

야구 투수 2

- (1) 생성자, 소멸자 이용
- (2) 객체 지향적 기법으로 투수 클래스와 팀 클래스 설계
- (3) vector 이용
- (4) 분할 컴파일

○ Pitcher.h

```
#ifndef PITCHER_H
#define PITCHER_H

#include <string>
using namespace std;

class Pitcher
{
protected:
    string name;
    double no_of_inning;    // 이닝 수
    int no_of_opp_hits;    // 피안타 수
    int no_of_opp_runs;    // 실점 수
    int no_of_res_opp_runs; // 자책점 수
    int no_of_wins;        // 승리 수
    int no_of_loses;       // 패전 수
    int no_of_saves;       // 세이브 수

public :
    Pitcher();              // 생성자
    Pitcher(string name);  // 생성자

    string get_name();
    double get_no_of_inning();
    int get_no_of_opp_hits();
    int get_no_of_opp_runs();
    int get_no_of_res_opp_runs();
    int get_no_of_wins();
    int get_no_of_loses();
    int get_no_of_saves();
};
```

```
void set_name();  
// void init_stat();  
  
void add_out();  
void add_opp_hits();  
void add_res_opp_runs();  
void add_opp_runs();  
void add_win();  
void add_lose();  
void add_save();  
  
double get_era();  
double get_winning_rate();  
  
void show_stat();  
};  
#endif
```

○ Pitcher.cpp

```
#include "stdafx.h"
#include <iostream>
#include <string>
#include <time.h>

#include "Pitcher.h"
using namespace std;

Pitcher::Pitcher()
{
    no_of_inning = no_of_opp_hits = no_of_opp_runs = no_of_res_opp_runs =
no_of_wins = no_of_loses = no_of_saves = 0;
}

Pitcher::Pitcher(string str)
{
    name = str;
    no_of_inning = no_of_opp_hits = no_of_opp_runs = no_of_res_opp_runs =
no_of_wins = no_of_loses = no_of_saves = 0;
}

string Pitcher::get_name()
{
    return name;
}

double Pitcher::get_no_of_inning()
{
    return no_of_inning;
}

int Pitcher::get_no_of_opp_hits()
{
    return no_of_opp_hits;
}

int Pitcher::get_no_of_opp_runs()
{
    return no_of_opp_runs;
}
```

```
int Pitcher::get_no_of_res_opp_runs()
{
    return no_of_res_opp_runs;
}

int Pitcher::get_no_of_wins()
{
    return no_of_wins;
}

int Pitcher::get_no_of_loses()
{
    return no_of_loses;
}

int Pitcher::get_no_of_saves()
{
    return no_of_saves;
}

void Pitcher::set_name()
{
    cout << "Enter player name : ";
    cin >> name;
}

// void Pitcher::init_stat()
// {
//     no_of_inning = 0.;
//     no_of_opp_hits = 0;
//     no_of_opp_runs = 0;
//     no_of_res_opp_runs = 0;
//     no_of_wins = 0;
//     no_of_loses = 0;
//     no_of_saves = 0;
// }

void Pitcher::add_out()
{
    no_of_inning = no_of_inning + 1./3.;
```

```
}
```

```
void Pitcher::add_opp_hits()
```

```
{
```

```
    no_of_opp_hits++;
```

```
}
```

```
void Pitcher::add_res_opp_runs()
```

```
{
```

```
    no_of_opp_runs++;
```

```
    no_of_res_opp_runs++;
```

```
}
```

```
void Pitcher::add_opp_runs()
```

```
{
```

```
    no_of_opp_runs++;
```

```
}
```

```
void Pitcher::add_win()
```

```
{
```

```
    no_of_wins++;
```

```
}
```

```
void Pitcher::add_lose()
```

```
{
```

```
    no_of_loses++;
```

```
}
```

```
void Pitcher::add_save()
```

```
{
```

```
    no_of_saves++;
```

```
}
```

```
double Pitcher::get_era()
```

```
{
```

```
    if (no_of_inning > 0)
```

```
        return (double) no_of_res_opp_runs * 9.0 / no_of_inning;
```

```
    else
```

```
        return 0.;
```

```
}
```

```

double Pitcher::get_winning_rate()
{
    if (no_of_wins + no_of_loses > 0)
        return (double) no_of_wins / (no_of_wins + no_of_loses);
    else
        return 0.;
}

void Pitcher::show_stat()
{
    cout << "=====" << endl;
    cout << "Name          : " << name << endl;
    cout << "No of Inning      : " << no_of_inning << endl;
    cout << "No of Opp Hits    : " << no_of_opp_hits << endl;
    cout << "No of Opp Runs    : " << no_of_opp_runs << endl;
    cout << "No of Res Opp Runs : " << no_of_res_opp_runs << endl;
    cout << "No of Wins        : " << no_of_wins << endl;
    cout << "No of Loses       : " << no_of_loses << endl;
    cout << "No of Saves       : " << no_of_saves << endl;
    cout << "ERA               : " << get_era() << endl;
    cout << "Winning Rate      : " << get_winning_rate() << endl;
    cout << "=====" << endl;
}

