中 北 大 学

数据结构课程设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| **学 院、系：** | 软件学院 | | |
| **专 业：** | 软件工程 | | |
| **班 级：** | 181300414 | | |
| **学 生 姓 名：** | 魏楚镛 | **学 号：** | 18130041423 |
| **设 计 题 目：** | 活期储蓄账目管理系统 | | |
|  |  | | |
| **起 迄 日 期:** | 2019年6月14日**~** 2019年6月28日 | | |
| **指 导 教 师:** | 杨秋翔 | | |

日期:2019年6月28日

**课 程 设 计 说 明 书**

|  |  |
| --- | --- |
| **1．设计目的：** | |
| 《数据结构》课程主要介绍最常用的数据结构，阐明各种数据结构内在的逻辑关系，讨论其在计算机中的存储表示，以及在其上进行各种运算时的实现算法，并对算法的效率进行简单的分析和讨论。进行数据结构课程设计要达到以下目的：   * 了解并掌握数据结构与算法的设计方法，具备初步的独立分析和设计能力； * 初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能； * 提高综合运用所学的理论知识和方法独立分析和解决问题的能力； * 训练用系统的观点和软件开发一般规范进行软件开发，培养软件工作者所应具备的科学的工作方法和作风。 | |
| **2 任务概述** | |
| 建立一个账目系统，通过在文件中读取数据去实现用户的开户，销户，存款，取款，以及查询，在总体中要能排列显示所有用户的记录。  在设计时要考虑用户的使用是否方便，查找记录是否舒适，要尽量做到高效率，并且要保护文件不损坏，不出现数据乱码。 | |
| **3 本设计采用的数据结构** | |
| 记录用户的信息：  typedef struct user{  int ID;//用户的账号  char name[21];//用户的名字  double money;//用户的钱  char password[7];//密码  }user;   1. 用户的名字和密码是因为要检测长度是否超出，故多加一位。 2. ID用int类型而不用char类型是因为要按顺序记录账号   数组：动态分配结构体user数组 | 记录所有用户的使用记录：  typedef struct //记录  {  int hour, mid, sec;//时间  int year, month, days;//日期  int ID;//用户账号  int fun;//功能  double wec;//交易额  double money;//余额  }score;  1.用数字为标记来记录业务：  “0”开户  “1”存款  “2”取款  “3“销户  使得存储减少读取速度加快 |
| **4 系统功能模块结构图及主要函数说明** | |
| **财务报销流程**开户户模块主要是输入需要开户的姓名，然后建立用户账号，输入账号密码。  从文档中取出数据后：  查询用户是根据提醒输入需要登录的账号，然后再输入其密码，之后就可以查看其账号  上的相关资料。  存款、取款以及查询余额的操作主要是输入用户账号，输入密码然后对该账号进行相关  操作的。  销户模块则是输入需要销户的账号，然后更改该用户ID为-1，即为销户。  最后将用户数据和记录存回相关文档。 | |
| add()函数：用户开户。建立用户的名字，存款金额，密码。名字如果多于20位将  会提示重新输入，密码只能用数字且6位否则将会有相应的提示。用户的ID使用  cout()函数查找文件中用户数+1赋予。然后记录文件。  Del()函数：用户销户。先创建用户数组(struct temp[])，然后读取用户文件记录，  根据ID查找用户，找到并确认密码（pass）后，将该ID更改为-1。然后记录文件。  save()函数：用户存款。先创建用户数组(struct temp[])，然后读取用户文件记录，  根据ID查找用户，找到后，更改该用户的余额（money）。然后记录文件。  Get()函数：用户取款。先创建用户数组(struct temp[])，然后读取用户文件记录，  根据ID查找用户，找到并确认密码后，更改该用户的余额（money）。然后记录文件。  Select()函数:查询用户余额。先创建用户数组(struct temp[])，然后读取用户文件  记录，根据ID查找用户，找到并确认密码后，显示用户的信息。  display()函数：显示用户资料。用while循环使用struct temp结构来读取用户文件，显示，直到读完。  score2函数：构建记录结构体。将读取的用户结构体的有关信息赋给  struct user temp2中。 | |
| **5 程序运行数据及其结果** | |
| **图5.1 系统界面**    **图5.2 用户开户**  **图5.3 用户取款**    **图5.4 用户取款**  **图5.5 用户资料显示** | |
| **图5.5 用户余额查询**  **图5.7 用户交易记录**  **图5.8 修改密码** | |
| **图5.9 用户销户（未取完）**  **图5.10 用户销户（成功）** | |
| **6 课程设计心得** | |
| 经过这几天的设计，总的来说感觉非常的有意义，刚刚看实验报告时认为这是件非常  难的事，但是最后还是做出来了，这其中还是有很多的困难的，其中要求使用文件时一  度让我碰壁，在这方面的基础并不是非常好老是在运行时把我辛辛苦苦的记录的文件删  除的一干二净。  不过也是因为这样才让我了解到编程的经验，我也觉得非常有意思，在这其中遇到  了很多奇怪的事，然后一一解决，当我改完最后一个代码时，找下一个buge时，发现  没有了，有一种莫名的的伤感，然后便是一种不可思议的的惊奇！我玩成了。  然后自己在改的过程中加了一些东西，比如时间，页面的排版，因为是金融储蓄，所  以，我很多的是在考虑用户的使用，并尽力的指导用户使用的更好，在操作上更完善，  查看记录时更轻松，舒适。 | |
