**体操赛事信息管理系统**

**详细设计**

目录

[1.引言 3](#_Toc502905519)

[1.1编写目的 3](#_Toc502905520)

[1.2背景 3](#_Toc502905521)

[1.3定义 4](#_Toc502905522)

[1.4参考资料 4](#_Toc502905523)

[2系统架构 4](#_Toc502905524)

[2.1硬件架构 4](#_Toc502905525)

[2.2技术平台 5](#_Toc502905526)

[2.3软件架构 5](#_Toc502905527)

[3.数据结构设计 6](#_Toc502905528)

[3.1设计思路 6](#_Toc502905529)

[3.2具体类设计 6](#_Toc502905530)

[4.客户端的设计 13](#_Toc502905531)

[4.1设计思路 13](#_Toc502905532)

[4.2 public class client 13](#_Toc502905533)

[4.3 TCP的数据报格式设计 14](#_Toc502905534)

[4.4 数据库设计public class Dao 17](#_Toc502905535)

[4.4.1类结构 17](#_Toc502905536)

[4.4.2类成员设计 18](#_Toc502905537)

[4.4.3方法login设计 19](#_Toc502905538)

[4.4.4 getOrderInfo方法设计 19](#_Toc502905539)

[5.服务器的设计 20](#_Toc502905540)

[5.1设计思路 20](#_Toc502905541)

[5.2public class SocketHandle 20](#_Toc502905542)

[5.2.1类结构设计 20](#_Toc502905543)

[5.2.2类成员设计 21](#_Toc502905544)

[5.3public class Server 22](#_Toc502905545)

[6.数据库表的设计 22](#_Toc502905546)

# 1.引言

## 1.1编写目的

本详细设计说明书的目的，旨在规范软件开发设计，推动项目在具体编码阶段有序正常的进行，使相关人员遵守统一的规范。

通过需求分析阶段的用例图设计以及概要设计阶段的功能模块划分等要素，确定系统各部分所需求的软件结构、数据结构和详细的类设计，并确定系统的接口设计、人机交互界面设计等。从而完成对系统设计的详细设计，以期使软件编程人员对于所需求软件的系统架构、系统流程、数据结构、详细类设计等基础性目标有一致的认识，降低系统实现的风险，加快项目实施进度，做到系统设计的规范性和全面性，以利于系统的实现、测试、维护和下一步的版本升级。

## 1.2背景

针对本开发软件系统的情况，特做此说明：

1. 待开发的软件系统的名称：体操赛事信息管理系统 ；
2. 本项目的任务提出者：刘斌；
3. 开发者：黄志豪、黄威豪、高亨利、骆家毅、胡哲、杨蕾、袁伟、李瞄；
4. 用户：体操赛事举办方相关工作人员（裁判、比赛控制人员、后台管理员）、体操赛事参赛方（运动代表队）；
5. 实现该软件的计算中心或计算机网络：PC、wifi、、蜂窝数据；
6. 该软件系统同其他系统或其他机构的基本的相互来往关系：无。

## 1.3定义

系统/本系统：如未特殊说明，均指体操赛事信息管理系统。

## 1.4参考资料

1. 《面向对象与Java程序设计》【朱福喜】（清华大学出版社）；
2. 《数据结构教程》【李春葆】（清华大学出版社）；
3. 《软件工程导论》【张海藩】（清华大学出版社）；
4. 《数据库原理与技术》【易为民】（清华大学出版社）；
5. 《面向对象分析与设计》【麻志毅】（机械工业出版社）；
6. GB8567-88[计算机软件产品开发文件编制指南]；
7. 体操赛事信息管理系统需求规格说明书；
8. 体操赛事信息管理系统概要设计。

