学生选课信息管理系统

设计报告

WELCOME

程序设计基础实验

191220008
陈南曈

目录

REPORT

需求分析

01 需求分析

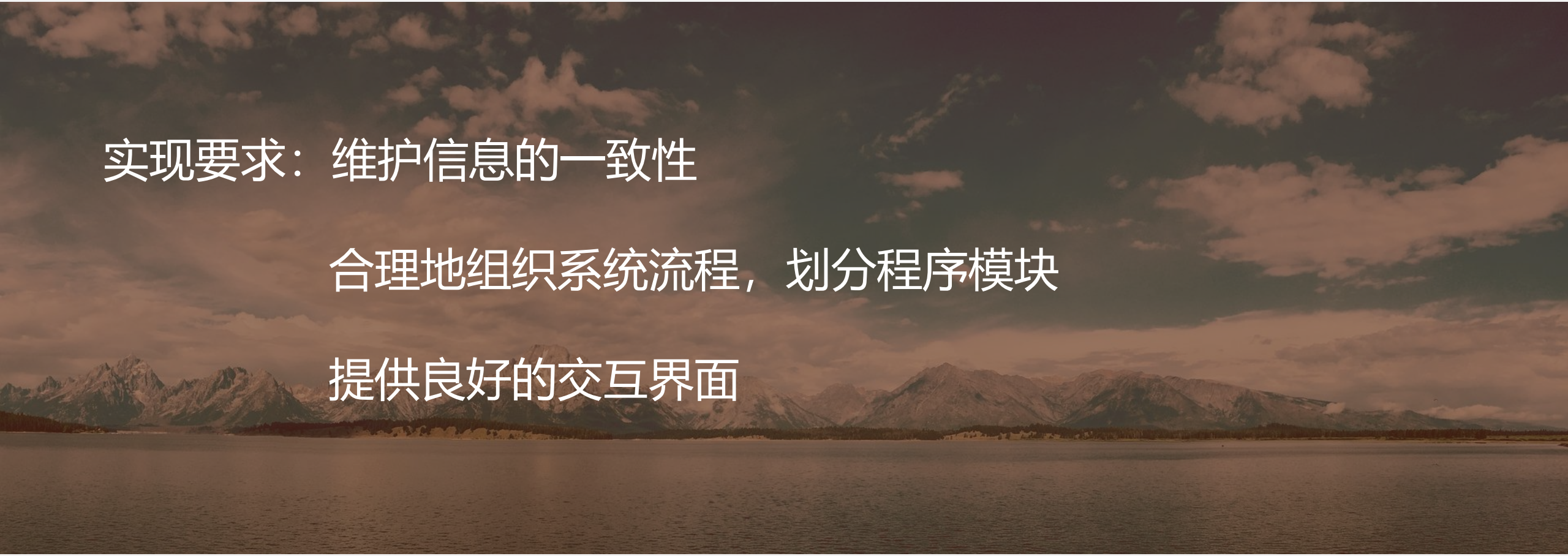
对象：管理员与学生

基本功能：

管理员：系统管理：用户登录、注册、注销
课程管理：课程信息录入、增加、修改、删除、查看全部课程、查看具体课程（选课学生名单、助教名单等）、

学生：系统管理：用户登录、注册、注销
课程管理：选课、查看个人课表、退课、查看课程信息
助教管理：举手报名、选择个人助教

01 需求分析



实现要求：维护信息的一致性

合理地组织系统流程，划分程序模块

提供良好的交互界面


数据结构

02 数据结构

学生注册信息：学生ID、学生密码、课程编号、个人助教……

课程信息：　　课程编号、课程名称、授课教师、容纳人数、目前已选、课程类型……

课程学生名单：学生ID

课程助教名单：助教ID、助教数目

02 数据结构

学生信息

```
struct Student
{
    char id[21];                          //学生ID
    char password[21];                    //ID密码
    char question[50];                    //找回密码问题
    char answer[50];                      //找回密码问题答案
    char assistant_course[10][4];         //自己当助教的课程
    int assistant_course_amount;          //自己当助教的课程数目
    char course[10][4];                   //选择的课程
    int course_amount;                    //选择的课程数目
    char course_assistant[10][20];        //选择的课程的个人助教
    char assistant_delete[10][4];         //个人助教退的课程
    int assistant_delete_flag;            //退课或撤销的个人助教数目
                                            （登录查阅后清零）
    int wrong_times;                      //密码输入错误的次数
    Student *next;                        //指向下一名学生
    //int admission;                        判断个人助教是否同意
};
```

