# 田径运动会信息管理系统

组员：吴惠英 、李苗苗、钟婷

***目录***

**一：需求分析**

**1.1：可行性分析**

**1.2：开发环境和开发工具**

**1.2.1C/C++语言简介**

**1.2.2开发背景**

**1.2.3开发环境**

**1.3：系统需求分析**

**1.4：系统总体设计**

**1.4.1：系统流程图**

**1.5：数据流图**

**二：概要设计**

**2.1数据结构**

**2.2数据库设计**

**2.3程序模块**

**三：详细设计**

**四：程序代码**

**五：结果与测试**

**5.1实验结果**

**5.2测试**

**5.2.1数据的录入测试**

**5.2.2数据的删除测试**

**一、需求分析**

1．1可行性分析

在当今社会上，体育运动越来越普及，参与运动会的人越来越多，素质越来越高。传统的运动会，一般是以人手组织为主，运动员的报名、赛事的安排、比赛场地的安排和成绩的记录等操作全都由人工操作，参加人数越多，运动会的规模就越大，管理工作就更加费时费力，而且容易出错。学校田径运动会管理系统为组织者提供了一个简捷方便的方式来组织管理运动会，学校田径运动会管理系统是一种“人机交互系统”，通过简单的键盘输入来完成烦琐的记录与安排工作，保证运动会流畅的进行。学校田径运动会管理系统是典型的数据库应用系统，可以完成田径运动会管理中的运动员基本信息录入、竞赛成绩处理及公告、成绩统计、信息查询等事务处理功能。根据学校田径运动会的一般进程，本系统分为数据修改、查询、删除、计算等四个工具模块来完成所需要的功能，依据关系型数据库理论设计数据库的结构。使用该系统可以极大地减小工作人员的劳动量，同时还可以大大提高工作效率、可靠性和正确性。

1．2开发环境和开发工具

**1.2.1C/C++语言简介**

C语言是一种计算机程序设计语言。它既具有高级语言的特点，又具有汇编语言的特点。它由美国贝尔研究所的D.M.RitcIue于1972年推出。1978后，C语言已先后被移植到大、中、小及微型机上。它可以作为工作系统设计语言，编写系统应用程序，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。它的应用范围广泛，具备很强的数据处理能力，不仅仅是在软件开发上，而且各类科研都需要用到C语言，适于编写系统软件，三维，二维图形和动画。

**1.2.2开发背景**

随着科学技术的不断发展，计算机科学日渐成熟，其强大的功能已为人们深刻认识，它已进入人类社会的各个领域并发挥着越来越重要的作用。采用计算机进行信息化管理已经成为衡量企业管理科学化和现代化的重要标志，而人事管理的全面自动化、信息化则是其中重要的组成部分。人事管理的好坏对于企业的决策者和管理者来说都至关重要，在很大程度上影响着企业的经济效益和社会效益。因此，本次实验开发的人事信息管理系统具有一定的使用价值和现实意义。

**1.2.3开发环境**

本文所采用的开发环境主要是基于C++语言在VisuaIC++平台上开发的系统，界面友好，功能强大，调试也很方便。这是微软出的一个C语言集成开发环境（IDE），主要有：VC++6.O等，分为企业版和学生版等，对于初学者来说VC++6.0是比较容易上手的。

**1.3系统需求分析**

本实验中，对不同的对象采取不同的存储结构，对比赛项目采取链式结构，而对于参赛运动员采取一个结构就可以了。本程序中主要用到线性表的一些基本操作，将比赛项目信息，运动员信息，赛事信息都构造成线性表。具体功能通过主函数分别调用各函数实现。

参考本实验的功能要求，本程序分为七个模块：主函数、信息录入函数、信息修改函数、信息删除函数、信息查询函数，信息输出函数。

信息录入函数有包括比赛项目信息录入函数（包括比赛编号、名称等），运动员信息录入函数（运动员的编号、姓名、性别、年龄、工作单位或省份）、比赛赛事信息录入函数（某运动员参加的比赛项目、成绩、名次）。