| #include<stdio.h>  #include<math.h>  #include<conio.h>  #include<string.h>  #include<stdlib.h>  #include<time.h>  #define file "erhuo.txt"  #define file2 "JiLu.txt"  typedef int Status;  /\*（1）实现储户开户。开户时需要填写登记卡，记录储户信息（包括姓名和账号），并设置开户余额为0。  （2）实现储户销户。需要先找到储户的账户信息，然后将其删除。  （3）向某账户存款。需要找到账户信息，然后修改账户的余额（原来的余额?+?存入金额）。  （4）从某账户取款。需要找到账户信息，然后修改账户的余额（原来的余额???取出金额）。  （5）排序显示所有账户信息。根据账号对所有账户信息排序。  （6）查询某账户余额。找到要查找的账户，然后显示账户余额。  （7）查询某账户交易记录。找到要查询的账户，然后显示该账户的交易记录。  （8）所有账户及其交易信息存储至文件，程序运行时从文件中读入。\*/  typedef struct user {//用户  int ID;  char name[21];  double money;  char password[7];  }user;  typedef struct //记录  {  int hour, mid, sec;//时间  int year, month, days;//日期  int ID;  int fun;  double wec;  double money;  }score;  void print()  {  printf("请选择所需业务\n");  printf("%8s%34s", "（1）开户", "（5）查询账户余额\n\n");  printf("%8s%42s", "（2）销户", "（6）排序显示所有账户信息\n\n");  printf("%8s%34s", "（3）存款", "（7）查询交易记录\n\n");  printf("%8s%30s", "（4）取款", "（8）修改密码\n\n");  printf("%8s","（9）退出系统\n\n");  printf("输入数字（1~8）");  }  void local(score &temp2)  {  struct tm \*d;  time\_t t;  time(&t);  d = localtime(&t);  temp2.days = d->tm\_mday;  temp2.hour = d->tm\_hour;  temp2.mid = d->tm\_min;  temp2.sec = d->tm\_sec;  temp2.year = d->tm\_year + 1900;  temp2.month = d->tm\_mon + 1;  return;  }  score score2(user temp, int n, double wec)  {//记录文件  score temp2;  local(temp2);  temp2.fun = n;  temp2.ID = temp.ID;  temp2.money = temp.money;  temp2.wec = wec;  return temp2;  }  void add()  {//开户  Status couts();  int mun, i;  FILE \*fd;  struct user temp;  score temp2;  //找序号  if ((fd = fopen(file, "rb")) == NULL)  {  printf("cannot open file!");  exit(1);  }  mun = 0;  while (fread(&temp, sizeof(struct user), 1, fd))  {  mun++;  }  temp.ID = mun + 1;  fclose(fd);  //建立名字  printf("%30s\n", "\*\*\*plase add user\*\*\*");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("Please enter name: ");  do {  scanf("%s", temp.name);  fflush(stdin);  if (strlen(temp.name) <= 20)  break;  else  printf("Don't char length more than 20!");  } while (1);  //密码设置  int mark = 0;  printf("Please enter password:");  while (1)  {  gets(temp.password);  fflush(stdin);  for (i = 0; i<7;)  {  if (temp.password[i] <'0' || temp.password[i]>'9')  {  break;  }  else i++;  }  if (i != 6 || strlen(temp.password) != 6)  {  if (i != 6)  printf("password must number!");  if (strlen(temp.password) != 6)  printf("password must 6!");  }  else break;  }  //存储金额  printf("输入存款金额：");  scanf("%lf", &temp.money);  //存入文件  if ((fd = fopen(file, "ab")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  else  {  fwrite(&temp, sizeof(user), 1, fd);  printf("Creat user succeal!");  }  fclose(fd);  //记录存入文件  temp2 = score2(temp, 0, temp.money);  if ((fd = fopen("JiLu.txt", "ab")) == NULL)  {  printf("cannot open file2!