02 数据结构

课程信息

```
struct Course
{
    char number[4];                 //课程编号
    char name[30];                  //课程名称
    char teacher[20];               //授课老师
    int capacity;                   //容纳人数
    int amount;                     //已选人数
    char type[5];                   //课程类型
    int assistantamount;            //助教数目
    char student[100][20];          //学生名单
    char assistant[100][20];        //助教名单
    Course *next;                   //指向下一门课程
};
```

模块划分

03 模块划分

main.cpp

MODULE 1

Login.h
+
Login.cpp

MODULE 2

AdministratorWindow.h
+
AdministratorWindow.cpp

MODULE 3

StudentWindow.h
+
StudentWindow.cpp

MODULE 4

03 模块划分

main.cpp

MODULE 1

· 主程序入口

·定义课程头指针、学生头指针、学生尾指针、管理员登录判断

· 调用开始函数：进入welcome函数和begin函数

03 模块划分

main.cpp

MODULE 1

```cpp
#include "Login.h"
#include "AdministratorWindow.h"
#include "StudentWindow.h"

Course* head = NULL;
Student* stuhead = NULL;
Student* stutail = NULL;
int administrator_first = 0;

int main()
{
    welcome();
    begin();
    return 0;
}
```

03 模块划分

Login.h
&
Login.cpp

MODULE 2

- 显示系统主界面
- 显示登录界面
- 实现学生端和管理员端接口：
  - 选择学生端（登录/注册）或者管理员端（登录）
  - 用ID/密码验证
- 退出系统
- 拓展功能：
  - 密文登录
  - 密码输入错误5次后锁定用户
  - 修改密码
  - 修改密保
  - 找回密码
  - 返回上一级
  - ……

03 模块划分

Login.h
&
Login.cpp

MODULE 2

```
15    //功能函数
16   void welcome() { ... }
45   void begin() { ... }
58   int login() { ... }
113  void student_login() { ... }
183  void student_register() { ... }
287  void administrator_login() { ... }
392  void password_change() { ... }
417  void question_set_or_change() { ... }
442  void password_retrieve() { ... }
467  void logdown() { ... }
472
473   //辅助函数
474  bool match_account(char stu_id[], char stu_password[]) { ... }
526  bool format_check(char stu_id[], char stu_password[]) { ... }
573  bool duplicat_check(char stu_id[]) { ... }
607  void register_account() { ... }
625  void student_password_change() { ... }
703  void administrator_password_change() { ... }
771  void student_question_set_or_change() { ... }
851  void administrator_question_set_or_change() { ... }
923  void student_password_retrieve() { ... }
1016 void administrator_password_retrieve() { ... }
```

03 模块划分

AdministratorWindow.h
&
AdministratorWindow.cpp

MODULE 3

- 显示管理员端功能选项界面
- 实现管理员端主要功能：
  - 录入课程信息
  - 查看全部课程信息
  - 增加课程
  - 删除课程
  - 修改课程（上限人数、授课老师）
  - 查看具体课程（学生名单、助教名单）
  - 注销登录
- 拓展功能：
  - 查看特定助教下的学生名单
  - 同时添加多门课程
  - 查看密保
  - 返回上一级
- 未拓展功能
  - 启动抽签选课
  - 查看消息栏
  - 同时删除多门课程
  - ……

03 模块划分

AdministratorWindow.h
&
AdministratorWindow.cpp

MODULE 3

```
5    //功能函数
6   void administrator_window() { ... }
82  void course_typein() { ... }
102 void allcourse_view() { ... }
135 void course_add() { ... }
236 void course_delete() { ... }
283 void course_change() { ... }
316 void specificcourse_view() { ... }
412 void administrator_security_check() { ... }
423 void administrator_logout() { ... }
432
433   //辅助函数
434 void course_get(char filename[]) { ... }
470 void course_update() { ... }
489 bool course_duplicatcheck(Course*p) { ... }
503 void teacher_change(char id[]) { ... }
526 void capacity_change(char id[]) { ... }
554 void student_view(Course* p) { ... }
587 void assistant_view(Course* p) { ... }
621 void student_of_assistant_view(Course*p) { ... }
687 bool assistant_check(Course*p,char assistant_id[]) { ... }
```

