

Data Structure Project #1

출제 날짜: 2019년 9월 11일

Due data: 2019 년 10월 9일 수요일 23:59:59 까지

본 프로젝트에서는 Binary Search Tree(BST) 와 Priority Queue를 이용하여 **축구 구단 관리 프로그램**을 구현한다.

당신은 축구 역사상 가장 위대한 클럽 맨체스터 유나이티드 FC의 감독이다. 현재 맨체스터 유나이티드 FC는 선수단의 태업으로 인하여 강등위기에 놓여있다. 맨체스터 유나이티드 FC의 구단주 이천수르는 이에 분노하여 모든 선수를 방출 시키고 감독인 당신에게 돈을 쥐여주며 팀을 새로 꾸릴 것을 명령하였고 저명한 축구선수들의 정보를 담은 자료(ShootForLog.txt)를 전달했다. 당신은 해당 자료(ShootForLog.txt)와 함께 구단주에게 받은 돈으로 가장 강력한 팀을 꾸릴 수 있는 방법을 프로그래밍을 통해 개발하고자 한다.

다음 장에서 ShootForLog.txt 에 저장된 데이터와 가장 강력한 팀에 대한 정의, 본 프로젝트의 프로그래밍 요구사항에 대해 자세히 설명한다.

순위	구단	구단가치(달러)
1	레알 마드리드 	36억 5000만
2	바르셀로나 	35억 5000만
3	맨체스터 유나이티드 	33억 2000만
4	바이에른 뮌헨 	26억 8000만
5	아스날 	20억 100만

그림 1. 포브스 선정 축구 구단 가치 순위

Programming Implementation

ShootForLog.txt 에 저장된 선수들에 대한 정보는 각 선수의 이름, 포지션 (Position), 이적료(원), 능력치로 구성 되어있다.

- 이름은 영문으로 저장되어 있다.
- 포지션의 종류는 Forward(공격수), Midfielder(미드필더), Defender (수비수), Goalkeeper(골키퍼) 로 총 4개의 값을 가질 수 있다.
- 이적료는 정수 값으로 저장되어 있다.
 - 같은 포지션내에서 이적료가 서로 같은 선수는 존재하지 않는다.
 - 이적료가 높다고 능력치가 높은 것은 아니다.
 - 단위는 (억) 이다.
- 능력치는 정수 값(1~99)으로 저장되어 있다.
 - 같은 포지션내에서 능력치가 서로 같은 선수는 존재하지 않는다.
 - 능력치가 높다고 이적료가 높은 것은 아니다.

```
Messi, Forward, 1300, 99
Suarez, Forward, 1250, 95
Ronaldo, Forward, 250, 63
Jeong Sung-Ryong, Goalkeeper, 845, 54
Seung-woo, Forward, 800, 96
Pogba, Midfielder, 2581, 98
Iniesta, Midfielder, 845, 97
Modric, Midfielder, 867, 93
Fabregas, Midfielder, 721, 86
Jisung, Forward, 1000, 97
```

그림 2. ShootForLog.txt 에 저장된 데이터 형식 및 양식 예시

가장 강력한 팀

본 프로젝트에서 요구하는 **가장 강력한 팀의 정의**는 아래의 조건을 충족해야 한다.

- 팀은 4명의 선수로 구성된다.
 - 4명의 선수는 공격수 1명, 미드필더 1명, 수비수 1명, 골키퍼 1명으로 구성되어야 한다.
- 4명의 선수의 이적료 합이 구단주가 당신에게 준 돈의 액수보다 같거나 작아야 한다.
- 선별한 4명의 선수의 능력치 합(=강력함으로 정의)이 구단주가 당신에게 준 돈의 액수로 구매할 수 있는 4명의 선수로 구성된 팀 중 가장 높아야 한다.
 - 능력치 합이 같은 2개 이상의 팀이 생기는 경우 각 팀의 선수들의 이적료의 합이 더욱 적은 팀이 더욱 강력한 팀이다.
 - 능력치 합이 같을 뿐 아니라 이적료의 합 또한 같은 2 개 이상의 팀의 조합이 있는 경우 해당 팀들의 강력함은 같은 것으로 보고 두 개의 팀 모두 정답으로 인정한다. (둘 중 하나의 팀만 출력해도 정답)