\n");  exit(1);  }  else {  fwrite(&temp2, sizeof(score), 1, fd);  }  fclose(fd);  }  int cout() //函数目的：计算文件中的记录总数  {  FILE \*fp;  int num;  struct user temp;  num = 0;  if ((fp = fopen(file, "rb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  while (fread(&temp, sizeof(struct user), 1, fp))  {  num++;  }  fclose(fp);  return num;  }  void select()//按账号查询账户余额  {  int i, maxmun;  char pass[6];  user temp;  FILE \*fd;  printf("%30s\n", "\*\*\*按照账号查询账户余额\*\*\*");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("输入要查询的账户");  scanf("%d", &i);  printf("输入密码");  scanf("%s", &pass);  fflush(stdin);  if ((fd = fopen(file, "rb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  maxmun = cout();  if (i>maxmun)  {  printf("账户不存在！");  return;  }  do {  fread(&temp, sizeof(user), 1, fd);  } while (temp.ID != i);  if (strcmp(temp.password, pass) != 0)  printf("密码错误!");  else {  printf("%8s %15s %6s %10s\n", "账号", "用户姓名", "密码", "存款余额");  printf("---------------------------------------------\n\n");  printf("NO.%d %15s %6s %.2f\n", temp.ID, temp.name, temp.password, temp.money);  }  fclose(fd);  }  Status password\_confrimation(char password[]) {  int x = 5;  char pass[6];  printf("请输入账户密码:");  scanf("%s", &pass);  fflush(stdin);  do  {  x -= 1;  if (strcmp(pass, password) != 0)  {  printf("密码错误，还有%d次机会\n", x);  scanf("%s", &pass);  fflush(stdin);  }  else return 1;  } while (x>1);  printf("退出");  return 0;  }  void save()  {  int cout();  printf("%30s\n", "\*\*\*用户存款业务\*\*\*");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  int i, m, num;  double money;  FILE \*fd;  user \*temp, use;  num = cout();  temp = (user\*)malloc(num \* sizeof(user));  char pass[6];  printf("输入存储账户：");  scanf("%d", &i);  if (i>num || i<0)  {  printf("该账户不存在！\n");  return;  }  //输入密码  printf("输入账户密码：");  scanf("%s", &pass);  fflush(stdin);  //记录账户  if ((fd = fopen("erhuo.txt", "rb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  for (m = 0; m<num; m++)  {  fread(&temp[m], sizeof(user), 1, fd);  }  fclose(fd);  //存款  printf("输入存款金额：");  do {  scanf("%lf", &money);  if (money<0)  {  printf("存款金额无效！\n");  }  } while (money<0);  //将数据存回文件中  if ((fd = fopen("erhuo.txt", "wb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  for (m = 0; m<num; m++)  {  if (temp[m].ID == i)  {  temp[m].money += money;  use = temp[m];  }  fwrite(&temp[m], sizeof(user), 1, fd);  }  fclose(fd);  free(temp);  printf("存款成功！");  //记录数据  score temp2;  if ((fd = fopen(file2, "ab")) == NULL)//以后追方式打开文件  {  printf("文件不存在！");  exit(1);  }  temp2 = score2(use, 1, money);  fwrite(&temp2, sizeof(score), 1, fd);  fclose(fd);  }  void get()  {//取款  printf("%30s\n", "\*\*\*用户取款业务\*\*\*");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  int i, m, num, x;  double money;  FILE \*fd;  user \*temp, use;  printf("输入取款账户:");  scanf("%d", &i);  num = cout();  if (i>num)  {  printf("账户不存在！");  return;  }  //记录账户  temp = (user\*)malloc(num \* sizeof(user));  if ((fd = fopen(file, "rb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  for (m = 0; m<num; m++)  {  fread(&temp[m], sizeof(user), 1, fd);  }  fclose(fd);  //cunkuan  if ((fd = fopen(file, "wb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  for (m = 0; m<num; m++)  {  if (temp[m].ID == i)  {  while (1)//假循环  {  x = password\_confrimation(temp[m].password);//确认密码  if (x == 0) break;  printf("输入取款金额:");  do {  scanf("%lf", &money);  if (money<0)  {  printf("取款金额无效！