NBA Analysis 项目设计文档

作者：组48\_高效率小组

目录

[NBA Analysis 项目设计文档 1](#_Toc414485071)

[作者：组48 1](#_Toc414485072)

[1 前言 2](#_Toc414485073)

[1.1 发布日期 2](#_Toc414485074)

[1.2 团队 2](#_Toc414485075)

[1.3 成员 2](#_Toc414485076)

[1.4 变更历史 2](#_Toc414485077)

[2 总体介绍 3](#_Toc414485078)

[2.1 编制目的 3](#_Toc414485079)

[2.2 对象 3](#_Toc414485080)

[2.3 产品概述 3](#_Toc414485081)

[2.4 参考资料 3](#_Toc414485082)

[2.5 词汇表 3](#_Toc414485083)

[3 系统设计描述主体 4](#_Toc414485084)

[3.1 展示层设计 4](#_Toc414485085)

[3.1.1 展示层分解图 4](#_Toc414485086)

[3.1.2 展示层模块的职责 4](#_Toc414485087)

[3.1.3 展示层模块的接口规范 4](#_Toc414485088)

[3.2 逻辑层设计 6](#_Toc414485089)

[3.2.1 概述 6](#_Toc414485090)

[3.2.2 接口规范 6](#_Toc414485091)

[3.3 数据层设计 8](#_Toc414485092)

[3.3.1 数据层概述 8](#_Toc414485093)

[3.3.2 数据层借口规范 8](#_Toc414485094)

[4 信息视角 10](#_Toc414485095)

[4.1 数据持久化对象 10](#_Toc414485096)

[4.2 Txt数据文件格式 11](#_Toc414485097)

[5 开发包图 12](#_Toc414485098)

# 前言

## 发布日期

2015年3月6日

## 团队

组48

## 成员

陈建伟、梅杰、王俊超、刘瀚文

## 变更历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 原因 | 版本号 |
| 刘瀚文 | 2015/3/6 | 建立文档模板 | V0.1 |
| 梅杰 | 2015/3/7 |  | V0.2 |
| 刘瀚文 | 2015/3/7 | 完成数据持久化对象设计 | V0.3 |
| 梅杰 | 2015/3/8 | 完善展示层接口 | V0.4 |
| 王俊超 | 2015/3/13 | 完成数据层设计 | V0.5 |
| 刘瀚文 | 2015/3/13 | 整理并完成文档 | V1.0 |
|  |  |  |  |

# 总体介绍

## 编制目的

本文档编写了NBA数据分析系统的架构概览，是用于构造系统阶段的重要架构决策。

## 对象

本文档都这是本小组全部成员。

## 产品概述

参见《CSEIII项目概述及迭代一需求说明》。

## 参考资料

1.《软件工程与计算（卷三）：团队与软件开发实践》

2.IEEE 1016-2009

## 词汇表

|  |  |
| --- | --- |
| 词汇名称 | 词汇含义 |
| NBA | 美国及加拿大职业篮球联盟 |
|  |  |

# 系统设计描述主体

## 展示层设计

### 展示层分解图

单支球队展示界面

球队选择界面

所有球队数据排名界面

所有球员选择界面

主界面

单个球员的数据展示界面

球员的数据筛选排名展示界面

### 展示层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| MainFrame | 界面Frame，负责界面的显示和界面的跳转。 |
| TeamUI | 负责球队的数据显示 |
| PlayerUI | 负责球员的数据显示 |

### 展示层模块的接口规范

用户界面层模块的接口规范如表3.1.3-1所示。

表3.1.3-1 用户界面层模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| MainFrame | 语法 | | Init(args[]) |
| 前置条件 | | 无 |
| 后置条件 | | 显示Frame |
| TeamUI | 语法 | | Init(args[]) |
| 前置条件 | | 启动球队界面 |
| 后置条件 | | 显示球队界面 |
| PlayerUI | 语法 | | Init(args[]) |
| 前置条件 | | 启动球员界面 |
| 后置条件 | | 显示球员界面 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| businesslogicservice.\*BLService | | 每个界面都有一个相应的业务逻辑接口 | |
| Businesslogicservice.getAllTeamInfo() | | 得到所有球队基本信息，以及已经计算好的比赛数据 | |
| Businesslogicservice.getSpecificTeamInfo(TeamVO vo) | | 得到特定球队基本信息，以及已经计算好的比赛数据 | |
| Businesslogicservice.getSpecificPlayerInfo(PlayerVO vo) | | 得到特定球员基本信息和已经计算好的比赛数据 | |