```

○ Baseball_team.h

```
#ifndef BASEBALL_TEAM_H
#define BASEBALL_TEAM_H

#include "Pitcher.h"

#include <vector>
#include <string>
using namespace std;

class Baseball_team
{
protected:
    string name;
    vector <Pitcher *> pitchers_list;

public :
    Baseball_team(string name);    // 생성자
    ~Baseball_team();             // 소멸자

    string get_name();
    void set_name();

    int get_no_of_pitchers();
    void add_pitcher();

    void make_pitcher_stat(string pitcher_name, int no_of_tries);

    Pitcher *get_best_win_rate_pitcher();

    void show_pitcher_stat(int index);
    void show_all_pitcher_names();
    void show_all_pitcher_stats();
};
#endif
```

○ Baseball_team.cpp

```
#include "stdafx.h"
#include <iostream>
#include "Baseball_team.h"

Baseball_team::Baseball_team(string str)
{
    name = str;
}

Baseball_team::~Baseball_team()
{
    for (unsigned int i = 0; i < pitchers_list.size(); i++)
        delete pitchers_list[i];
    pitchers_list.clear();
}

string Baseball_team::get_name()
{
    return name;
}

void Baseball_team::set_name()
{
    cout << "Enter team name : ";
    cin >> name;
}

int Baseball_team::get_no_of_pitchers()
{
    return (pitchers_list.size());
}

void Baseball_team::add_pitcher()
{
    Pitcher *temp = new Pitcher;
    cout << "Adding a pitcher " << endl;
    temp->set_name();

    pitchers_list.push_back(temp);
}
```



```

void Baseball_team::make_pitcher_stat(string pitcher_name, int no_of_tries)
{
    bool not_found = true;
    int rnd_number;
    Pitcher *player;

    for (unsigned int i = 0; not_found && (i < pitchers_list.size()); i++) {
        if (pitchers_list[i]->get_name() == pitcher_name) {
            not_found = false;
            player = pitchers_list[i];
        }
    }

    if (not_found) {
        cout << "No such pitcher " << endl;
        return;
    }

    for (int i = 0; i < no_of_tries; i++) {
        rnd_number = rand() % 100;

        if (rnd_number > 75) {
            player->add_opp_hits();

            if (rnd_number > 98)
                player->add_lose();
            else if (rnd_number > 97)
                player->add_res_opp_runs();
            else if (rnd_number > 95)
                player->add_opp_runs();
        }
        else {
            player->add_out();

            if (rnd_number < 2)
                player->add_win();
            else if (rnd_number < 3)
                player->add_save();
        }
    }
}

```

```

void Baseball_team::show_pitcher_stat(int index)
{
    if (index < pitchers_list.size())
        pitchers_list[index]->show_stat();
    else
        cout << "Invalid index number !" << endl;
}

void Baseball_team::show_all_pitcher_names()
{
    for (unsigned int i = 0; i < pitchers_list.size(); i++)
        cout << i + 1 << " " << pitchers_list[i]->get_name() << endl;
}

void Baseball_team::show_all_pitcher_stats()
{
    for (unsigned int i = 0; i < pitchers_list.size(); i++)
        pitchers_list[i]->show_stat();
}

Pitcher* Baseball_team::get_best_win_rate_pitcher()
{
    double temp;
    double current_best_value;
    int current_best_index = -1;

    int no_of_pitchers = pitchers_list.size();

    if (no_of_pitchers > 0) {
        current_best_index = 0;
        current_best_value = pitchers_list[0]->get_winning_rate();

        for (int i = 1; i < no_of_pitchers; i++) {
            temp = pitchers_list[i]->get_winning_rate();

            if (current_best_value < temp) {
                current_best_value = temp;
                current_best_index = i;
            }
        }
    }
    return pitchers_list[current_best_index];
}

```

```

    }
    else
        return ((Pitcher *)NULL);
}

```

○ Pitcher2.cpp

```

#include "stdafx.h"
#include <iostream>
#include <time.h>
#include "Baseball_team.h"

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    Baseball_team lions("Lions");
    string temp_name;

    srand((unsigned int)time(NULL));

    for (int i = 0; i < 10; i++)
        lions.add_pitcher();

    cout << "Make Stats" << endl << endl;
    for (int i = 0; i < 10; i++) {
        cout << "Enter pitcher name : ";
        cin >> temp_name;
        lions.make_pitcher_stat(temp_name, 100);
    }

    for (int i = 0; i < lions.get_no_of_pitchers(); i++)
        lions.show_pitcher_stat(i);

    lions.show_all_pitcher_names();
    lions.show_all_pitcher_stats();

    (lions.get_best_win_rate_pitcher())->show_stat();

    return 0;
}

```