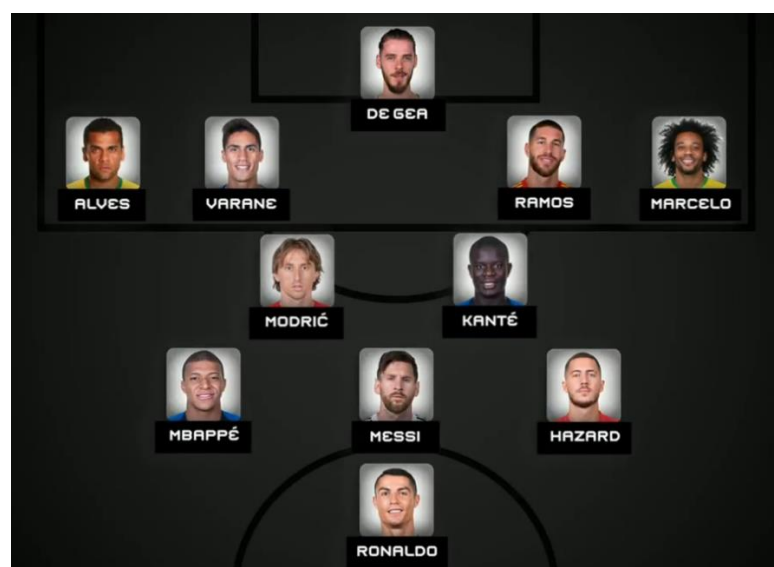


그림 3. 2018 월드 베스트 11

Programming requirements

프로젝트의 전체적인 작업은 크게 3 단계의 동작으로 이루어 진다.

- Stage1. Setup
- Stage2. Print Players
- Stage3. Search the Best Team & Delete Players of Best team from the fwBST, mfBST, dfBST, and gkBST

프로그램 실행 시 두 개의 입력(Argument)이 존재한다. 하나는 ShootForLog.txt 이며 나머지 하나는 구단주에게 받은 돈의 액수(억원)이다. 구단주에게 받은 돈의 액수는 **최소 1664억 이상**이다.

실행 예제)

linux

./실행파일 ShootForLog.txt 돈의 액수(억원)

1700억원을 받은 경우

./실행파일 ShootForLog.txt 1700

4000억원을 받은 경우

./실행파일 ShootForLog.txt 4000

Stage 1. Setup

먼저, 프로그램 실행 시 ShootForLog.txt 를 읽고 선수들의 정보를 포지션 (Forward, Midfielder, Defender, Goalkeeper)에 따라 4개의 Binary Search Tree(fwBST, mfBST, dfBST, gkBST)에 저장하도록 한다. 이 때 Binary Search Tree의 서브 트리를 생성하는 기준은 '능력치' 이다. 삽입 시, 삽입하고자 하는 노드가 루트 노드보다 능력치가 작은 경우 왼쪽 서브 트리에 노드를 삽입하고 큰 경우 오른쪽 서브 트리에 노드를 삽입하도록 한다. 아래의 그림은 위 과정을 도식화한 것이다.