03 模块划分

StudentWindow.h
&
StudentWindow.cpp

MODULE 4

· 显示学生端功能选项界面

· 实现学生端主要功能：
  · 查看课程信息
  · 选择课程
  · 查看个人课表
  · 退出课程
  · 举手报名助教
  · 选择个人助教
  · 注销登录（判断选课是否符合要求，不符合将会给出提示）

· 拓展功能
  · 同时选择多门课程
  · 撤销助教报名（该助教下的学生登录后显示提示）
  · 更换助教和取消助教
  · 返回上一级

· 可能拓展功能：
  · 选择抽签式课程
  · 选择助教需经过助教本人同意
  · 查看消息栏
  · ……

03 模块划分

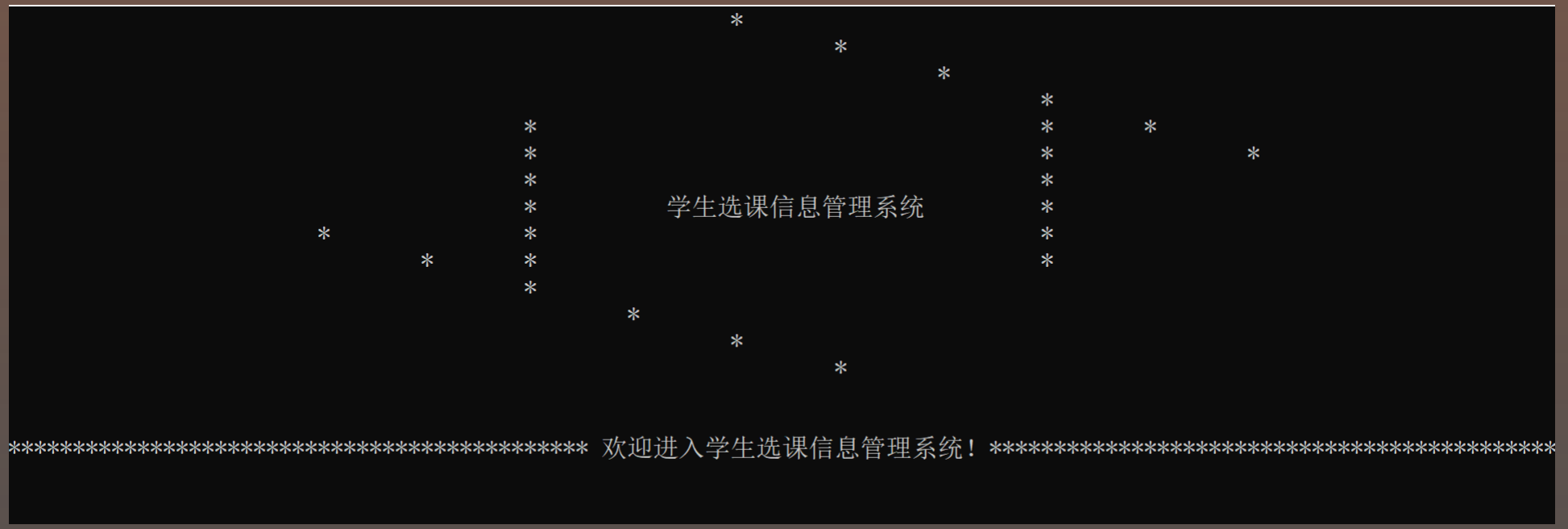
StudentWindow.h
&
StudentWindow.cpp

MODULE 4

```
5     //功能函数
6   ⊞void student_window() { ... }
65  ⊞void course_view() { ... }
98  ⊞void course_select() { ... }
187 ⊞void student_course_view() { ... }
237 ⊞void student_course_drop() { ... }
319 ⊞void assistant_apply_or_drop() { ... }
343 ⊞void my_assistant() { ... }
369 ⊞void student_security_check() { ... }
383 ⊞void student_logout() { ... }
415
416   //辅助函数
417 ⊞void assistant_drop_check() { ... }
433 ⊞void course_upsort(int course_amount,char course[][4],char course_assistant[][20]) { ... }
455 ⊞void student_course_update() { ... }
473 ⊞void course_update2() { ... }
490 ⊞bool student_course_duplicatcheck(char id[],Student*s) { ... }
499 ⊞void assistant_drop_remind(char id[]) { ... }
520 ⊞void assistant_apply() { ... }
570 ⊞void assistant_drop() { ... }
635 ⊞void assistant_choose() { ... }
715 ⊞void assistant_change() { ... }
798 ⊞void assistant_cancel() { ... }
850 ⊞void assistant_update() { ... }
877 ⊞void otherstudent_course_update(char otherstudent_name[]) { ... }
```


界面设计

04 界面设计



主界面

04 界面设计


C:\南\程序设计基础实验\第四~七周\Project1\Debug\Project1.exe
学生选课信息管理系统
********** 欢迎进入学生选课信息管理系统！**********
1、学生登录 2、学生注册 3、管理员登录 4、修改密码 5、密保功能 6、找回密码 7、退出系统
请输入操作序号：