## 逻辑层设计

### 概述

业务逻辑层负责分析由数据层传入的球员及球队的原始数据，经过公式计算后，将分析数据发送至展示层。为了保证层间的信息隐藏，业务逻辑层与数据层公用POJO对象，使用data service接口向下调用；与展示层公用VO对象，使用blservice向上提供接口，保证了内部实现的相对独立性。

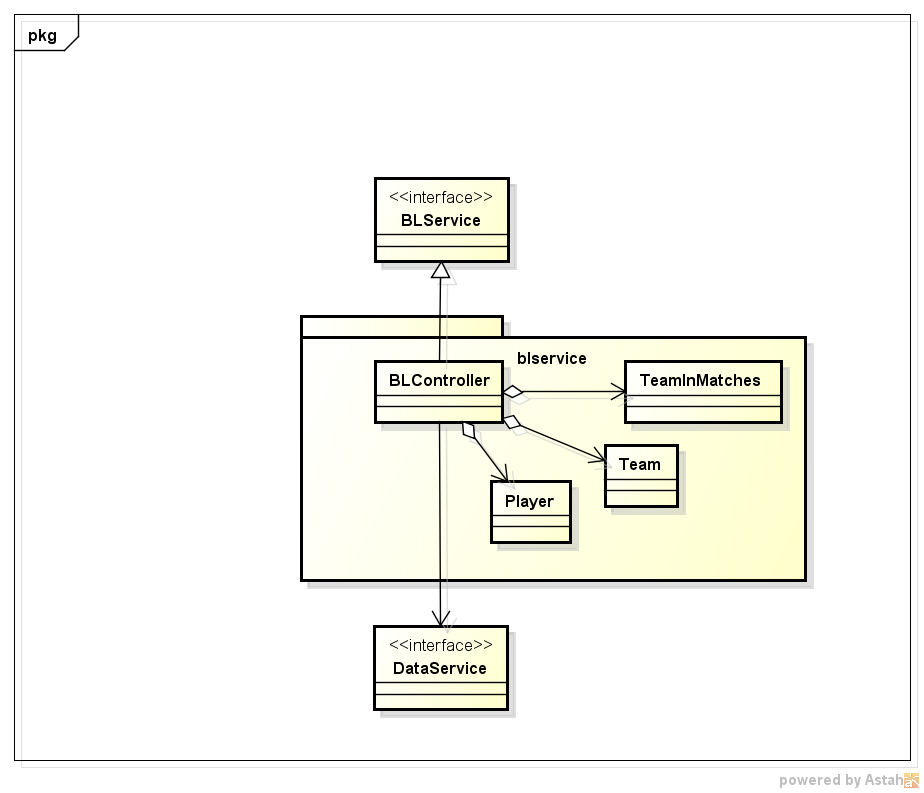


Figure 1逻辑层类图

### 接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| 接口名称 | 接口信息 | |
| blservice.getPlayerAnalysis | 语法 | ArrayList<PlayerVO> getPlayerAnalysis() |
| 前置条件 | 查询所有球员信息 |
| 后置条件 | 返回未经排序的所有球员列表 |
| blservice.getTeamAnalysis | 语法 | ArrayList<TeamVO> getTeamAnalysis() |
| 前置条件 | 查询所有球队信息 |
| 后置条件 | 返回未经排序的所有球队列表 |
| getPlayerAnalysis | 语法 | PlayerVO getPlayerAnalysis(String name) |
| 前置条件 | 输入球员名字 |
| 后置条件 | 返回这位球员的信息，没有找到球员返回NULL |
| TeamVO getTeamAnalysis | 语法 | TeamVO getTeamAnalysis(String name) |
| 前置条件 | 输入球员名字 |
| 后置条件 | 返回这支球队的信息，没有找到球队返回NULL |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| getPlayerPics | 得到一位球员的两张图片 | |
| getTeamPic | 得到一支球队的队徽 | |
| getAllTeams | 得到一系列球队持久化对象 | |
| getAllMatch | 得到一系列比赛持久化对象 | |
| getAllPlayers | 得到一系列球员持久化对象 | |

