

C++ 중간고사

※ 주의사항

- ① 시험시간이 겹치는 학생들은 그 사유를 명확히 밝혀 불이익을 당하는 일이 없도록 하세요. 겹치는 시험 시간은 지연 시간에서 감하겠습니다.
- ② 학번 끝자리가 홀수번인 학생과 짝수번인 학생의 문제가 다르므로 유의하세요. (다른 문제를 푸는 경우 부정행위로 간주합니다.)
- ③ 경우에 따라서 나중에 프로그램을 직접 실행해서 보여달라고 요구할 수 있으니 프로그램 소스 관리 철저히 하세요. (시연을 요청했을 때 실행하는 것을 못 보이면 부정행위를 한 것으로 간주합니다.)
- ④ 시험 문제 게시 후 3시간 이내 감점이 없고, 6시간 이내 제출하면 10%, 12 시간 이내 제출하면 20% 감점, 24시간 이내 30% 감점하겠습니다. (문제별로 감점)
- ⑤ 5장에서 배운 객체지향 프로그래밍으로 문제를 해결하세요.
- ⑥ 문제별로 프로그램 설명, 소스, 출력결과 캡처 화면을 하나의 파일로 묶어서 kgu@mmu.ac.kr로 제출
- ⑦ 메일 제목은 [학번 이름 중간고사 1], [학번 이름 중간고사 2]로 작성해서 보내세요.

[홀수]

1. 프로야구 타자 기록 관리 프로그램

- (1) 타자 이름, 타석 수, 타수, 안타 수, 홈런 수, 볼넷 등 사사구 수를 유지하는 객체 생성
- (2) 각각의 정보를 알려주는 멤버 함수 작성
- (3) 홈런 추가 멤버 함수 작성 : 타석수, 타수 1 증가, 안타 수, 홈런 수 1 증가
- (4) 안타 추가 멤버 함수 작성 : 타석수, 타수 1 증가, 안타 수 1 증가
- (5) 사사구 추가 멤버 함수 작성 : 타석 수 1 증가, 사사구 수 1 증가
- (6) 아웃 추가 멤버 함수 작성 : 타석 수, 타수 1 증가
- (7) 현재 타율을 알려주는 멤버 함수 작성 : 안타수 / 타수
- (8) 현재 출루율을 알려주는 멤버 함수 작성 : (안타수 + 사사구 수) / 타석 수
- (9) 메인함수에서 해당 객체를 생성 후 각 멤버 함수를 수행한 결과를 보이시오.

2. 1번 문제에서 만든 클래스를 이용하여 10명 이상의 선수들을 이용하여, 가장 타율이 높은 선수, 가장 홈런이 많은 선수, 가장 출루율이 높은 선수들의 이름과 해당 기록을 출력하시오.

[짜수]

1. 프로야구 투수 기록 프로그램

- (1) 투수 이름, 투구 이닝 수, 피안타 수, 실점 수, 자책점 수, 승리, 패전, 세이브 기록을 유지하는 객체 생성
- (2) 각각의 정보를 알려주는 멤버 함수 작성
- (3) 아웃 추가 멤버 함수 작성 : 1 아웃이면 1/3 이닝 추가, 3아웃이면 1 이닝 추가
- (4) 피안타 추가 멤버 함수
- (5) 자책점 추가 멤버 함수 : 자책점, 실점을 같은 값 증가
- (6) 비자책점 추가 멤버 함수 : 실점만 증가
- (7) 승리/패전/세이브 추가 멤버 함수 작성
- (8) 현재 방어율을 알려주는 멤버 함수 작성 : 자책점 수 / 투구 이닝 수
- (9) 승률을 알려주는 멤버 함수 작성 : 승리 / (승리 + 패전)
- (10) 메인함수에서 해당 객체를 생성 후 각 멤버 함수를 수행한 결과를 보이시오.

2. 1번 문제에서 만든 클래스를 이용하여 10명 이상의 선수들을 이용하여, 가장 승률이 높은 선수, 가장 방어율이 낮은 선수, 가장 피안타수가 적은 선수들의 이름과 해당 기록을 출력하시오.

```

#include "stdafx.h" // Visual Studio 2010
#include <iostream>
#include <string>
#include <time.h>
using namespace std;

class Hitter
{
protected:
    string name;
    int no_of_at_bats; // 타석 수
    int no_of_hittings; // 타수
    int no_of_hits; // 안타수
    int no_of_homeruns; // 홈런 수
    int no_of_bbs; // 사사구 수

public:
    string get_name();
    int get_no_of_at_bats();
    int get_no_of_hittings();
    int get_no_of_hits();
    int get_no_of_homeruns();
    int get_no_of_bbs();

    void set_name();
    void init_stat();

    void add_homerun();
    void add_hit();
    void add_bb();
    void add_out();

    double batting_average(); // 타율
    double on_base_average(); // 출루율

    void show_stat();
};

