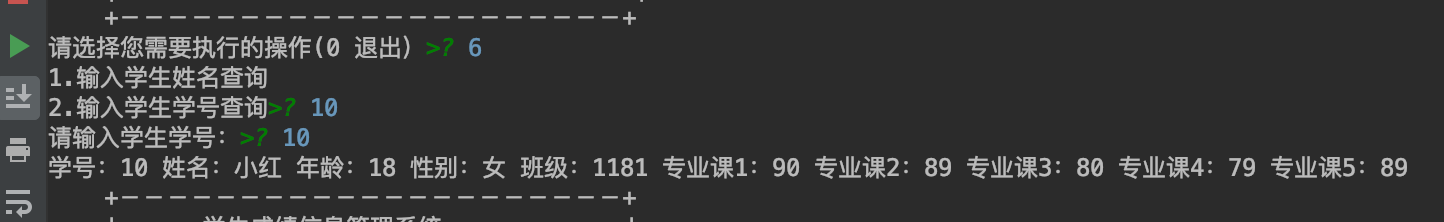
图5-3 平均成绩图

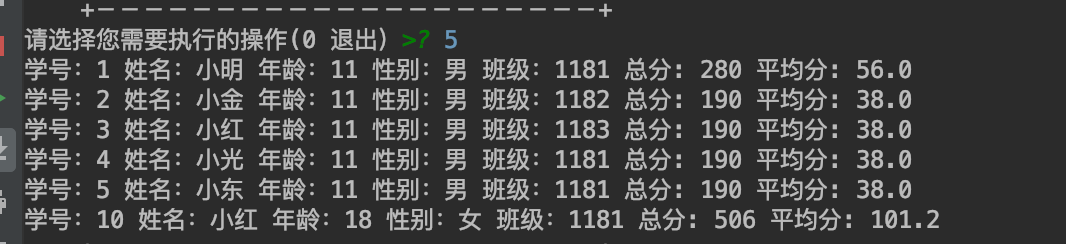
图5-4 学生信息查询图

图5-5 班级查询总分、平均分图

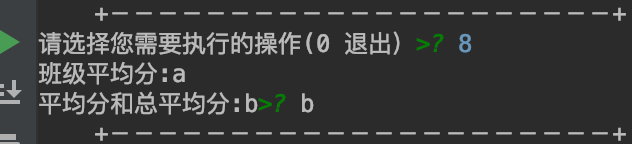


图5-6 总分以柱状图的命令图



图5-7 各项成绩柱状图

总 结

本学生成绩管理系统由九个模块组成，每个模块相互联系又相互独立。这个学生成绩管理系统存在着很多不足之处，由于自己本身编程能力的问题，这个系统可以实现的功能非常有限。只能实现学生管理系统最基本的功能，可以进行简单的输入、输出学生资料，查询和修改学生信息，能直接从文件中读入数据，能对系统所作的修改、操作进行保存等。而且在每个模块也有不足的地方，比如在输入学生资料时，对每个信息项没有严格的规定，可以输入任何的字符。同时有些模块还不能实现本应该有的功能，这有待以后努力，争能够实现该有的功能。

在21世纪的科技时代，科学技术突飞猛进，计算机已经不仅是在科技上应用，而且在生活中也是同样得到了广泛在应用。如今，不管是小学、初中、高中、甚至是很多大学生的学生成绩管理基本上是靠人工进行管理，但随着时间的变化，学校规模的扩大，有关学生成绩管理工作和所涉及到的数据量越来越大越来越多，大多数学校不得不靠增加人力、物力、财力来进行学生成绩管理。但是人工管理成绩档案具有效率低、查找麻烦、可靠性不高、保密性低等因素。因此开发出一个不仅仅是适用于大中专院校以及其它高校通用的学生成绩管理系统是必要的。

写软件我们是照着一定的思想一定思路去写下去，有人说其实写软件并不难，只要给你一个思路你就可以不断地往下写。我觉得确是如此，其实我们平时写软件我觉得并不是很难的事，平时即使遇到处理不了的技术问题，哪里编译出现错误等等其实都可以在网上提问。技术问题一般我们都可以在网上得到答案，但软件的思想不是我们看了一下就能真正理解的。

不论采用哪种方法来开发软件，分析的过程都是提取系统需求的过程。分析工作主要包括3项内容，这就是理解，表达和验证。首先，系统分析员通过用户及领域专家的充分交流，力求完全理解用户需求和该应用邻域中的关键性的背景知识，并用某种无二义性的方式把这种理解表达成文档资料。分析过程得出的最重要的文档资料是软件需求规格说明（在面向对象分析中，主要由对象模型，动态模型和功能模型组成）。

由于问题复杂，而且人与人之间的交流带有随意性和非形式化的特点，上述理解过程通常不能一次就达到理解的效果。因此，还必须进一步验证软件需求规格说明的正确性，完整性和有效性，如果发现了问题则进行修正。显然，需求分析过程是系统分析员与用户及领域。

