**小学期CPP课报告之View Model part**

课题：篮球数据可视化（俱乐部数据以及球员个人数据的收集与可视化分析）

本模块负责人：林灿东 - 计科1601 – 3160103968

其他同组组员：梁家宁 - 计科1601 – 3160105677

谭泽源 – 计科 -

**模块功能阐述：**

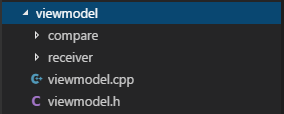
本模块负责承接view模块以及model模块。一方面，model中的数据/算法返回值不一定可直接为view使用，所以view model（简称vm）需要设置一些**数据**变量将model中的数据转化成view可用的格式。这些变量将与view进行**绑定**，而且由于view与vm不互知，vm需要向view**暴露绑定的变量**。并且当model中的数据发生更新时，vm需要设置一个**通知接收器**，以便接收来自model的关于数据更新的通知，进而更新在vm中的相关变量。同时，当vm接收到此类通知时，需要通过一个**通知发送器**对view发送通知，告诉view数据更新了，应该相应地更新数据在界面上的显示。另一方面，当view需要某个数据时（已在vm中有相应的绑定数据），view是通过执行命令来让vm更新绑定的数据，进而实现view中数据的更新。而这些**命令**都需要在vm中实例化，并且也需要向view**暴露命令**。同时，当命令执行完后，vm需要通过另一个**通知发送器**告诉view请求的数据已经可用，可以更新界面显示了。并且vm中的两个通知发送器都需要**绑定对应的接收器**。

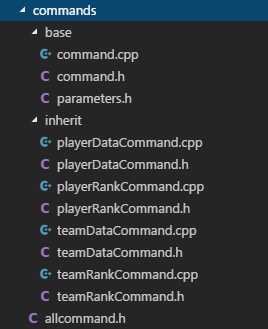
**补充说明：**

1.由于在mvvm模式下，数据、命令、接收器与发送器之间的绑定都是通过绑定指针来实现的，所以为了方便管理，都统一使用智能指针。

2.本工程中，由于业务流程，model中的数据是固定的不会发生变化，因此去除了vm中的属性改变通知发送器以及view中的属性改变通知接收器。同时，在vm中设置一些临时变量。每当view请求数据时，vm执行命令将数据存在临时变量中，当vm向view发送命令完成通知时，vm再将临时变量的值赋给绑定的数据，进而实现view中数据的更新。

**vm相关文件目录：**





**源码解读：**

**ViewModel主体部分：**

class viewModel

{

private:

//绑定的数据

shared\_ptr<vector<shared\_ptr<player\_avg>>> PlayerRank;

shared\_ptr<vector<shared\_ptr<player\_avg>>> PlayerData;

shared\_ptr<vector<shared\_ptr<team\_avg>>> TeamData;

shared\_ptr<vector<shared\_ptr<team\_avg>>> TeamRank;

//临时变量

shared\_ptr<vector<shared\_ptr<player\_avg>>> tmpPlayer;

shared\_ptr<team\_avg> tmpTeam;

//接收器，负责接收model部分的属性改变通知

shared\_ptr<receiverFromModel> receiveModel;

public:

//暴露绑定的数据

shared\_ptr<vector<shared\_ptr<player\_avg>>> getPlayer();

shared\_ptr<vector<shared\_ptr<player\_avg>>> getPlayerRank();

shared\_ptr<vector<shared\_ptr<team\_avg>>> getTeam();

shared\_ptr<vector<shared\_ptr<team\_avg>>> getTeamRank();

//暴露接收器与发送器

shared\_ptr<receiverFromModel> getModelReceiver();

void setSpc(shared\_ptr<propertyReceiver> ptr);

void setScc(shared\_ptr<commandReceiver> ptr);

//计算临时数据的算法

void RequestPlayerData(shared\_ptr<dataParameter> ptr);

void RequestPlayerRank(shared\_ptr<rankParameter> ptr);

void RequestTeamData(shared\_ptr<dataParameter> ptr);

void RequestTeamRank(shared\_ptr<rankParameter> ptr);

//将临时数据更新到绑定数据上

void UpdatePlayerData();

void UpdatePlayerRank();

void UpdateTeamData();

void UpdateTeamRank();

//命令部分

//实例化的命令

shared\_ptr<playerDataCommand> PlayerDataCmd;

shared\_ptr<playerRankCommand> PlayerRankCmd;

shared\_ptr<teamDataCommand> TeamDataCmd;

shared\_ptr<teamRankCommand> TeamRankCmd;

//向view暴露命令

shared\_ptr<command> getPlayerDataCmd();

shared\_ptr<command> getPlayerRankCmd();

shared\_ptr<command> getTeamDataCmd();

shared\_ptr<command> getTeamRankCmd();

//其他成员

//内含的model部分

model m;

//属性改变通知发送器

senderPropertyChange spc;

//命令完成通知发送器

senderCommandComplete scc;

viewModel();

};

**命令部分：**

class command

{

protected:

//命令的参数

std::shared\_ptr<parameters> parameter;

public:

command();

//暴露设置参数的接口

void setParameter(std::shared\_ptr<parameters> para);

//执行命令（虚函数）

virtual void action() = 0;

};

**参数部分：**

class parameters {};

class dataParameter : public parameters

{

public:

std::string name; // 球队/球员名字

std::string option; // 球队/球员哪项数据的历史

std::string begin;

std::string end;

dataParameter(std::string name, std::string option, std::string begin, std::string end) : name(name), option(option), begin(begin), end(end) {}

}; // 绑定一个vector，存储球队/球员的该项数据的历史记录

class rankParameter : public parameters

{

public:

std::string option; // 用来排序的选项，按照爬取的名字来看

int season; // 赛季

rankParameter(std::string option, int season): option(option), season(season) {}

}; // 绑定一个大小为15的vector，存储前15名球队/球员的该项数据