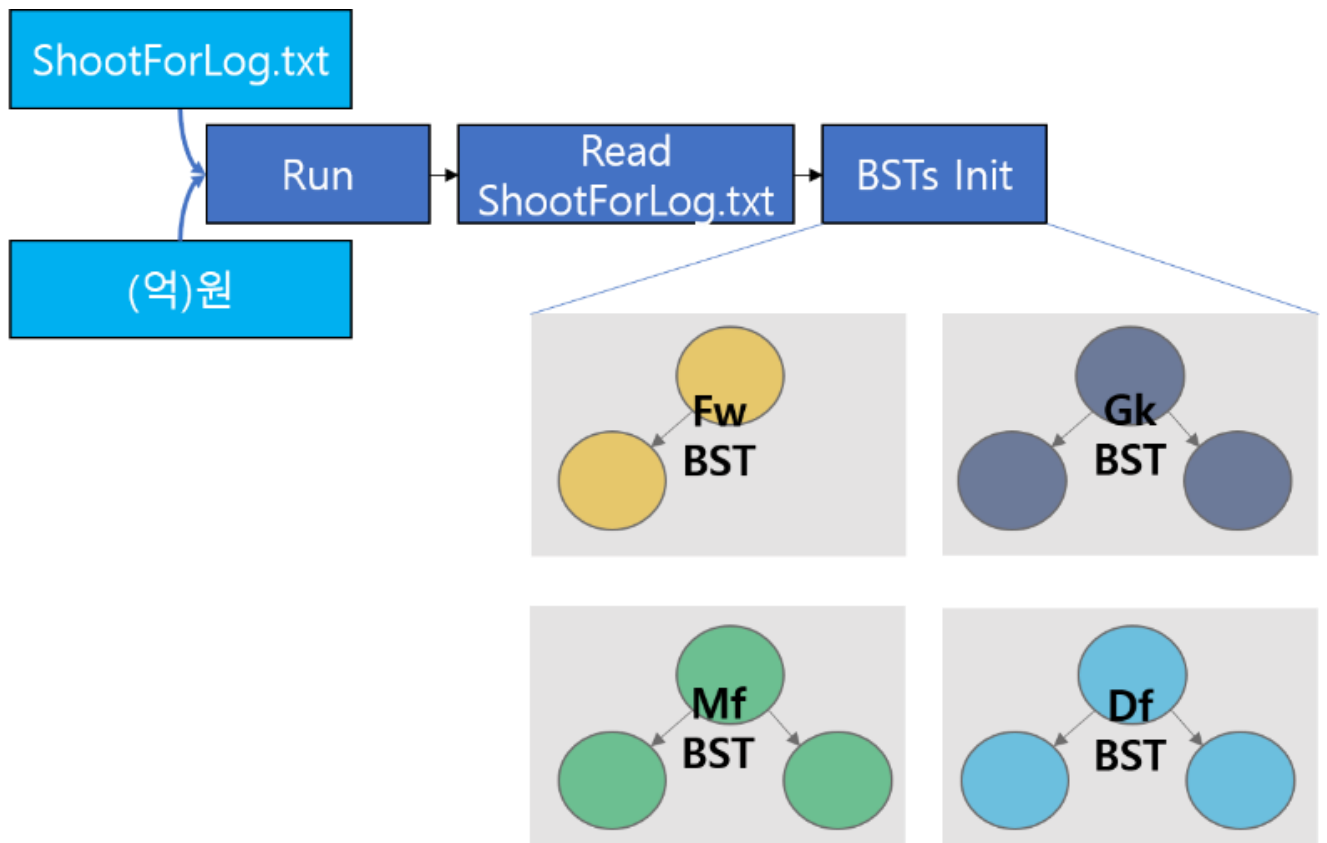


그림 4. Setup 단계의 동작 도식화(BST의 형태는 실제 구현과 다를 수 있음)