系统界面

04 界面设计

管理员界面

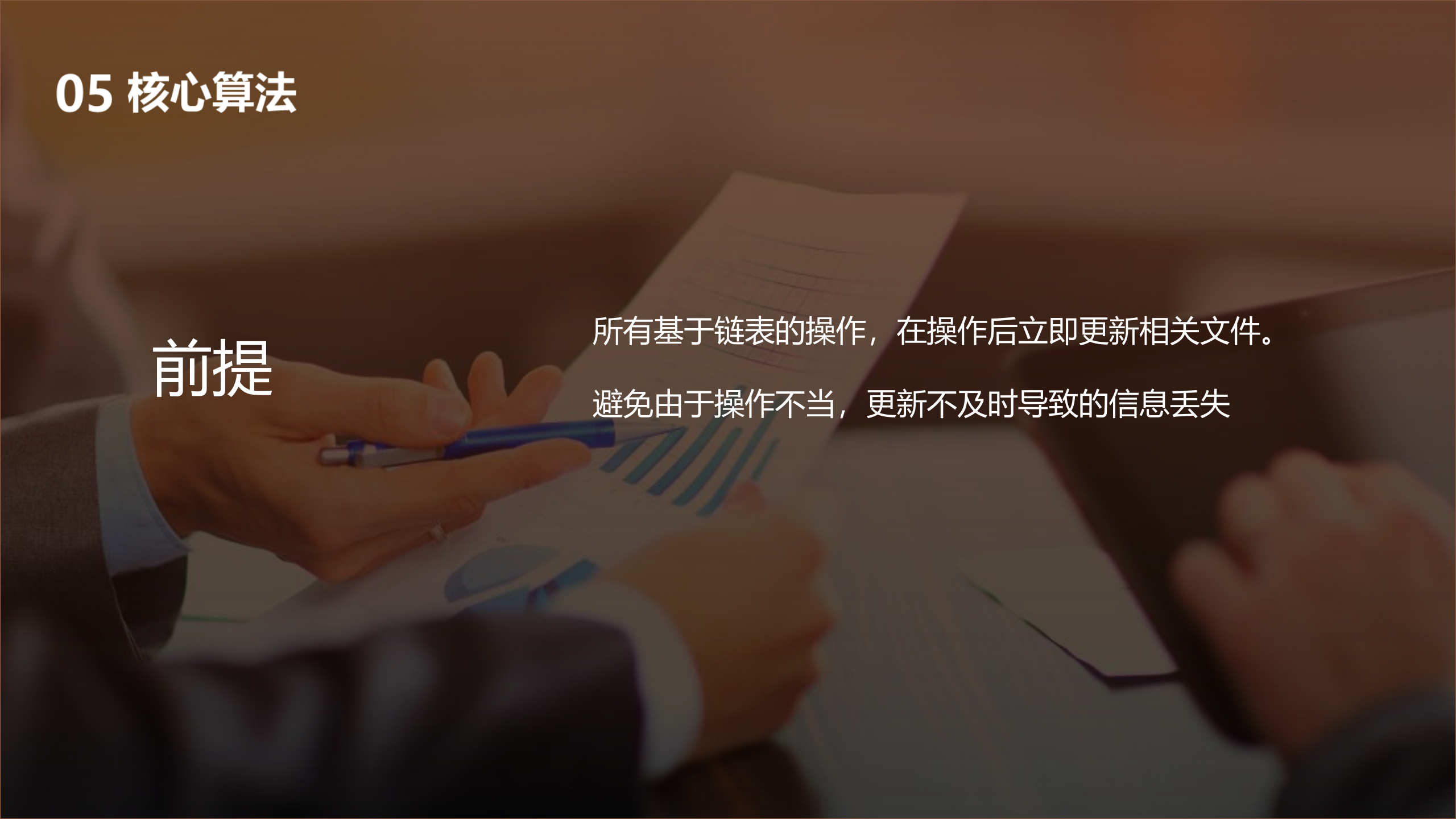
04 界面设计


C:\南\程序设计基础实验\第四~七周\Project1\Debug\Project1.exe
1、学生登录 2、学生注册 3、管理员登录 4、修改密码 5、密保功能 6、找回密码 7、退出系统
请输入操作序号：2
（ID与密码只能含有小写字母和数字，且不超过20个字符！）
请输入学生ID（输入'*'返回上一级）：191220008
请输入学生个人密码（输入'*'返回上一级）：123456
（若不设置密保问题，则ID被锁定时需先在“修改密保”中设置密保后才能找回密码！）
是否设置用于找回密码的密保问题（1、是 2、否）：2
********** 注册成功！并成功登录学生端！**********
1、查看课程信息 2、选择课程 3、查看个人课表 4、退出课程 5、助教报名/撤销 6、我的个人助教 7、查看密保 8、注销登录
请输入操作序号：

学生界面


核心算法

05 核心算法



前提

所有基于链表的操作，在操作后立即更新相关文件。

避免由于操作不当，更新不及时导致的信息丢失

05 核心算法

两大链表

算法描述：

```
5     Course* head = NULL;
6     Student* stuhead = NULL;
```

主要功能均基于课程链表和学生链表这两大链表实现

通过用课程编号对某门课程进行查找，并进行操作

```
8     char stu_id[21];
9     char stu_password[21];
10    char filename2[30];
```

每次学生登录后将该学生ID保存在全局变量stu_id中

此后对该名学生的操作，均通过在链表中查找该stu_id来实现

将stu_id对应的学生文件名存至filename2，便于每次文件更新

05 核心算法

助教退课

算法描述：

```
16      char assistant_delete[10][4];
17      int assistant_delete_flag;
```

在Student结构中：
用assistant_delete存储个人助教退课的课程
用assistant_delete_flag存储个人助教退课的数目

05 核心算法

助教退课(和撤销)

算法描述：

助教退课时，用assistant_drop_remind函数提醒学生

assistant_delete_flag会加1，course_assistant[i]置为”NULL”
同时将退课的课程编号存储至assistant_delete

```
499  void assistant_drop_remind(char id[])
500  {
501      Student* s = stuhead;
502      while(s)
503      {
504          for (int i = 0; i < s->course_amount; i++)
505          {
506              if (!strcmp(s->course[i], id))
507              {
508                  if (!strcmp(s->course_assistant[i], stu_id))
509                  {
510                      strcpy(s->course_assistant[i], "NULL");
511                      strcpy(s->assistant_delete[s->assistant_delete_flag], id);
512                      s->assistant_delete_flag++;
513                      otherstudent_course_update(s->id);
514                  }
515              }
516          }
517          s = s->next;
518      }
519  }
```

05 核心算法

助教退课(和撤销)