# 2系统架构

## 2.1硬件架构

硬件架构图如图2.1所示：

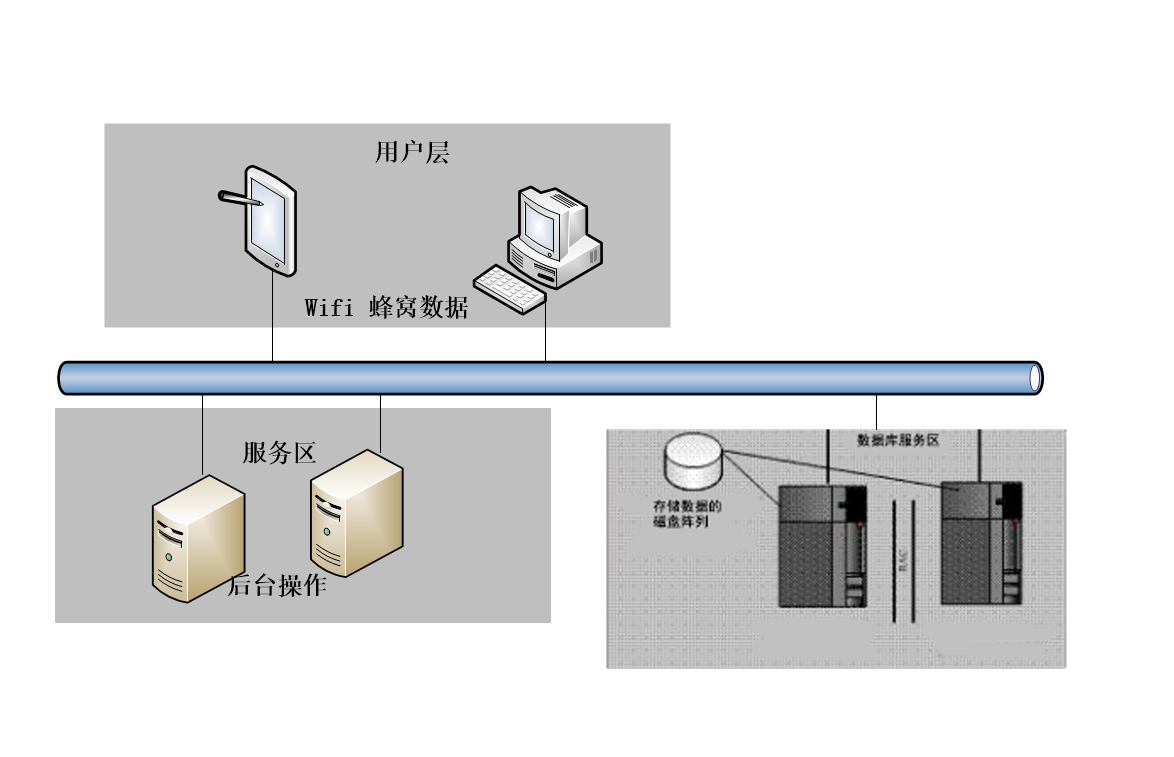


图2.1 硬件架构图

## 2.2技术平台

* 客户端：硬件配置：CPU 内存

软件配置：Andriod

* 服务器：硬件配置：内存500M以上 网卡：带宽100Mb/s以上

软件配置：Windows 7/8/10 Mysql Community Server JVM

* 数据库：Mysql
* 接口：Andriod标准接口、数据库与JAVA之间的JDBC连接接口；

## 2.3软件架构

软件架构图如图2.3所示：

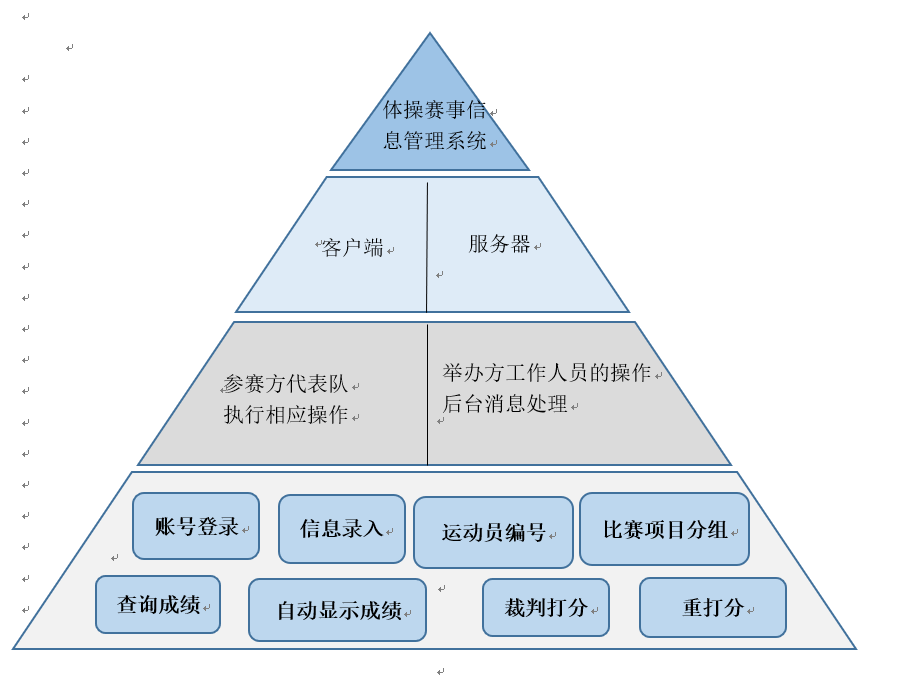


图2.3 软件架构图

# 3.数据结构设计

## 3.1设计思路

将各个具有不同属性的对象放在相应的类里面，进行属性的获取和设置。

## 3.2具体类设计

* public class Admin 系统管理员类

1. 属性说明

String adminUser;//账号

String adminPwd;//密码

1. 方法设计

public String getAdminUser()

public void setAdminUser(String adminUser)

public String getAdminPwd()

public void setAdminPwd(String adminPwd)

* public class Controller 比赛控制人员类

1. 属性说明

String ctrlUser;//账号

String ctrlPwd;//密码

1. 方法设计

public String getCtrlUser()

public void setCtrlUser(String ctrlUser)

public String getCtrlPwd()

public void setCtrlPwd(String ctrlPwd)

* public class Referee 裁判员类

1. 属性说明

String idNumber;//身份证号

String phone;//电话

String name;//姓名

String teamName;//队伍名

String refereeUser;//账号

String refereePwd;//密码

1. 方法设计

public String getIdNumber()

public void setIdNumber(String idNumber)

public String getPhone()

public void setPhone(String phone)

public String getName()

public void setName(String name)

public String getTeamName()

public void setTeamName(String teamName)

public String getRefereeUser()

public void setRefereeUser(String refereeUser)

public String getRefereePwd()

public void setRefereePwd(String refereePwd)

* public class Team 代表队类

1. 属性说明

String teamName;//队名

String teamUser;//队账号

String teamPwd;//密码

1. 方法设计

public String getTeamName()

public void setTeamName(String teamName)

public String getTeamUser()

public void setTeamUser(String teamUser)

public String getTeamPwd()

public void setTeamPwd(String teamPwd)

* public class Staff 队伍工作人员类

1. 属性说明

String idNumber;//身份证号

String phone;//电话

String name;//姓名

String teamName;//队名

char sex;//性别

String duty;//职务

1. 方法设计

public String getIdNumber()

public void setIdNumber(String idNumber)

public String getPhone()

public void setPhone(String phone)

public String getName()

public void setName(String name)

public String getTeamName()

public void setTeamName(String teamName)

public char getSex()

public void setSex(char sex)

public String getDuty()

public void setDuty(String duty)