信息修改函数包括比赛项目、运动员信息、赛事信息修改。

信息查询函数包括根据比赛项目、运动员编号进行查询。

信息输出函数包括比赛项目、运动员信息、赛事信息输出等。

1.4系统总体设计

支持输入各个项目的基本信息

可以按比赛项目编号或名称查询、删除、修改信息。

可以按运动员编号或姓名查询、删除、修改信息

数据输入信息并能随时查询

**1.4.1系统流程图**

运动员信息

运动信息表

运动项目安排系统

事务

运动项目信息

**1.5数据流图**

比赛项目

报名单

项目

1.4

决定比赛名单

1.3

登记队员

1.2

运动员编号

1.1

检查报名单

运动员编号

正式报名单

合格报名单

运动员号码单

不合格报名单

运动员名单

项目成绩

2.2

记录成绩

单项成绩

2.1

检查破纪录

单项成绩

运动员名单

最高纪录

运动员名单

2.3

累计团队总分

2.5

汇报各队成绩

各队成绩

团队成绩

团队名次

2.4

决定团队名次

图1-1 运动会管理系统

项目成绩

运动员号码单

团体成绩

单项成绩

报名单

各队成绩

登记报名单

统计成绩

运动员名单

团体名次

比赛项目

项目参加者名单

图1-2 信息录入

**二、概要设计**

2.1数据结构

从整个管理系统来看，我们知道这里主要有三个主体，一个是比赛项目信息，一个是运动员，另一个是比赛赛事项目信息。这三者不单单是作为简单的主体存在，他们都有很多功能，所以我们使用类和链表来定义。如下：

比赛项目的定义：

classMatch

{

public:

Match \*next；//为创建链表做准备

int number; //比赛项目编号

char call[10]；//比赛项目名称

char time[10]；//比赛时间

char place[20]; //比赛地点

friend void input1();

};

运动员的定义：

Class Athlete

{

public:

AthIete \*next;

int number; //运动员的编号

char name[10]；//运动员的姓名

char part[20]；//运动员所属工作单位或省份

char sex[10]; //运动员性别

int age; //运动员年龄

friendvoidinput2();

}

比赛赛事项目的定义：

Class Message

{

public:

Message \*next;

char name[10]；//运动员姓名

char avent[0]；//运动员参加的某比赛项目名称

int score；//成绩

int ca; //名次

friend void input3();

}

2.2数据库设计

比赛项目表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 说明 |
| 项目编号 | int | 10 | 主键not null |
| 项目名称 | char | 10 | not null |
| 比赛时间 | char | 10 | not null |
| 比赛地点 | char | 20 | not null |

运动员信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 说明 |
| 运动员编号 | int | 10 | 主键 |
| 运动员姓名 | char | 10 | not null |
| 运动员工作单位 | char | 20 | not null |
| 运动员性别 | char | 10 | not null |
| 运动员年龄 | int | 10 | not null |

比赛赛事项目表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 说明 |
| 运动员姓名 | char | 10 | 外键not null |
| 比赛项目名称 | char | 10 | not null |
| 成绩 | int | 10 | not null |
| 名次 | int | 10 | not null |

