

프로그래밍 과제 #1 [경마게임]

1. 목적

클래스와 캡슐화를 이용한 객체지향프로그래밍을 연습하고 객체의 동적 할당과 생성자, 소멸자를 만들어본다.

2. 내용

간단한 경마게임을 만든다. 말의 숫자와 플레이어가 몇 번째 말을 선택할지를 물어보고 그 숫자를 입력 받는다. 말이 우승하려면 총 20의 길이만큼 가야하고 1 turn에 0~5 사이의 무작위 숫자만큼 움직인다. 매 turn마다 말의 위치를 출력해준다. 모든 말이 다 20만큼 이동하면 제일 먼저 도착한 말의 숫자를 출력해주고 사용자가 선택한 말과 같을 시 우승 메시지를 출력한다.

3. 동작 예시

[예시1] 정상 입력과 우승시

```
선택 Microsoft Visual Studio 디버그 콘솔
경마게임!
몇 마리의 말로 플레이하시겠습니까?
4
몇 번째 말을 선택하시겠습니까?
3

Time : 1
=====
1| ( )=> 0
2| ( )=> 0
3| ( )=> 5
4| ( )=> 4
=====

Time : 2
=====
1| ( )=> 3
2| ( )=> 0
3| ( )=> 9
4| ( )=> 6
=====

Time : 3
=====
1| ( )=> 4
2| ( )=> 3
3| ( )=> 11
4| ( )=> 6
=====

Time : 4
=====
1| ( )=> 5
2| ( )=> 5
3| ( )=> 16
4| ( )=> 6
=====

Time : 5
=====
1| ( )=> 10
2| ( )=> 9
3| ( )=> 18
4| ( )=> 7
=====

Time : 6
=====
1| ( )=> 13
2| ( )=> 12
3| ( )=> 20
4| ( )=> 9
=====
```

```
선택 Microsoft Visual Studio 디버그 콘솔

Time : 7
=====
1| ( )=> 16
2| ( )=> 15
3| ( )=> 20
4| ( )=> 11
=====

Time : 8
=====
1| ( )=> 20
2| ( )=> 19
3| ( )=> 20
4| ( )=> 15
=====

Time : 9
=====
1| ( )=> 20
2| ( )=> 20
3| ( )=> 20
4| ( )=> 16
=====

Time : 10
=====
1| ( )=> 20
2| ( )=> 20
3| ( )=> 20
4| ( )=> 16
=====

Time : 11
=====
1| ( )=> 20
2| ( )=> 20
3| ( )=> 20
4| ( )=> 16
=====

Time : 12
=====
1| ( )=> 20
2| ( )=> 20
3| ( )=> 20
4| ( )=> 20
=====

우승자는 3번째 말입니다!
축하드립니다! 당신은 승리하셨습니다.
```

[예시2] 잘못된 입력과 패배시

```
Microsoft Visual Studio 디버그 콘솔
경매게임!
몇 마리의 말로 플레이하시겠습니까?
a
잘못된 숫자 입력입니다.
다시 입력해주세요
sfd
잘못된 숫자 입력입니다.
다시 입력해주세요
8
최대 5마리의 말까지 플레이 가능합니다.다시 입력해주세요
0
최소 2마리의 말이 필요합니다.
다시 입력해주세요
4
몇 번째 말을 선택하시겠습니까?
cie
잘못된 숫자 입력입니다.
다시 입력해주세요
5
잘못된 입력입니다. 입력은 1이상 4이하여야 합니다.
다시 입력해주세요
-3
잘못된 입력입니다. 입력은 1이상 4이하여야 합니다.
다시 입력해주세요
2

Time : 1
=====
1|-----()=> 0
2|-----()=> 2
3|-----()=> 5
4|-----()=> 1
=====

Time : 2
=====
1|-----()=> 5
2|-----()=> 7
3|-----()=> 10
4|-----()=> 4
=====

Time : 3
=====
1|-----()=> 5
2|-----()=> 7
3|-----()=> 14
4|-----()=> 8
=====

Time : 4
=====
1|-----()=> 10
2|-----()=> 12
3|-----()=> 16
4|-----()=> 11
=====
```

```
Microsoft Visual Studio 디버그 콘솔

Time : 5
=====
1|-----()=> 15
2|-----()=> 15
3|-----()=> 20
4|-----()=> 16
=====

Time : 6
=====
1|-----()=> 16
2|-----()=> 17
3|-----()=> 20
4|-----()=> 20
=====

Time : 7
=====
1|-----()=> 16
2|-----()=> 20
3|-----()=> 20
4|-----()=> 20
=====

Time : 8
=====
1|-----()=> 17
2|-----()=> 20
3|-----()=> 20
4|-----()=> 20
=====

Time : 9
=====
1|-----()=> 18
2|-----()=> 20
3|-----()=> 20
4|-----()=> 20
=====

Time : 10
=====
1|-----()=> 20
2|-----()=> 20
3|-----()=> 20
4|-----()=> 20
=====

우승자는 3번째 말입니다!
당신은 패배하셨습니다. :(
```

4. 조건

- 반드시 클래스를 이용해(클래스 2개) 객체지향 프로그래밍을 할 것(C스타일로 프로그래밍 시 0점 처리함)
- 말의 수는 최소 2마리, 최대 5마리로 제한한다. 이 범위 외의 값이 입력되면 오류 메시지를 출력하고 올바른 값을 다시 입력 받도록 한다.
- 플레이어가 선택하는 말의 숫자는 1에서 최대 말의 숫자 사이의 수이다. (1이상, n이하) 이

조건에 부합하지 않은 수를 입력 받게 될 시 오류메시지를 출력하고 올바른 값을 다시 입력 받도록 한다.

- 말의 수만큼의 공간을 동적 할당 받게 코드를 작성하고 소멸자를 만들도록 한다.
- 무작위의 숫자를 받기 위해서 `<cstdlib>`와 `<ctime>` 라이브러리를 이용한다. `srand()`와 `rand()` 함수를 사용하고 `srand()`의 seed 값으로 `time(nullptr)` 값을 준다.
- `cin`이 올바르지 못한 type의 값을 입력받는 경우 `cin`의 상태 플래그가 `failbit`로 설정돼서 정상작동을 못하게 된다. 이 경우 `cin.clear()`를 통해 `cin`의 상태플래그를 초기화시켜주고 `cin.ignore()`를 이용해 버퍼에 남은 입력을 비워준다.
- 클래스를 `.h`파일과 `.cpp`파일로 분리하여 작성한다.