* public class Athlete 运动员类

1. 属性说明

int athleteNo;//运动员编号

byte sex;//性别

String idNumber;//身份证号

String name;//姓名

AgeGroup ageGroup;//年龄范围

float academicRecord;//文化成绩

String teamName;//队名

1. 方法设计

public int getAthleteNo()

public void setAthleteNo(int athleteNo)

public byte getSex()

public void setSex(byte sex)

public void setSex(byte sex)

public String getIdNumber()

public void setIdNumber(String idNumber)

public String getName()

public void setName(String name)

public AgeGroup getAgeGroup()

public void setAgeGroup(AgeGroup ageGroup)

public float getAcademicRecord()

public void setAcademicRecord(float academicRecord)

public String getTeamName()

public void setTeamName(String teamName)

* public class Match 赛事表类

1. 属性说明

int eventNo;//项目编号

String matchType;//比赛类型

Date matchTime;//比赛时间

char preOrFinal;//预决赛

int groupNo;//组号

1. 方法设计

public int getEventNo()

public void setEventNo(int eventNo)

public String getMatchType()

public void setMatchType(String matchType)

public Date getMatchTime()

public void setMatchTime(Date matchTime)

public char getPreOrFinal()

public void setPreOrFinal(char preOrFinal)

public int getGroupNo()

public void setGroupNo(int groupNo)

* public class OrderInfo 出场顺序类

1. 属性说明

int eventNo;//项目编号

AgeGroup ageGroup;//年龄组

String preOrFinal;//预决赛

int groupNo;//组号

int athleteNo;//运动员编号

int order;//出场编号

1. 方法设计

public int getEventNo()

public void setEventNo(int eventNo)

public AgeGroup getAgeGroup()

public void setAgeGroup(AgeGroup ageGroup)

public String getPreOrFinal()

public void setPreOrFinal(String preOrFinal)

public int getGroupNo()

public void setGroupNo(int groupNo)

public int getAthleteNo()

public void setAthleteNo(int athleteNo)

public int getOrder()

public void setOrder(int order)

* public class AgeGroup 年龄组类

1. 属性说明

int ageMin;//年龄上界

int ageMax;//年龄下界

1. 方法设计

public int getAgeMin()

public AgeGroup(int ageMin, int ageMax)

public void setAgeMin(int ageMin)

public int getAgeMax()

public void setAgeMax(int ageMax)

* public class Event 比赛项目类

1. 属性说明

int eventNo;//项目编号

String sex;//男女组

String eventName;//项目名

1. 方法设计

public int getEventNo()

public void setEventNo(int eventNo)

public String getSex()

public void setSex(String sex)

public String getEventName()

public void setEventName(String eventName)

* public class ScoreResult 赛事结果类

1. 属性说明

int athleteNo;//运动员编号

int eventNo;//项目编号

String refereeUser;//裁判账号

String scoreType;//打分类型

float score;//得分

1. 方法设计

public int getAthleteNo()

public void setAthleteNo(int athleteNo)

public int getEventNo()

public void setEventNo(int eventNo)

public String getRefereeUser()

public void setRefereeUser(String refereeUser)

public String getScoreType()

public void setScoreType(String scoreType)

public float getScore()

public void setScore(float score)

* public enum TcpType 数据报的枚举类型

ScoreInstr(1), // 控制人员发出打分指令

Score (2), // 裁判的打分分数

ScoreResult(3), // 总裁判返回的打分结果

GetFreeReferee(4), // 控制人员获取空闲裁判列表

FreeList(5), // 服务器返回的空闲裁判列表

SetFree(6), // 裁判一轮比赛结束后将自己设为空闲

Login(7), // 登录，包括裁判、控制人员、总裁判的登录

LoginResult(8), // 服务器返回登录结果，账号密码是否正确

GetEvent(9), // 获取项目列表

EventList(10), // 服务器返回的项目列表

OrderInfo(11), // 服务器根据项目编号返回的出场顺序列表

GetAthleteScore(12), // 根据运动员编号获取其所有项目成绩

AthleteScore(13), // 服务器返回的运动员成绩

GetTeamScore(14), // 根据代表队账号获取其团体成绩

TeamScore(15), // 服务器返回的团体成绩

GetOrderInfo(16), // 根据项目编号和年龄组获取项目列表

SaveScore(17); // 保存总裁判发来的打分分数

TcpType(int index)

public int getIndex()