2.3程序模块

田径运动会信息管理系统

数据的显示

数据的查询

数据的修改

数据的插入

退出系统

数据的删除

运动信息输入

比赛项目信息修改

运动信息修改

比赛赛事信息输入

比赛项目信息输入

比赛赛事信息修改

比赛项目信息查询

运动信息查询

比赛赛事信息查询

比赛项目信息显示

运动信息显示

比赛赛事信息显示

运动信息输入删除

比赛项目信息删除

比赛赛事信息删除

通过对某高校运动会管理信息系统的基本功能调研和分析，各个子系统的主要功能

如下：

比赛项目信息管理：主要负责比赛项目信息的录入、修改、查询，显示比赛项目信息；

运动员注册：负责参赛运动员的注册登记，并可随时查询、打印参赛运动员信息：

成绩统计：用于录入、统计各项比赛的的预、决赛成绩，统计各运动员积分。本模块可以查询每个参赛运动员的成绩信息，打印各种成绩报表。

**三、详细设计**

**主菜单流程图：**

开始

显示主菜单

输入选择

N

是否合理

提示错误信息

Y

N

是否为1

N

N

是否为3

是否为2

是否为4

Y

Y

N

Y

Y

调用修改函数

调用删除函数

调用查询函数

调用录入函数

N

是否为5

是否为6

Y

Y

退出系统

调用显示函数

**数据的录入流程图：**

开始

主菜单

数据录入

Y

N

是否为1

N

是否为2

是否为3

Y

Y

Y

请输入比赛赛事信息

请输入运动员信息

请输入比赛项目信息

是否继续输入

Y

N

**数据的查询流程图：**

开始

主菜单

数据查询

Y

是否为1

N

是否为3

是否为2

N

Y

Y

请输入对应信息的姓名和比赛的项目

请输入对应信息的编号

请输入对应信息的编号

Y

Y

是否为1

N

是否为1

是否为2

是否为2

Y

Y

Y

Y

请输入远动员的名称

请输入运动员的编号

请输入比赛项目的名称

请输入比赛项目的编号

四、程序代码

比赛项目信息录入：

void input1()

{

Ofstream fout("e:\\比赛项目.dat",ios::app);

char ch;

Match a;

do

{

cout<<”请分别输入比赛项目编号、名称、比赛时间、比赛地点."<<endl;

cin>>a.number>>a.call>>a.place>>a.time;

a.next=NULL;

fout.write((char\*)&a,sizeof(Match));

cout<<"是否继续输入，如果继续键入Yly<<endl;

cin>>ch;

}while(ch=='y’llch=='Y');

fout.close();

}

运动员信息修改：

int change2() //对运动员相关信息修改

cout<<"您正在进行修改运动员基本信息操作！"<<endl；

creat2();

Athlete \*p1;

P1=head4;

cout<<"请输入要修改运动员的编号："<<endl；

int number1; //要修改的运动员编号

int sign=0; //设置的标记变量

cin>>number1；

while(p1!=NULL)

{

if(p1->number==number1) {sign=1,break;}

p1=->next;

if(sign==0)

cout<<"没有找到要修改的记录！"<<endl，

else

{

cout<<"请选择要修改运动员的哪些信息"<<endl；

cout<<"修改运动员的编号一一1"<<endl;

cout<<"修改运动员的姓名一一2"<<endl；

cout<<"修改运动员所属工作单位或省份一一3"<“endl；

cout<<"修改运动员性别一一4"<<endl;

cout<<"修改运动员年龄一一4"<<endl;

cout<<"请输入对应信息的编号"<<endl；

int a;

int nu; //新的整形数据

char info[20]；//新的字符数组

cin>>a;

switch(a)

{

case1：cout<<"请输入新的运动员编号:"<<endl;

cin>>nu;

pl->number=nu;

cout<<"运动员的编号修改成功！"<<endl;

break;

case2:cout<<"请输入新的运动员姓名:"<<endl;

cin>>info;

strcpy(pl->name,info);

“ut<<”运动员姓名修改成功！"<<endl;

break，

case3：cout<<"请输入新的运动员所属工作单位或省份："<“endl；

cin>>info;

strcpy(p1->part,info);

cout<<"运动员所属工作单位或省份修改成功！"<<endl;

break;

case4：cout<<"请输入新的性别:"<<endl;

cin>>info;

Strcpy(p1->sex,info);

cout<<”运动员性别修改成功！"<<endl；

break;

case5:cout<<"请输入新的年龄:"<<endl;

cin>>nu;

p1->age=nu;

cout<<”运动员年龄修改成功！"<<endl；

}

ofstream fout("e:\\动员.dat"ios::out);

pl->head4;

while(p1!—NULL)

{

fout.write((char\*)p1,sizeof(Athlete));

p1=p1->next;

fout.close();

return 0；

比赛赛事信息显示：

voidprint3()

{

creat3();

Message \*p1=head7;

cout<<"请分别输出比赛参赛运动员姓名、比赛项目名称、比赛成绩、比赛名次"<<endl;

while(p1!=NULL)

{

cout<<p1->name<<’t'<<p1->avent<<’\t’<<p1->score<<’\t’<<pl->ca<<endl;

p1=p1->next;