```

```
string Hitter::get_name()
{
    return name;
}

int Hitter::get_no_of_at_bats()
{
    return no_of_at_bats;
}

int Hitter::get_no_of_hittings()
{
    return no_of_hittings;
}

int Hitter::get_no_of_hits()
{
    return no_of_hits;
}

int Hitter::get_no_of_homeruns()
{
    return no_of_homeruns;
}

int Hitter::get_no_of_bbs()
{
    return no_of_bbs;
}

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

```
void Hitter::init_stat()
{
    no_of_at_bats = 0;
    no_of_hittings = 0;
    no_of_hits = 0;
    no_of_homeruns = 0;
    no_of_bbs = 0;
}

void Hitter::add_homerun()
{
    no_of_homeruns++;
    no_of_hits++;
    no_of_hittings++;
    no_of_at_bats++;
}

void Hitter::add_hit()
{
    no_of_hits++;
    no_of_hittings++;
    no_of_at_bats++;
}

void Hitter::add_bb()
{
    no_of_bbs++;
    no_of_at_bats++;
}

void Hitter::add_out()
{
    no_of_hittings++;
    no_of_at_bats++;
}

double Hitter::batting_average()
```

```

{
    if (no_of_hittings > 0)
        return ((double) no_of_hits / (double) no_of_hittings);
    else
        return (0.);
}

double Hitter::on_base_average()
{
    if (no_of_at_bats > 0)
        return ((double)(no_of_hits + no_of_bbs) / (double)
no_of_at_bats);
    else
        return (0.);
}

void Hitter::show_stat()
{
    cout << "Name          : " << name << endl << endl;
    cout << "No of At-Bats   : " << no_of_at_bats << endl;
    cout << "No of Hittings  : " << no_of_hittings << endl;
    cout << "No of Hits      : " << no_of_hits << endl;
    cout << "No of Homeruns  : " << no_of_homeruns << endl;
    cout << "No of BBs       : " << no_of_bbs << endl;
    cout << "Batting Average : " << batting_average() << endl;
    cout << "On-Base Average : " << on_base_average() << endl << endl;
}

void show_homerun_top(Hitter plist[], int num_of_players)
{
    int max_homeruns = 0;
    int player_index = -1;
    int temp;

    for(int i = 0; i < num_of_players; i++) {

```

```

        if ((temp = plist[i].get_no_of_homeruns()) > max_homeruns) {
            max_homeruns = temp;
            player_index = i;
        }
    }

    cout << "홈런왕 성적" << endl;

    plist[player_index].show_stat();
}

```

```

int _tmain(int argc, _TCHAR* argv[])    // Visual Studio 2010
{
    int no_of_players;

    cout << "선수 수는 ? ";
    cin >> no_of_players;

    Hitter *player_list = new Hitter[no_of_players];
    time_t current;

    for(int i = 0; i < no_of_players; i++) {
        player_list[i].set_name();
        player_list[i].init_stat();
    }

    srand((unsigned int) time(&current));

    for (int i = 0; i < no_of_players; i++) {
        while (player_list[i].get_no_of_at_bats() < 100) {
            switch (rand() % 3) {
                case 0:
                    if (rand() % 10)
                        player_list[i].add_hit();
                    else

```

```
                player_list[i].add_homerun();
                break;
            case 1:
                player_list[i].add_bb();
                break;
            case 2:
                player_list[i].add_out();
                break;
        }
    }

    for(int i = 0; i < no_of_players; i++) {
        player_list[i].show_stat();
    }

    show_homerun_top(player_list, no_of_players);

    delete[](player_list);

    return 0;
}
```

```
#include "stdafx.h"           // Visual Studio 2010
#include <iostream>
#include <time.h>

#include "mid_even.h"

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_wins()
{
    no_of_wins++;
}
```

```
void Pitcher::add_lose()
{
    no_of_loses++;
}
```

```
void Pitcher::add_save()
{
    no_of_saves++;
}
```

```
double Pitcher::era()
{
    if (no_of_inning > 0)
        return (double) no_of_res_opp_runs * 9.0 / no_of_inning;
    else
        return 0.;
}
```

```
double Pitcher::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 << "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                  : " << era() << endl;
    cout << "Winning Rate         : " << winning_rate() << endl;
}

int _tmain(int argc, _TCHAR* argv[])    // Visual Studio 2010
{
    Pitcher player1;

    time_t current;

    player1.set_name();
    player1.init_stat();

    srand((unsigned int) time(&current));

    while (player1.get_no_of_inning() < 100.) {
        int rnd_number = rand() % 100;

        if (rnd_number > 75) {
            player1.add_opp_hits();

            if (rnd_number > 98)
                player1.add_lose();
        }
    }
}

```

```
        else if (rnd_number > 97)
            player1.add_res_opp_runs();
        else if (rnd_number > 95)
            player1.add_opp_runs();
    }
    else {
        player1.add_out();

        if (rnd_number < 2)
            player1.add_wins();

        else if (rnd_number < 3)
            player1.add_save();
    }
}
player1.show_stat();

return 0;
}
```