# 4.客户端的设计

## 4.1设计思路

体操赛事信息管理系统的使用者通过客户端来进行相应的报名、查询、分组、打分等操作，涉及和数据库后台的交互，所以客户端的设计将涉及到发送和接受消息的数据包设计以及数据库操作的内容。

## 4.2 public class client

基于socket进行进程间的通信，调用socket类的构造方法在特定的端口号创建与主机之间的连接：

Socket socket = new Socket("localhost", 2048);

发送消息或接受消息时按TcpType里数据包里的变量及顺序写入或读出：

public static void main(String[] args)

## 4.3 TCP的数据报格式设计

public enum TcpType：

此枚举类型用于详细说明数据包的类型，与客户端进行消息通信时遵从数据包里说明的变量和顺序。

各个数据报格式如下：

* 控制人员发出打分指令：

数据报类型（TcpType.ScoreInstr）；

本身控制人员的账号（String ctrlUser）；

总裁判账号（String mainRefereeUser）；

接收该指令的裁判账号的数组（String[] refereeUser）；

出场顺序列表（OrderInfo[] orders）。

* 裁判的打分分数：

数据报类型（TcpType.Score）；

本身裁判的账号（String refereeUser）；

总裁判账号（String mainRefereeUser）；

打分信息（ScoreResult rs）。

* 总裁判返回的打分结果：

数据报类型（TcpType.ScoreResult）；

本身总裁判的账号（String mainRefereeUser）；

接收结果的裁判账号（String[] refereeUser）；

打分结果（boolean succeed）。

* 控制人员获取空闲裁判列表：

数据报类型（TcpType. GetFreeReferee）。

* 服务器返回的空闲裁判列表：

数据报类型（TcpType. FreeList）；

空闲裁判列表（Referee[] freeList）。

* 裁判一轮比赛结束后将自己设为空闲：

数据报类型（TcpType. SetFree）；

本身的裁判账号（String refereeUser）。

* 登录，包括裁判、控制人员、管理员的登录：

数据报类型（TcpType. Login）；

登录选项（String option）：

“admin”：管理员登录；

“referee”：裁判登录；

“controller”：控制人员登录；

账号（String user）；

密码（String pwd）。

* 服务器返回登录结果，账号密码是否正确：

数据报类型（TcpType. LoginResult）；

登录结果（boolean succeed）。

* 获取项目列表：

数据报类型（TcpType. GetEvent）。

* 服务器返回的项目列表：

数据报类型（TcpType. EventList）；

项目列表（Event[] eventList）。

* 服务器根据项目编号返回的出场顺序列表：

数据报类型（TcpType. GetOrderInfo）；

项目编号（int eventNo）；

年龄组（AgeGroup ageGroup）。

* 服务器根据项目编号返回的出场顺序列表：

数据报类型（TcpType. OrderInfo）；

出场顺序列表（OrederInfo[] orders）。

* 根据运动员编号获取其所有项目成绩：

数据报类型（TcpType. GetAthleteScore）；

运动员编号（String athleteNo）。

* 服务器返回的运动员成绩：

数据报类型（TcpType. AthleteScore）；

运动员成绩列表列表（ScoreResult[] scoreResult）。

* 根据代表队账号获取其团体成绩：

数据报类型（TcpType. GetTeamScore）；

代表队账号（String teamUser）。

* 服务器返回的团体成绩：

数据报类型（TcpType. TeamScore）；

团体成绩（float teamScore）。

## 4.4 数据库设计public class Dao

### 4.4.1类结构

|  |  |  |  |
| --- | --- | --- | --- |
| Dao | | | |
| 成员方法 | 参数 | 返回值 | 备注 |
| connect | String url, String usr, String pwd | void | 连接到数据库 |
| login | String option, String user, String pwd | boolean | 登录账号，根据option的值登录对应账号，密码正确返回true，否则返回false |
| getAllAthlete | 无 | Athlete[] | 获取所有运动员 |