Stage 2. Print Players

각 포지션 별로 BST에 데이터를 저장 후, 이를 중위(inorder)순회 하며 선수 목록을 출력한다. 포지션 별로 출력하는 순서는 공격수, 미드필더, 수비수, 골키퍼 순이다.

```
*****Forward List*****
(node.m_name: Ronaldo), (node.m_position: Forward), (node.m_transfer_fee: 250), (node.m_ability: 63)
(node.m_name: Sterling), (node.m_position: Forward), (node.m_transfer_fee: 675), (node.m_ability: 85)
(node.m_name: Mbappe), (node.m_position: Forward), (node.m_transfer_fee: 2475), (node.m_ability: 87)
(node.m_name: Griezmann), (node.m_position: Forward), (node.m_transfer_fee: 809), (node.m_ability: 88)
(node.m_name: Kane), (node.m_position: Forward), (node.m_transfer_fee: 844), (node.m_ability: 89)
(node.m_name: Neymar), (node.m_position: Forward), (node.m_transfer_fee: 1119), (node.m_ability: 90)
(node.m_name: Lewandowski), (node.m_position: Forward), (node.m_transfer_fee: 950), (node.m_ability: 91)
(node.m_name: Salah), (node.m_position: Forward), (node.m_transfer_fee: 960), (node.m_ability: 92)
(node.m_name: Hazard), (node.m_position: Forward), (node.m_transfer_fee: 677), (node.m_ability: 93)
(node.m_name: Dybala), (node.m_position: Forward), (node.m_transfer_fee: 870), (node.m_ability: 94)
(node.m_name: Suarez), (node.m_position: Forward), (node.m_transfer_fee: 1250), (node.m_ability: 95)
(node.m_name: Seung-woo), (node.m_position: Forward), (node.m_transfer_fee: 800), (node.m_ability: 96)
(node.m_name: Jisung), (node.m_position: Forward), (node.m_transfer_fee: 1000), (node.m_ability: 97)
(node.m_name: Son), (node.m_position: Forward), (node.m_transfer_fee: 1200), (node.m_ability: 98)
(node.m_name: Messi), (node.m_position: Forward), (node.m_transfer_fee: 1300), (node.m_ability: 99)
*****
*****Midfielder List*****
(node.m_name: Vidal), (node.m_position: Midfielder), (node.m_transfer_fee: 389), (node.m_ability: 75)
(node.m_name: Robben), (node.m_position: Midfielder), (node.m_transfer_fee: 589), (node.m_ability: 85)
(node.m_name: Fabregas), (node.m_position: Midfielder), (node.m_transfer_fee: 721), (node.m_ability: 86)
(node.m_name: Toure), (node.m_position: Midfielder), (node.m_transfer_fee: 1011), (node.m_ability: 87)
(node.m_name: Eriksen), (node.m_position: Midfielder), (node.m_transfer_fee: 947), (node.m_ability: 88)
(node.m_name: Schweinsteiger), (node.m_position: Midfielder), (node.m_transfer_fee: 713), (node.m_ability: 91)
(node.m_name: Silva), (node.m_position: Midfielder), (node.m_transfer_fee: 342), (node.m_ability: 92)
(node.m_name: Modric), (node.m_position: Midfielder), (node.m_transfer_fee: 867), (node.m_ability: 93)
(node.m_name: Kroos), (node.m_position: Midfielder), (node.m_transfer_fee: 762), (node.m_ability: 94)
(node.m_name: Busquets), (node.m_position: Midfielder), (node.m_transfer_fee: 321), (node.m_ability: 95)
(node.m_name: Zidane), (node.m_position: Midfielder), (node.m_transfer_fee: 1115), (node.m_ability: 96)
(node.m_name: Iniesta), (node.m_position: Midfielder), (node.m_transfer_fee: 845), (node.m_ability: 97)
(node.m_name: Pogba), (node.m_position: Midfielder), (node.m_transfer_fee: 2581), (node.m_ability: 98)
(node.m_name: Kang in), (node.m_position: Midfielder), (node.m_transfer_fee: 900), (node.m_ability: 99)
*****
*****Defender List*****
(node.m_name: Luiz), (node.m_position: Defender), (node.m_transfer_fee: 741), (node.m_ability: 65)
(node.m_name: Maicon), (node.m_position: Defender), (node.m_transfer_fee: 777), (node.m_ability: 77)
(node.m_name: Neville), (node.m_position: Defender), (node.m_transfer_fee: 1212), (node.m_ability: 80)
(node.m_name: Pique), (node.m_position: Defender), (node.m_transfer_fee: 813), (node.m_ability: 81)
(node.m_name: Zanetti), (node.m_position: Defender), (node.m_transfer_fee: 819), (node.m_ability: 85)
(node.m_name: Lahm), (node.m_position: Defender), (node.m_transfer_fee: 817), (node.m_ability: 88)
(node.m_name: Carlos), (node.m_position: Defender), (node.m_transfer_fee: 999), (node.m_ability: 89)
(node.m_name: Vidic), (node.m_position: Defender), (node.m_transfer_fee: 349), (node.m_ability: 92)
(node.m_name: Ramos), (node.m_position: Defender), (node.m_transfer_fee: 913), (node.m_ability: 93)
(node.m_name: Maldini), (node.m_position: Defender), (node.m_transfer_fee: 498), (node.m_ability: 94)
(node.m_name: Yeonggwon), (node.m_position: Defender), (node.m_transfer_fee: 1218), (node.m_ability: 95)
(node.m_name: Nesta), (node.m_position: Defender), (node.m_transfer_fee: 1050), (node.m_ability: 96)
(node.m_name: Puyol), (node.m_position: Defender), (node.m_transfer_fee: 924), (node.m_ability: 97)
(node.m_name: Alves), (node.m_position: Defender), (node.m_transfer_fee: 247), (node.m_ability: 98)
(node.m_name: van Dijk), (node.m_position: Defender), (node.m_transfer_fee: 1450), (node.m_ability: 99)
*****
*****Goalkeeper List*****
(node.m_name: Jeong Sung-Ryong), (node.m_position: Goalkeeper), (node.m_transfer_fee: 846), (node.m_ability: 54)
*****
```

그림 5. 포지션별 BST의 중위순회 출력 결과

Stage 3. Search the Best Team & Delete Players of Best team from the fwBST, mfBST, dfBST, and gkBST

각각의 포지션 별로 선수 한 명을 선별하며 현재 보유한 재산으로 구매 가능한 가장 강력한 팀을 탐색한다. 가장 강력한 팀의 탐색이 끝나면 선수 각각의 포지션 별로 가장 강력한 팀의 존재하는 선수들의 정보를 제거한다. 이 후, 가장 강력한 팀을 출력하고 포지션 별로 선수 목록(가장 강력한 팀의 존재하는 선수들의 정보가 제거 돼있는)을 출력한다.