算法描述：

学生每次登陆后，用assistant_drop_check函数判断是否有助教退课，如果有则显示

显示后，将assistant_delete_flag重新置为0，assistant_delete清空

```
417  void assistant_drop_check()
418  {
419      Student* s = stuhead;
420      while (strcmp(s->id, stu_id))
421          s = s->next;
422      if(s->assistant_delete_flag)
423      {
424          for (int i = 0; i < s->assistant_delete_flag; i++)
425          {
426              cout << "您选择的 " << s->assistant_delete[i] << " 课程的个人助教已退课或已撤销助教！" << endl;
427              strcpy(s->assistant_delete[i], "\0");
428          }
429          cout << endl;
430          s->assistant_delete_flag = 0;
431      }
432  }
```

05 核心算法

密码管理系列

功能一：密码输入错误5次后锁定用户

算法描述：

```
18      int wrong_times;
```

学生：在Student结构中：
用int wrong_times来存储密码输入错误的次数

```
10      int admin_wrong_times = 0;
```

管理员：定义 int admin_wrong_times全局变量
来存储密码输入错误的次数

密码每输入错误一次，（admin_)wrong_times自增一次
在每次的登录或找回密码后，次数重新置为0

05 核心算法

密码管理系列

功能二：修改密码

算法描述：

输入ID和密码来验证身份

学生：通过ID在Student链表中找到该名学生，将输入的密码与该节点中的密码比对，若相同，则允许修改密码，并将修改后的密码存入该节点中以替换之前的密码

管理员：将输入的ID和密码与全局变量my_id和my_password进行比对，若相同，则允许修改密码，并将修改后的密码存入my_password以替换之前的密码

密码仍会进行格式判断

05 核心算法

功能二：修改密码

密码管理系列

```
392  void password_change()
393  {
394      cout << "请输入您想修改密码的身份（1、学生 2、管理员 3、返回主界面）：";
395      int choice = 0;
396      cin >> choice;
397      while (1)
398      {
399          if (choice >= 1 && choice <= 3)
400              break;
401          else
402          {
403              cout << "您输入的操作序号不正确，请重新输入！" << endl << endl;
404              cout << "请输入您想修改密码的身份（1、学生 2、管理员 3、返回主界面）：";
405              cin >> choice;
406          }
407      }
408      cout << endl << endl;
409      switch (choice)
410      {
411      case 1: student_password_change(); break;
412      case 2: administrator_password_change(); break;
413      case 3: begin(); break;
414      }
415      begin();
416  }
```

05 核心算法

密码管理系列

功能三：密保功能

算法描述：

```
9     char question[51];
10    char answer[51];
```

学生：在注册时可以设置密保，在Student结构中，用question和answer存储密保问题和答案，初始化成”NULL”

```
6     char my_question[51] = "NULL";
7     char my_answer[51] = "NULL";
```

管理员：在登录时可以设置密保，用全局变量my_question my_answer存储密保问题和答案，初始化成”NULL”

05 核心算法

密码管理系列

功能三：找回密码

算法描述：

学生：输入ID，在Student链表中找到该学生节点，输出对应的密保问题，若学生的答案与存储的密保答案一致，则输出该ID的密码，并提供继续修改密码的功能

管理员：输入ID，输出my_question中的密保问题，若输入的答案与存储的密保答案my_answer一致，则输出管理员的密码，并提供继续修改密码的功能

未设置密保的用户需现在“密保功能”中设置密保后才能找回密码

成功找回密码后，将密码输入错误次数重置为0，解锁用户

05 核心算法

密码管理系列

功能四：查看密保

算法描述：

学生：在Student链表中找到该学生的节点，输出其密保问题和答案

管理员：直接输出my_question和my_answer

05 核心算法

抽签选课（未实现）

大致思路描述：

由管理员启动某抽签类课程的抽签函数，在总人数范围内通过伪随机数函数生成数字，生成的数字即为抽中学生在注册信息系统列表中的位置。

同时，管理员将该课程由可选改为不可选，已关闭该课程选课通道。

抽签结束后，将抽签结果发送至报名该课程的学生消息栏中。

学生选择该课程时，不受容纳人数的限制，但必须已选择至少4门专业课和2门选修课才能选择抽签的课程，且最多选择1门。

为实现上述功能，将在课程信息中增添选项：选课方式（抽签/非抽签）、课程状态（可选/不可选）

05 核心算法

消息栏(未实现)