| setAthleteNo | String idNumber, int athleteNo | void | 设置运动员编号 |
| setAthleteOrder | int athleteNo, int eventNo, int groupNo, int order | void | 设置运动员出场顺序 |
| getOrderInfo | int eventNo, AgeGroup ageGroup | Vector<OrderInfo> | 通过项目编号和年龄组获取出场信息 |
| getEvent | 无 | Vector<Event> | 获取项目列表 |
| saveScore | ScoreResult rs | void | 保存成绩 |
| getScoreByAthleteNo | int athleteNo | ScoreResult[] | 通过运动员编号获取成绩 |
| getAthletesByTeamName | String teamName | int[] | 通过队名获取运动员编号列表 |

### 4.4.2类成员设计

* 成员变量

private Connection con;

private PreparedStatement prepstmt;

* 成员方法

public Dao()

public Dao(String url, String usr, String pwd)

private void connect(String url, String usr, String pwd)

public boolean login(String option, String user, String pwd)

public Athlete[] getAllAthlete()

public void setAthleteNo(String idNumber, int athleteNo)

public void setAthleteOrder(int athleteNo, int eventNo, int groupNo, int order)

public Vector<OrderInfo> getOrderInfo(int eventNo, AgeGroup ageGroup)

public Vector<Event> getEvent()

public void saveScore(ScoreResult rs)

public int[] getAthletesByTeamName(String teamName)

### 4.4.3方法login设计

该方法是用来登录账号，根据option的值登录对应账号，密码正确返回true，否则返回false。

* 方法原型

public boolean login(String option, String user, String pwd)

* 方法说明

本方法通过输入账号、密码以及登录身份，根据option的值连接数据库选择对应账号及进行登录。

* 输入、输出数据元素

输入数据：登录身份、账号、密码

输出数据：是否成功登录

### 4.4.4 getOrderInfo方法设计

该方法是用来通过项目编号和年龄组获取出场信息。

* 方法原型

public Vector<OrderInfo> getOrderInfo(int eventNo, AgeGroup ageGroup)

* 方法说明

通过项目编号和年龄组获取出场信息，根据输入的项目编号和年龄组，连接数据库进行相关select选择，获得出场信息。

* 输入、输出数据元素

输入数据：编号、年龄组

输出数据：对应的出场信息

# 5.服务器的设计

## 5.1设计思路

服务器对客户端传回的消息进行相应处理相应，赛事的工作人员（裁判、比赛控制人员）在服务器端进行比赛控制、比赛评分的操作，同客户端一样涉及到与数据库的交互，数据库类的设计与客户端的通用。

## 5.2public class SocketHandle

### 5.2.1类结构设计

|  |  |  |  |
| --- | --- | --- | --- |
| SocketHandle | | | |
| 成员方法 | 参数 | 返回值 | 备注 |
| run | 无 | void | 接受客户端消息并进行处理 |
| login | 无 | void | 登录 |
| scoreInstr | 无 | void | 控制人员发出打分指令 |
| score | 无 | void | 裁判的打分分数 |
| scoreResult | 无 | void | 总裁判返回的打分结果 |
| getFreeList | 无 | void | 控制人员获取空闲裁判列表 |
| setFree | 无 | void | 裁判一轮比赛结束后将自己设为空闲 |
| getEvent | 无 | void | 获取项目列表 |
| getOrderInfo | 无 | void | 服务器根据项目编号返回的出场顺序列表 |
| getTeamScore | 无 | void | 根据代表队账号获取其团体成绩 |
| getAthleteScore | 无 | void | 根据运动员编号获取其所有项目成绩 |