## 数据层设计

### 数据层概述

数据层负责分析由文件读入的球员及球队的原始数据，经过分析处理后，将分析数据分类打包成PO发送至逻辑层层。使用DataService向逻辑层提供接口，保证相对独立性

### 数据层借口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| 接口名称 | 接口信息 | |
| DataService.getAllPlayers | 语法 | HashMap<String, PlayerPO> getAllPlayers(); |
| 前置条件 | 查询所有球员信息 |
| 后置条件 | 返回未经排序的所有球员hash表 |
| DataService.getAllTeams | 语法 | HashMap<String, TeamPO> getAllTeams(); |
| 前置条件 | 查询所有球队信息 |
| 后置条件 | 返回未经排序的所有球队hash表 |
| DataService.getAllMatch | 语法 | ArrayList<MatchPO> getAllMatch(); |
| 前置条件 | 查询所有比赛信息 |
| 后置条件 | 返回所有比赛信息列表 |
| DataService.getPlayerAction | 语法 | ImageIcon getPlayerAction (String name) |
| 前置条件 | 输入球员名字 |
| 后置条件 | 返回球员动作图片ImageIcon,没有找到球员返回NULL |
| DataService.getPlayerPortrait | 语法 | ImageIcon getPlayerPortrait(String name); |
| 前置条件 | 输入球员名字 |
| 后置条件 | 返回球员的肖像图片ImageIcon，没有找到球员返回NULL |
| DataService.getTeamPic | 语法 | ImageIcon getTeamPic(String abbservation) |
| 前置条件 | 输入球队简称 |
| 后置条件 | 返回这支球队的队徽ImageIcon，没有找到球队返回NULL |

# 信息视角

## 数据持久化对象

|  |  |
| --- | --- |
| 类名 | 说明 |
| HeightPO | 英尺、英寸 |
| MatchPO | 包含赛季、时间、全场比分、各节比分、两个参赛队伍(TeamInMatchesPO) |
| PlayerPO | 球员信息，见下图 |
| TeamInMatchesPO | 包含球队简称和上场队员(PlayerInMatchesPO)表 |
| PlayerInMatchesPO | 包含球员本场比赛数据 |
| TeamPO | 球队信息，见下图 |
| ScorePO | 两个int类型比分 |