BST 로부터 선수를 제거할 때, 제거될 노드가 Left 서브트리와 Right 서브트리를 갖는 경우, Left 서브트리 내에서 가장 큰 능력치를 갖는 노드를 찾고 해당 노드를 제거될 노드에 위치시키도록 한다.

다음 장의 그림 6. 은 구단주에게 4,400 억을 받았을 때 구매 가능한 가장 강력한 팀의 출력결과와 해당 선수들을 포지션별 BST 로부터 제거한 결과이다.

```

Best Players
(node.m_name: Messi), (node.m_position: Forward), (node.m_transfer_fee: 1300), (node.m_ability: 99)
(node.m_name: Kang in), (node.m_position: Midfielder), (node.m_transfer_fee: 900), (node.m_ability: 99)
(node.m_name: Alves), (node.m_position: Defender), (node.m_transfer_fee: 247), (node.m_ability: 98)
(node.m_name: Jeong Sung-Ryong), (node.m_position: Goalkeeper), (node.m_transfer_fee: 846), (node.m_ability: 54)
sum_transfer_fee 3293
sum_ability 350

-----
The Transfer window close
*****Forward List*****
(node.m_name: Sterling), (node.m_position: Forward), (node.m_transfer_fee: 675), (node.m_ability: 85)
(node.m_name: Mbappe), (node.m_position: Forward), (node.m_transfer_fee: 2475), (node.m_ability: 87)
(node.m_name: Griezmann), (node.m_position: Forward), (node.m_transfer_fee: 809), (node.m_ability: 88)
(node.m_name: Kane), (node.m_position: Forward), (node.m_transfer_fee: 844), (node.m_ability: 89)
(node.m_name: Neymar), (node.m_position: Forward), (node.m_transfer_fee: 1119), (node.m_ability: 90)
(node.m_name: Lewandowski), (node.m_position: Forward), (node.m_transfer_fee: 950), (node.m_ability: 91)
(node.m_name: Salah), (node.m_position: Forward), (node.m_transfer_fee: 960), (node.m_ability: 92)
(node.m_name: Hazard), (node.m_position: Forward), (node.m_transfer_fee: 677), (node.m_ability: 93)
(node.m_name: Dybala), (node.m_position: Forward), (node.m_transfer_fee: 870), (node.m_ability: 94)
(node.m_name: Suarez), (node.m_position: Forward), (node.m_transfer_fee: 1250), (node.m_ability: 95)
(node.m_name: Seung-woo), (node.m_position: Forward), (node.m_transfer_fee: 800), (node.m_ability: 96)
(node.m_name: Jisung), (node.m_position: Forward), (node.m_transfer_fee: 1000), (node.m_ability: 97)
(node.m_name: Son), (node.m_position: Forward), (node.m_transfer_fee: 1200), (node.m_ability: 98)
*****
*****Midfielder List*****
(node.m_name: Vidal), (node.m_position: Midfielder), (node.m_transfer_fee: 389), (node.m_ability: 75)
(node.m_name: Robben), (node.m_position: Midfielder), (node.m_transfer_fee: 589), (node.m_ability: 85)
(node.m_name: Fabregas), (node.m_position: Midfielder), (node.m_transfer_fee: 721), (node.m_ability: 86)
(node.m_name: Toure), (node.m_position: Midfielder), (node.m_transfer_fee: 1011), (node.m_ability: 87)
(node.m_name: Eriksen), (node.m_position: Midfielder), (node.m_transfer_fee: 947), (node.m_ability: 88)
(node.m_name: Schweinsteiger), (node.m_position: Midfielder), (node.m_transfer_fee: 713), (node.m_ability: 91)
(node.m_name: Silva), (node.m_position: Midfielder), (node.m_transfer_fee: 342), (node.m_ability: 92)
(node.m_name: Modric), (node.m_position: Midfielder), (node.m_transfer_fee: 867), (node.m_ability: 93)
(node.m_name: Kroos), (node.m_position: Midfielder), (node.m_transfer_fee: 762), (node.m_ability: 94)
(node.m_name: Busquets), (node.m_position: Midfielder), (node.m_transfer_fee: 321), (node.m_ability: 95)
(node.m_name: Zidane), (node.m_position: Midfielder), (node.m_transfer_fee: 1115), (node.m_ability: 96)
(node.m_name: Iniesta), (node.m_position: Midfielder), (node.m_transfer_fee: 845), (node.m_ability: 97)
(node.m_name: Pogba), (node.m_position: Midfielder), (node.m_transfer_fee: 2581), (node.m_ability: 98)
*****
*****Defender List*****
(node.m_name: Luiz), (node.m_position: Defender), (node.m_transfer_fee: 741), (node.m_ability: 65)
(node.m_name: Maicon), (node.m_position: Defender), (node.m_transfer_fee: 777), (node.m_ability: 77)
(node.m_name: Neville), (node.m_position: Defender), (node.m_transfer_fee: 1212), (node.m_ability: 80)
(node.m_name: Pique), (node.m_position: Defender), (node.m_transfer_fee: 813), (node.m_ability: 81)
(node.m_name: Zanetti), (node.m_position: Defender), (node.m_transfer_fee: 819), (node.m_ability: 85)
(node.m_name: Lahm), (node.m_position: Defender), (node.m_transfer_fee: 817), (node.m_ability: 88)
(node.m_name: Carlos), (node.m_position: Defender), (node.m_transfer_fee: 999), (node.m_ability: 89)
(node.m_name: Vidic), (node.m_position: Defender), (node.m_transfer_fee: 349), (node.m_ability: 92)
(node.m_name: Ramos), (node.m_position: Defender), (node.m_transfer_fee: 913), (node.m_ability: 93)
(node.m_name: Maldini), (node.m_position: Defender), (node.m_transfer_fee: 498), (node.m_ability: 94)
(node.m_name: Yeonggwon), (node.m_position: Defender), (node.m_transfer_fee: 1218), (node.m_ability: 95)
(node.m_name: Nesta), (node.m_position: Defender), (node.m_transfer_fee: 1050), (node.m_ability: 96)
(node.m_name: Puyol), (node.m_position: Defender), (node.m_transfer_fee: 924), (node.m_ability: 97)
(node.m_name: van Dijk), (node.m_position: Defender), (node.m_transfer_fee: 1450), (node.m_ability: 99)
*****
*****Goalkeeper List*****
*****
계속하려면 아무 키나 누르십시오 . . .