### 5.2.2类成员设计

* 成员变量

private Socket client;

private Dao dao;

private Vector<String> freeList;

private Map<String, SocketHandle> refereeList;

public ObjectOutputStream out;

private ObjectInputStream in;

* 成员方法

public void run()

private void login()

private void scoreInstr()

private void score()

private void scoreResult()

private void getFreeList()

private void setFree()

private void getEvent()

private void getOrderInfo()

private void getTeamScore()

private void getAthleteScore()

接受客户端消息并进行相应的处理，赛事工作人员相关操作的处理，获取项目信息，进行打分的处理

## 5.3public class Server

* 成员变量

private static Dao dao = new Dao();

private static Vector<String> freeList = new Vector<>();

private static Map<String, SocketHandle> refereeList = new HashMap<>();

处理与客户端、数据库建立连接，进行socket的侦听

# 6.数据库表的设计

* 系统管理员Admin表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可为空 | 说明 |
| adminUser | Varchar | 否（主键） | 账号 |
| adminPwd | Varchar | 否 | 密码 |

* 比赛控制人员Controller表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可为空 | 说明 |
| ctrlUser | Varchar | 否（主键） | 账号 |
| ctrlPwd | Varchar | 否 | 密码 |

* 裁判员Referee表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可未空 | 说明 |
| idNumber | String | 否（主键） | 身份证号 |
| phone | Bigint | 否 | 电话 |
| name | Varchar | 否 | 姓名 |
| teamName | Varchar | 否 | 队名 |
| refereeUser | Varchar | 否 | 账号 |
| refereePwd | Varchar | 否 | 密码 |

* 代表队Team表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可为空 | 说明 |
| teamUser | Varchar | 否（主键） | 队账号 |
| teamName | Varchar | 否 | 队名 |
| teamPwd | Varchar | 否 | 密码 |

* 队伍工作人员 Staff表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可未空 | 说明 |
| name | Varchar | 否（主键） | 姓名 |
| idNumber | String | 否 | 身份证号 |
| phone | Bigint | 否 | 电话 |
| sex | String | 否 | 性别 |
| duty | Varchar | 否 | 职务 |
| teamName | Varchar | 否 | 队名 |

* 运动员 Athlete表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可未空 | 说明 |
| athleteNo | Int | 否（主键） | 运动员编号 |
| idNumber | String | 否 | 身份证号 |
| name | Varchar | 否 | 姓名 |
| sex | String | 否 | 性别 |
| ageGroup | Int | 否 | 年龄组 |
| academicRecord | Float | 否 | 文化成绩 |

* 年龄组 AgeGroup表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可为空 | 说明 |
| ageMax | Int | 否 | 年龄上界 |
| ageMin | Int | 否 | 年龄下界 |

* 比赛项目 Event表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可为空 | 说明 |
| eventNo | Int | 否（主键） | 项目编号 |
| sex | String | 否 | 性别 |
| eventName | Varchar | 否 | 项目名 |

* 赛事表 Match表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可未空 | 说明 |
| eventNo | Int | 否 | 项目编号 |
| matchNo | Int | 否（主键） | 比赛编号 |
| matchType | String | 否 | 比赛类型 |
| matchTime | Time | 否 | 比赛时间 |
| preOrFinal | String | 否 | 预决赛 |
| groupNo | Int | 否 | 组号 |

* 出场顺序信息 OrderInfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可为空 | 说明 |
| matchNo | Int | 否（主键） | 比赛编号 |
| athleteNo | Int | 否 | 运动员编号 |
| order | Int | 否 | 出场编号 |

* 赛事结果 ScoreResult表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 是否可未空 | 说明 |
| athleteNo | Int | 否 | 运动员编号 |
| eventNo | Int | 否（主键） | 项目编号 |
| score | Float | 否 | 得分 |
| scoreType | String | 否 | 打分类型 |
| refereeUser | Varchar | 否 | 裁判账号 |