\n");  }  if (money>temp[m].money)  printf("取款金额多于余额！\n");  } while (money<0 && money<temp[m].money);  temp[m].money -= money;  use = temp[m];  printf("取款成功！");  break;  }  }  fwrite(&temp[m], sizeof(user), 1, fd);  }  fclose(fd);  free(temp);  //记录文件  score temp2;  if ((fd = fopen(file2, "ab")) == NULL)  {  printf("文件不存在！");  exit(1);  }  temp2 = score2(use, 2, money);  fwrite(&temp2, sizeof(score), 1, fd);  fclose(fd);  }  void display() //显示用户资料  {  printf("%30s\n", "\*\*\*用户资料显示\*\*");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  FILE \* fp;  struct user temp;  if ((fp = fopen(file, "rb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  printf("%5s %14s %14s %10s\n", "账号", "用户姓名", "密码", "存款余额");  printf("---------------------------------------------\n\n");  while (fread(&temp, sizeof(struct user), 1, fp))  {  if(temp.ID==-1)  continue;  printf("NO.%d %20s %10s %.2f\n", temp.ID, temp.name, temp.password, temp.money);  }  fclose(fp);  }  void Del()  {//销户  int count();  printf("%30s\n", "\*\*\*注销账户\*\*\*");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  FILE \* fd;  user \*temp, use;  int i, maxnum, m;  char f;  printf("输入注销账户：");  scanf("%d", &i);  maxnum = cout();  if (i>maxnum || i<0)  {  printf("该账户不存在！");  return;  }  //记录账户  temp = (user\*)malloc(maxnum \* sizeof(user));  if ((fd = fopen(file, "rb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  for (m = 0; m<maxnum; m++)  {  fread(&temp[m], sizeof(user), 1, fd);  }  fclose(fd);  //shanchu  //将数据存回文件中  if ((fd = fopen(file, "wb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  for (m = 0; m<maxnum; m++)  {  if (temp[m].ID == i)  {  if (temp[m].money != 0.00)  {  printf("请取出所有存款。");  fwrite(&temp[m], sizeof(user), 1, fd);  continue;  }  else //余额为0  {  int x;  if ((x = password\_confrimation(temp[m].password)) == 0)  return;  else  {  printf("请再次确认是否销户?(Y or N?)");  scanf("%c", &f);  while (f != 'Y' && f != 'y' && f != 'n' && f != 'N')  {  printf("输入错误，重新输入");  scanf("%c", &f);  }  if (f == 'Y' || f == 'y')  {  use = temp[m];  temp[m].ID = -1;  printf("销户成功！\n");  printf("NO.%d %15s %15s %.2f\n", use.ID, use.name, use.password, use.money);  //break;  }  }  }  }  fwrite(&temp[m], sizeof(user), 1, fd);  }  fclose(fd);  free(temp);  //记录  score temp2;  score2(use, 1, 0);  if ((fd = fopen(file2, "ab")) == NULL)  {  printf("文件不存在！");  exit(1);  }  temp2 = score2(use, 3, 0);  fwrite(&temp2, sizeof(score), 1, fd);  fclose(fd);  }  void Tran\_record()  {//交易记录  printf("%30s", "\*\*\*交易记录\*\*\*\n");  FILE \*fd;  user temp;  score temp2;  int i, maxnum, x;  printf("输入要查询账户：");  scanf("%d", &i);  maxnum = cout();  if (i>maxnum)  {  printf("该账户不存在！