开发一个学生成绩管理系统，采用计算机对学生成绩进行管理，进一步提高了办学效益和现代化水平。为广大教师和学生提高工作效率，实现学生成绩信息管理工作流程的系统化、规范化和自动化。现在我国的大中专院校的学生成绩档案管理水平普遍都不是很高，有的还停留在全用纸介质基础上，这种管理方式已不能适应时代的发展，社会的需求，因为它浪费了大量的人力物力，也存在着许多不足的因素。在今天信息时代这种传统的管理方法必然会被计算机为基础的信息管理系统所代替。一个高效的学生成绩管理系统可以存储历届的学生成绩档案，不需要大量的人力，只需要几名专门录入员即可操作系统，节省大量人力，可以迅速查到所需信息、高效、安全，学生在能方便的查看自己的成绩。

致 谢

 首先，感谢淮阴工学院、计算机工程系提供的实践机会，实验室人员提供的实验环境。没有学校给我们这么好的环境，我们根本不可能完成这次实训，也不可能收获这么多宝贵的知识，更加不可能获得这么多宝贵的实践经验，和面对挫折勇往直前的学习动态。

另一方面，要在此感激刘老师对我的指导和帮忙，本次论文从选题、方案选择、实际撰写，导师都为我提出了很多的意见和提议，这为我该次论文成功供给了不容忽视的帮忙，在提纲设计、开题撰写的过程中教师多次帮忙我对论述资料进行重新思考，支持其中不连贯的、错误的部分，帮忙我更顺利的完成全文。另外感激学习和写作过程中各位教师和朋友的帮忙，是你们在我需要帮忙的时候供给资源，让我能够顺利的进行资料收集。同样，没有你们的协助我也无法顺利完成本次的问卷调查和很多的问卷表整合。

我必须感谢我的父母。焉得谖草，言树之背，养育之恩，无以回报。作为他们的孩子，我秉承了他们朴实、坚韧的性格，也因此我有足够的信心和能力战胜前进路上的艰难险阻；也因为他们的日夜辛劳，我才有机会如愿完成自己的大学学业，进而取得进一步发展的机会。

还有就是参考文献的相关作者，是你们给了我很多启发和学习的方向。和研究的课题和相关学习目标。感谢同班好友，我们平日里一起的交流和讨论，让我时刻感受到你们的活跃思维和宽阔视野，给予了我在科研方面极大的正能量。

最终再次感激刘师，感激所有支持和帮忙过我的同学和朋友们。

参考文献

[1]郝佳睿.学生信息管理微信小程序数据获取与构建[J].福建电脑,2020,36(11):101-103.

[2]张雪莲.试析Python编程语言的特点及应用[J].电脑编程技巧与维护,2020(11):29-30+33.

[3]郑洪英,盛建强,吴险峰.教学能力大赛促进信息化技术在专业的应用——以高职《python嵌入式编程技术》课程为例[J].品位经典,2020(10):92-93.

[4]承春明,赵欣慧,王建军,孟世斌,赵东坡,韩源.信息管理系统自动巡检平台设计与实现[J].电力信息与通信技术,2020,18(09):105-110.

[5]马婷婷,李浩峰.学生成绩管理系统设计与实现[J].电脑知识与技术,2020,16(24):81-82+88.

[6]吴春梅,蒋林利,余荣川.基于Python和Django框架的二级学院资料室图书管理系统设计与实现[J].无线互联科技,2020,17(16):67-70.

[7]田伟,赵士龙,余佥.Python在海上渔船管理系统中的运用[J].电脑编程技巧与维护,2020(08):94-97.

[8]江志华.学生成绩管理系统的设计与实现[J].信息与电脑(理论版),2020,32(10):79-81.

[9]马赫,冯思度,张红伟.学生成绩管理系统的设计与实现[J].电脑知识与技术,2019,15(06):61-62.

**指导教师评语**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学号 | 1181103114 | 姓名 | 章磊 | 班级 | | 计算机1181 | |
| 选题  名称 | 学生成绩管理系统的设计与实现 | | | | | | |
| 序号 | 评价内容 | | | | 权重（%） | | 得分 |
| 1 | 考勤记录、学习态度、工作作风与表现。 | | | | 5 | |  |
| 2 | 自学情况：  上网检索机时数、文献阅读情况。 | | | | 10 | |  |
| 3 | 论文选题是否先进，是否具有前沿性或前瞻性。 | | | | 5 | |  |
| 4 | 成果验收：  是否完成设计任务；能否运行、可操作性如何等。 | | | | 20 | |  |
| 5 | 报告的格式规范程度、是否图文并茂、语言规范及流畅程度；主题是否鲜明、重心是否突出、论述是否充分、结论是否正确；是否提出了自己的独到见解。 | | | | 30 | |  |
| 6 | 文献引用是否合理、充分、真实。 | | | | 5 | |  |
| 7 | 答辩情况：  自我陈述、回答问题的正确性、用语准确性、逻辑思维、是否具有独到见解等。 | | | | 25 | |  |
| 合计 |  | | | | | | |
| 指导教师（签章）：  年 月 日 | | | | | | | |