```

그림 6. 구단주로부터 4,400억원을 받았을 때 Stage 3. 의 출력 결과

본 프로젝트에서는 프로젝트를 수행하기 위하여 4개의 클래스를 정의하고 이에 대한 스켈레톤 코드를 제공한다.

스켈레톤 코드는 아래의 링크에서 다운로드 할 수 있다.

https://github.com/developer0hye/2019_Kwangwoon_Univ_CE_DS_Project_1

SoccerPlayerData .h & .cpp

- 선수들의 정보(이름, 포지션, 이적료, 능력치)를 저장하기 위한 클래스

TreeNode .h & .cpp

- 선수들의 정보를 Binary Tree 형태로 저장하기 위한 노드 클래스

BinarySearchTree .h & .cpp

- 선수들의 정보를 Binary Search Tree 형태로 저장하기 위한 트리 클래스
- 소멸자에서는 동적할당 된 모든 데이터의 메모리를 해제해야 한다.

TransferWindowManager .h & .cpp

- 선수들의 정보를 전체적으로 관리하고 처리하기 위한 클래스

DS_Project1.cpp

- 프로그램을 실행하기 위한 메인 소스 코드 파일
-
- 스켈레톤 코드에 선언된 클래스의 함수는 모두 정의가 되어야 하고 프로젝트 구현에 있어 이를 사용하여야 한다.
- main.cpp 의 내용은 수정하지 않도록 한다.
- 데이터의 처리를 위하여 각 클래스의 멤버 함수, 멤버 변수 또는 추가적으로 함수를 선언하고 정의할 수 있다. (새로운 클래스는 선언할 수 없다.)
- STL 사용은 자유롭다.
- 프로그램 종료 시 이전에 동적할당 된 모든 데이터의 메모리가 해제되어야 한다.
- 프로그램 종료 시 Segmentation Fault 가 뜨는 경우 코드 점수 총점의 20% 감점
- .cpp 파일에 추가적인 코드 작성 필요 없다고 판단되는 경우 .h 에만 작성해도 문제 없음.

문제 관련하여 질문이 있을 시 아래의 연락처로 연락주세요.

권용혜, 비마관 201호, robotmanyh@naver.com

변주형, 비마관 201호, guswngud3@kw.ac.kr