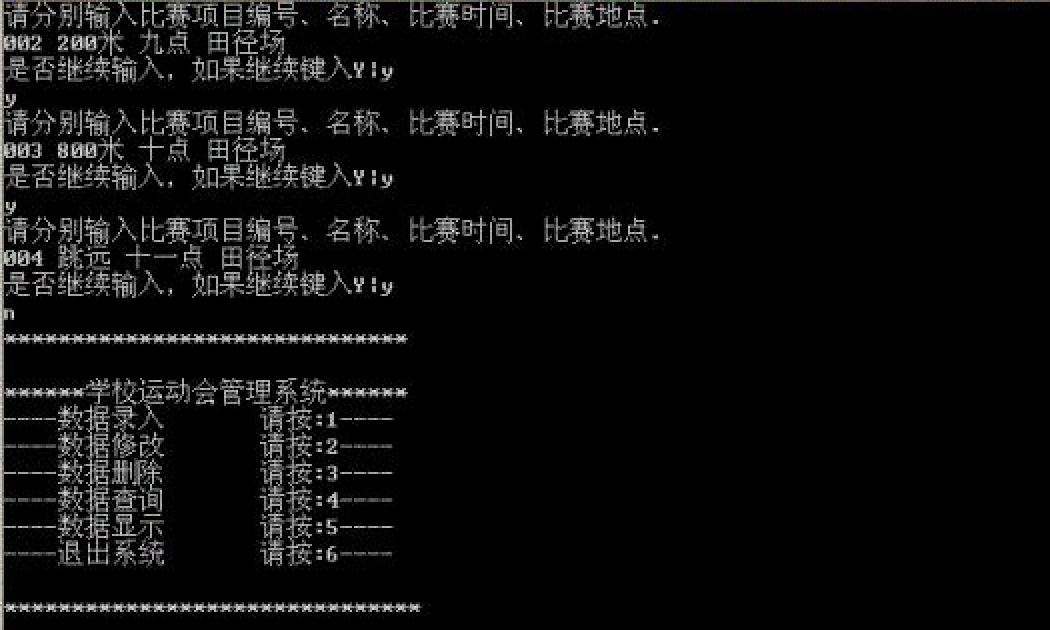
}

}

五、系统测试

5.1实验结果

信息录入：



信息修改：



信息删除：



信息查询：



信息输出及退出系统：



5.2测试

**5.2.1数据的录入测试**

表5-1 比赛项目的录入名称测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | 预期的输出 | 实际的输出 | 评价 |
| 比赛项目的名称 | 成功，不会出现信息的错误 | 显示结果是正确的 | 该功能可行 |
| 比赛项目的名称 | 不成功，会出现信息的错误 | 显示结果是错误的 | 该功能不可行 |

表5-2 比赛项目的录入时输入时间测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | 预期的输出 | 实际的输出 | 评价 |
| 比赛时间 | 成功，不会出现信息的错误 | 显示结果是正确的 | 该功能可行 |
| 比赛时间 | 不成功，会出现信息的错误 | 显示结果是错误的 | 该功能不可行 |

表5-3 比赛项目的录入时输入地点测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | 预期的输出 | 实际的输出 | 评价 |
| 比赛地点 | 成功，不会出现信息的错误 | 显示结果是正确的 | 该功能可行 |
| 比赛地点 | 不成功，会出现信息的错误 | 显示结果是错误的 | 该功能不可行 |

表5-5 比赛项目的录入时输入名称测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | 预期的输出 | 实际的输出 | 评价 |
| 运动员名字 | 成功，不会出现信息的错误 | 显示结果是正确的 | 该功能可行 |
| 运动员名字 | 不成功，会出现信息的错误 | 显示结果是错误的 | 该功能不可行 |

表5-6 比赛项目的录入时输入名称测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | 预期的输出 | 实际的输出 | 评价 |
| 运动员单位 | 成功，不会出现信息的错误 | 显示结果是正确的 | 该功能可行 |
| 运动员单位 | 不成功，会出现信息的错误 | 显示结果是错误的 | 该功能不可行 |

**5.2.1数据的删除测试**

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | 预期的输出 | 实际的输出 | 评价 |
| 要删除的运动员的编号 | 成功 | 显示结果是错误的 | 该功能不可行 |