\n");  return;  }  if ((fd = fopen(file, "rb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  while (fread(&temp, sizeof(struct user), 1, fd))  {  if (temp.ID == i)  break;  }  fclose(fd);  if (temp.ID != i)  {  printf("用户已被注销。");  return;  }  //确认密码  if ((x = password\_confrimation(temp.password)) == 0)  return;  //显示记录  if ((fd = fopen(file2, "rb")) == NULL)  {  printf("文件不存在！");  exit(1);  }  printf("%12s %18s %12s %10s %6s %10s\n", "时间", "账号", "用户姓名", "业务", "交易", "余额");  printf("--------------------------------------------------------------------\n\n");  while (fread(&temp2, sizeof(score), 1, fd))  {  if (temp2.ID == i)  {  printf("%2d年%2d月%2d日\t%2d:%2d:%2d", temp2.year, temp2.month, temp2.days, temp2.hour, temp2.mid, temp2.sec);  switch (temp2.fun)  {  case 0:printf("%cNO.%5d %15s %6s %10.2f %10.2f\n", ' ', temp2.ID, temp.name, "开户", temp2.wec, temp2.money); break;  case 1:printf("%cNO.%5d %15s %6s %10.2f %10.2f\n", ' ', temp2.ID, temp.name, "存款", temp2.wec, temp2.money); break;  case 2:printf("%cNO.%5d %15s %6s %10.2f %10.2f\n", ' ', temp2.ID, temp.name, "取款", temp2.wec, temp2.money); break;  case 3:printf("%cNO.%5d %15s %6s %10.2f %10.2f\n", ' ', temp2.ID, temp.name, "销户", temp2.wec, temp2.money); break;  }  }  }  return;  }  //改密码  void pass\_exc()  {  FILE \*fd;  char pass[6];  int n, i, m, maxnum;  user \*temp;  printf("请输入要修改密码的账户：");  scanf("%d",&n);  maxnum = cout();  if (n>maxnum)  {  printf("该账户不存在！\n");  return;  }    temp = (user\*)malloc(maxnum \* sizeof(user));  if ((fd = fopen(file, "rb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  for (m = 0; m<maxnum; m++)  {  fread(&temp[m], sizeof(user), 1, fd);    if (temp[m].ID == n)  {  printf("Please enter password:");  scanf("%s",pass);  if (strcmp(pass, temp[m].password) != 0)  {  printf("密码错误！\n\t退出");  return;  }  }  }  fclose(fd);    if ((fd = fopen(file, "wb")) == NULL)  {  printf("cannot open file!\n");  exit(1);  }  for (m = 0; m<maxnum; m++)  {  if (temp[m].ID == n)  {    printf("请修改密码（退出业务输入 N）：");  while (1)  {scanf("%s",pass);  fflush(stdin);  for (i = 0; i<7;)  {  if (pass[i] <'0' || pass[i]>'9')  {  break;  }  else i++;  }  if (i != 6 || strlen(temp[m].password) != 6)  {  if (i != 6)  printf("password must number!");  if (strlen(temp[m].password) != 6)  printf("password must 6!");  }  else break;  }  strcpy(temp[m].password, pass);  printf("修改成功！");  }  fwrite(&temp[m], sizeof(score), 1, fd);  }  fclose(fd);  free(temp);  }  int main()  {  int x;  while (1)  {  printf("%40s", "Current savings account management\n");  printf("---------------------------------------------\n\n");  print();  scanf("%d", &x);  system("cls");  switch (x)  {  case 1:add(); break;  case 2:Del(); break;  case 3:save(); break;  case 4:get(); break;  case 5:select(); break;  case 6:display(); break;  case 7:Tran\_record(); break;  case 8:pass\_exc();break;  case 9:printf("\n\n欢迎下次光临！\n\n\t祝你事业有成！");return 0;  default:  printf("对不起，你所选择的功能不存在!\n");  }  getch();  system("cls");  }  } | |
|  | |