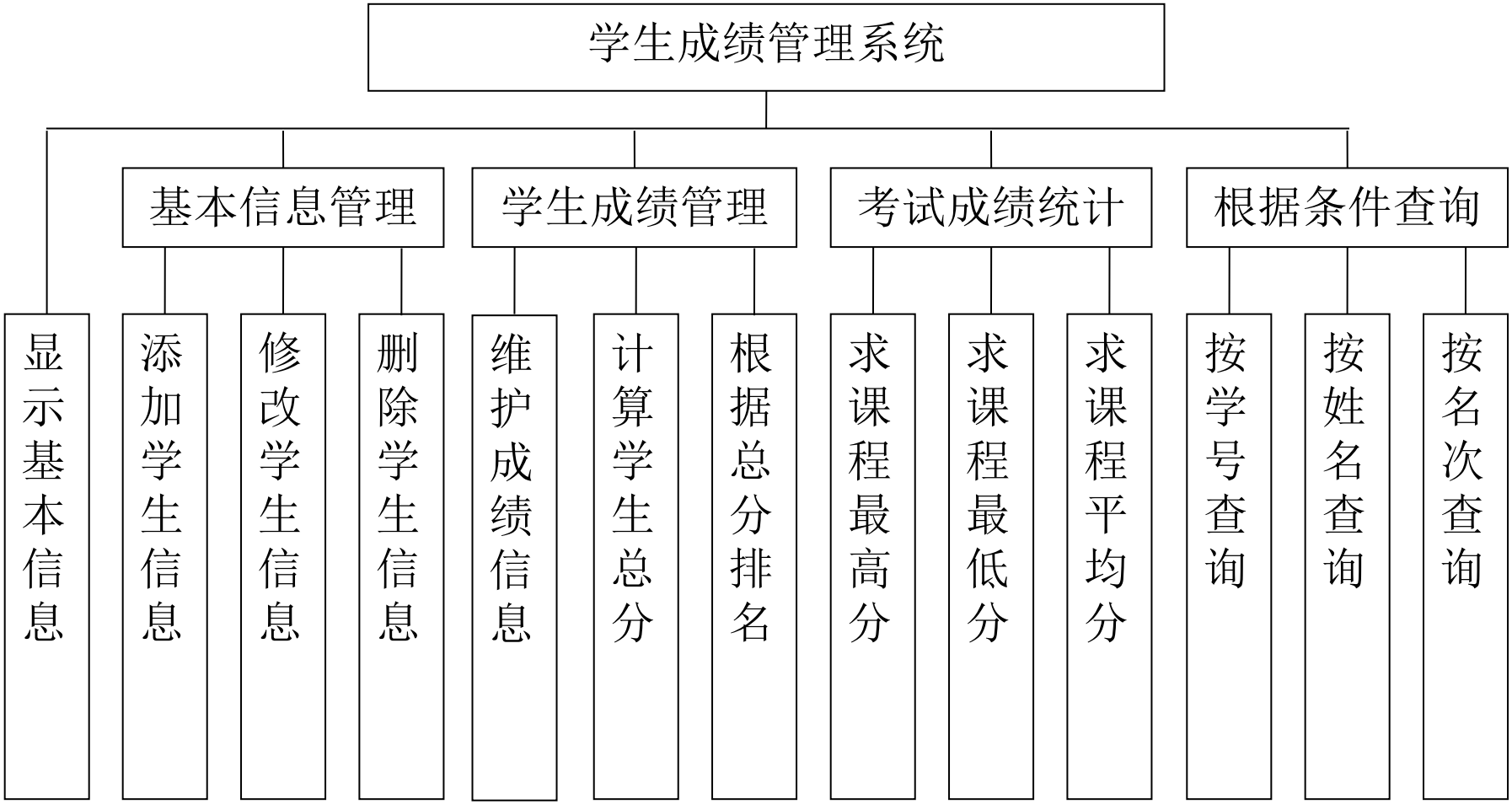
第11章 学生成绩管理系统的设计与实现

学习目标

- 学会使用Python设计并开发一个完整系统
- 能够根据问题的求解需要定义合理的数据结构，设计相应算法
- 掌握包、模块、函数在系统中的实现方法，会合理划分程序

11.1 系统概述

- 一个综合的学生成绩管理系统，要求能够管理若干个学生几门课程成绩，需要实现以下功能：读取以数据文件形式存储的学生信息；可以按学号增加、修改、删除学生的信息；按照学号、姓名、名次等方式查询学生信息；可以按照学号顺序浏览学生信息；可以统计每门课的最高分、最低分和平均分；计算每个学生的总分并进行排名。


学生成绩管理系统
显示基本信息
基本信息管理
添加学生信息
修改学生信息
删除学生信息
学生成绩管理
维护成绩信息
计算学生总分
根据总分排名
考试成绩统计
求课程最高分
求课程最低分
求课程平均分
根据条件查询
按学号查询
按姓名查询
按名次查询

- 为实现该系统，需要解决以下问题：
  1. 数据的表示，用什么样的数据类型能够正确、合理、全面地表示学生的信息，每个学生必须要有那些信息。
  2. 数据的存储，用什么样的结构存储学生的信息，有利于可扩充性并方便操作。
  3. 数据的永久存储，数据以怎样的形式保存在磁盘上，避免数据的重复录入。
  4. 如何能做到便于操作，即人机接口的界面友好，方便使用者的操作。
  5. 如何抽象各个功能，做到代码复用程度高，函数的接口尽可能简单明了。