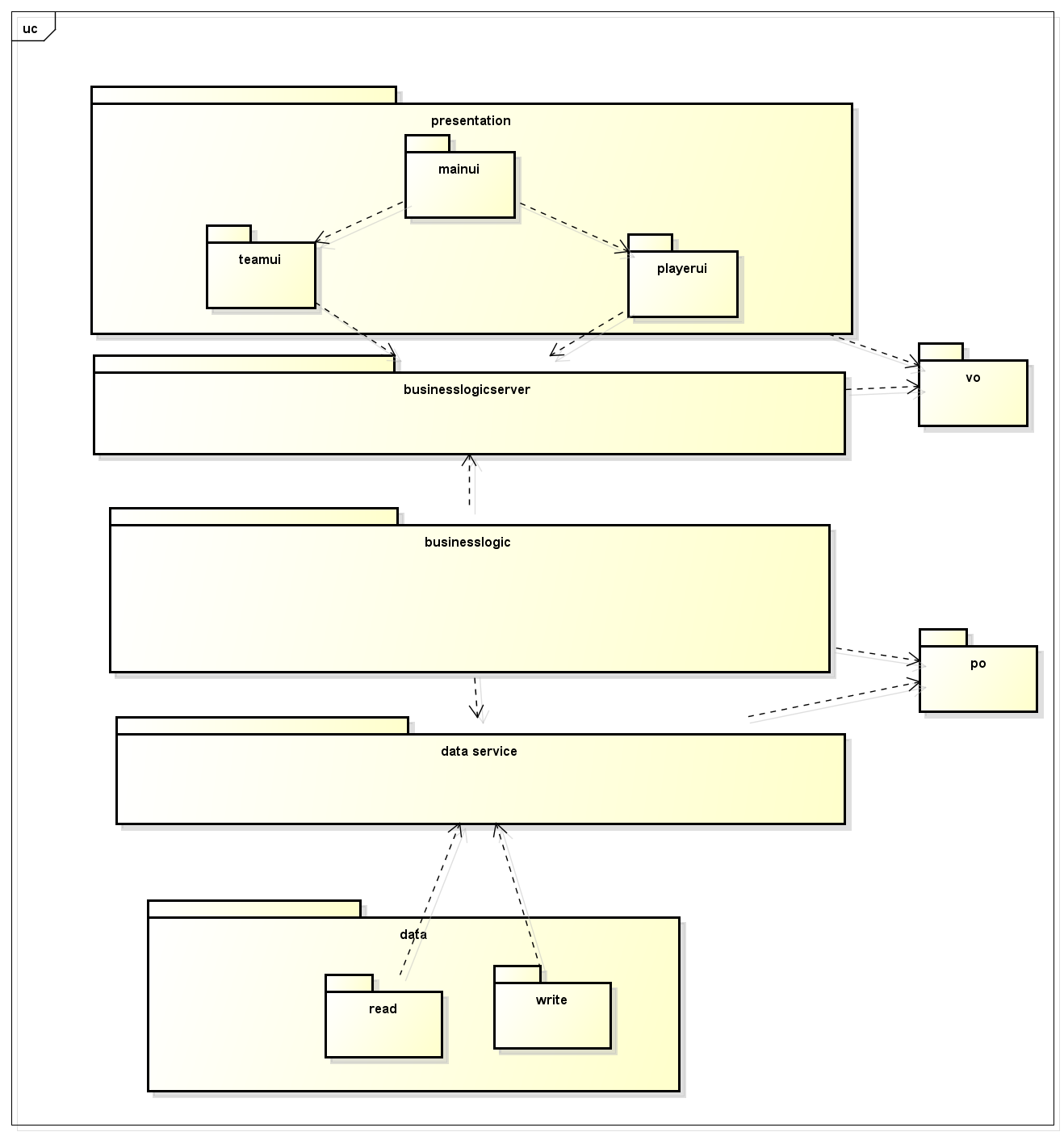
Table 1数据持久化对象编码实例

|  |
| --- |
| PlayerPO节选 |
| **public** **class** PlayerPO {  **public** PlayerPO(String name, **int** number, **char** position, HeightPO height,  **int** weight, Calendar birth, **int** age, **int** exp, String school) {  **super**();  **this**.name = name;  **this**.number = number;  **this**.position = position;  **this**.height = height;  **this**.weight = weight;  **this**.birth = birth;  **this**.age = age;  **this**.exp = exp;  **this**.school = school;  }  String name;  **int** number;  **char** position;  HeightPO height;  **int** weight;  Calendar birth;  **int** age;  **int** exp;  String school; |
| TeamPO节选 |
| **public** **class** TeamPO {  **public** TeamPO(String fullName, String abbreviation, String location,  **char** division, String zone, String home, **int** setupTime) {  **super**();  **this**.fullName = fullName;  **this**.abbreviation = abbreviation;  **this**.location = location;  **this**.division = division;  **this**.zone = zone;  **this**.home = home;  **this**.setupTime = setupTime;  }  String fullName;  String abbreviation;  String location;  **char** division;  String zone;  String home;  **int** setupTime; |

## Txt数据文件格式

见《CSEIII数据说明》

# 开发包图



# 顺序图