大致思路描述：

该系统内可接受的消息：抽签课程结果、选择助教结果

抽签课程结果：抽签成功，管理员将该课程录入抽中学生课表链表节点，同时int message变为1，学生每次登录后，判断int message 是否为1，为1则通过学生个人课表判断是否生成抽签课程，来直接显示抽签结果，同时int message置为0。打开消息栏也可查看。

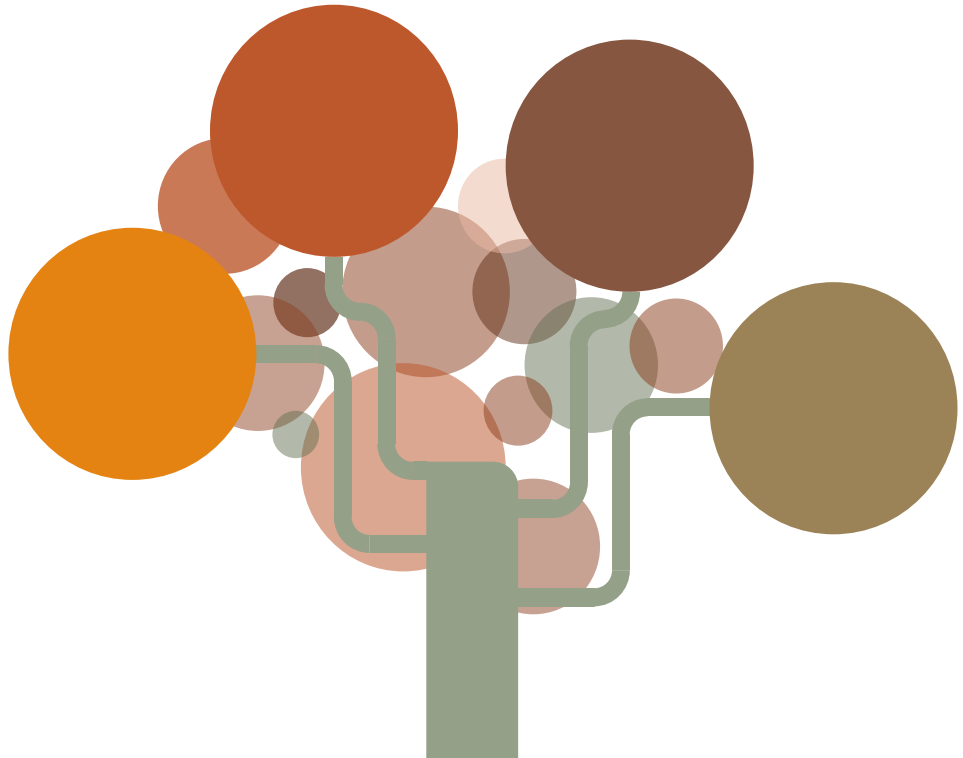
选择助教结果：助教将学生该课程信息链表节点中助教改为自己，同时int message变为1，学生每次登录后，判断int message 是否为1，为1则通过学生个人课表判断是否，来直接显示抽签结果，同时int message置为0。打开消息栏也可查看。


功能拓展

06 功能拓展

1、 输入密码错误5次则锁定用户，需找回密码（管理员/学生）
2、 修改密码（管理员/学生）
3、 密保功能（管理员/学生）
4、 找回密码（通过密保）（管理员/学生）
5、 抽签选课（由管理员启动，同时关闭其选课通道，
最后将选课结果返回给选课学生，由消息栏提示。
增加选课方式：抽签/非抽签，是否可选）（学生）
6、 同时选择多门课程（学生）
7、 查看某一门课程特定助教下的学生名单（管理员）
8、 选择助教需经过助教本人同意，并返回将信息给学生本人，
由消息栏提示（学生）
9、 增加功能——消息栏（登陆后若有消息显示提醒，没有消息
则不提示）（学生）
10、基本每一处都能返回上一级
11、退出系统
12、撤销助教报名（该助教下的学生登录后显示提示）
13、更换助教和取销助教
14、课程编号和课程名称都实现操作
15、密文登录
16、同时删除多门课程
17、同时添加多门课程

标红的功能已实现！




Thanks
project1
End