需要表示的信息	成员名	类型	成员值的获得方式
学号	num	整数	用户输入
姓名	name	字符串	用户输入
性别	gender	字符串	用户输入
3门课程的成绩	score	列表	用户输入
总分	total	整数	根据3门课成绩计算
名次	rank	整数	根据总分计算

```
class Student(object):                          #学生记录数据域
        def
__init__(self,num=0,name='',sex='',score=[0,0,0],total=0,rank=0):
                self.num = num                  #学号
                self.name = name                        #姓名
                self.sex = sex                          #性别
                self.score = score                      #3门课成绩
                self.total = total              #总分
                self.rank = rank                #名次
```

为学生类型定制的基本操作

- def readStu(stu,n=100):
- def printStu(stu,n=100):
- def equal(s1,s2,condition):
- def larger(s1,s2,condition):
- def reverse(stu):
- def calcuTotal(stu):
- def calcuRank(stu):
- def calcuMark(m,stu,n):
- def sortStu(stu,condition):
- def searchStu(stu,s,condition,f):
- def deleteStu(stu,s):

- 需要对其中的部分函数再作一些说明：
  1. readStu( )和 printStu( )函数都是实现读入或输出n个元素的，当实参n为1时，该函数的功能就是读入或输出一个记录，依然是可以正确执行的，在后续的程序中有时需要对单个记录进行输入/输出处理，有时是批量的输入/输出，这两个函数适用于这两种需求。
  2. equal 函数中的形式参数 condition，是为了使函数更通用。因为程序中需要用到多种判断相等的方式：按学号、按分数、按名次、按姓名，没有必要分别写出4个判相等的函数，所以用同一个函数实现，通过 condition 参数来区别到底需要按什么条件进行判断，简化了程序的接口。

3. larger 函数中形式参数 condition 的用法和意义与equal函数中相同，在程序中进行排序时主要是根据学号或分数进行，因此本函数中 condition 的取值只定义了两种，读者在实现程序时，如果还有其他需要判断大小的情况，则增加 condition 变量的选值就可以了
4. calcuRank 函数用来计算所有同学的名次，本函数中，要充分考虑相同总分的同学名次相同，并且在有并列名次的情况下，后面同学的名次应该跳过空的名次号。例如，有两个同学并列第5，则下一个分数的同学应该是第7名而不应该是第6名，这在赋值的时候用双分支 if 来控制。

5. searchStu 函数用来实现按一定条件的查询，该函数将被查询模块调用，查询的依据有学号、姓名、名次，本系统中，只有按学号查询得到的结果是唯一的，因为在进行插入、删除等基本信息的管理时已经保证了学号的唯一性。按姓名及名次查询都有可能得到多条记录结果，因此，该函数中用 f 数组来存储符合条件的记录的下标，通过此参数将所有查询下标返回给主调用函数，从而才得出查询后所有符合条件的结果。函数的返回值是符合查询条件的元素个数，这样便于主调用函数控制数组输出时的循环次数。
6. 函数 calcuMark 用来求三门课程的最高分、最低分、平均分，共有9个信息，因此形式参数表中用一个二位数组来返回这9个求解的结果，第一下标代表哪门课，第二下标的0、1、2分别对应于最高分、最低分、平均分。

数据的永久保存

- def createFile(stu):
- def readFile(stu):
- def saveFile(stu):

用两级菜单四层函数实现系统

菜单	主菜单	基本信息	成绩管理	成绩统计	条件查询
函数名	def menu()	def menuBase()	def menuScore()	def menuCount()	def menuSearch()
对应功能模块	学生成绩管理系统	基本信息管理	学生成绩管理	学生成绩统计	根据条件查询
被哪个函数调用	main函数	baseManage函数	scoreManage函数	countManage函数	searchManage函数

- 在编制上述程序时，请注意以下几点：
  1. 如果第一次运行，数据文件是空的，则会自动调用建立文件这个函数，用户需要从键盘上先输入一系列元素，程序执行保存操作。
  2. 在运行插入、删除、修改之后，一定要注意，必须选择第三个一级菜单功能，即"3.学生成绩管理"功能，并且重新选择其下的两个子菜单分别计算总分和排名，才是目前对基本信息进行修改之后最新的成绩与排名情况。在查询时，根据姓名和名字查询都有可能显示多条记录，上述演示中就有查名次显示出来并列名次的两条记录信息。

3. 每一级菜单函数都放在循环体中调用，目的是使得每一次操作结束后，重新显示菜单。该系统的功能划分还可以有其他的方法，请读者自行设计其他的方案，并仿照此程序的实现方法，自己设计一个类似的信息管理系统。
4. 在开发一个系统的时候，一定要考虑数据的存储问题，因为每次运行原始数据都从键盘读入是不科学也是不可行的，因此文件的操作非常重要，对应于对象列表或字典类型的数据用文本文件更为直观，用户也可根据需要选择使用二进制形式进行文件保存。

5. 对系统要实现的功能按自顶向下、逐步细化、模块化的思想进行结构化设计是非常重要的。每一个功能用一个或多个函数对应实现，在设计时充分考虑对功能的抽象，如何定义函数，使函数的功能更加通用，能为多个功能提供服务。函数与函数之间怎样传递数据，即参数和返回值类型如何正确设定。每一个函数的功能必须清晰、代码简洁明了，这是系统设计中非常重要的问题。

6. 另外，友好的人机交互界面，将大大方便使用者，这也是系统设计时需要考虑的问题。因为开发者一定要将使用者理解为完全不懂程序，只是在一个易操作的界面指导下使用程序完成特定功能。因为，菜单设计要清晰合理，程序中的意外错误也要有充分的考虑，提示信息要丰富完整。


输入理想的程序
输出快乐的人生