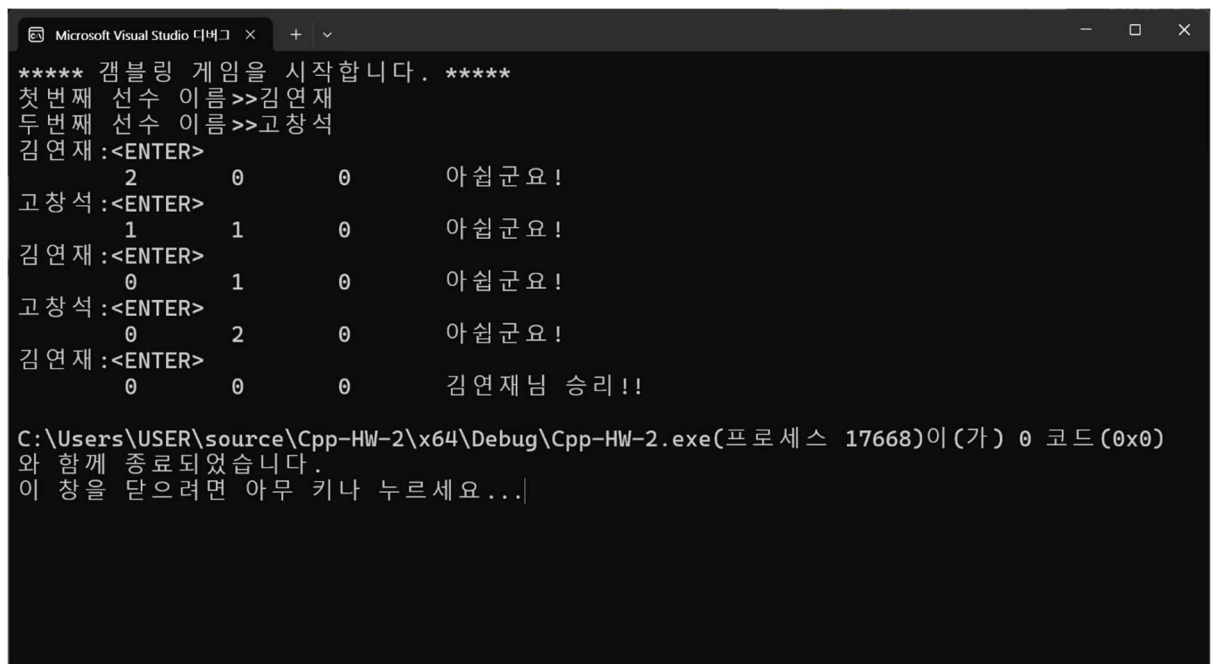


객체지향프로그래밍 과제#2

소프트웨어전공 202284011 김연재

소프트웨어전공 202004006 고창석

- 소스 수행 결과 화면 캡처



```
Microsoft Visual Studio 디버그 콘솔
***** 갬블링 게임을 시작합니다. *****
첫 번째 선수 이름 >> 김연재
두 번째 선수 이름 >> 고창석
김연재 : <ENTER>
      2      0      0      아쉽군요 !
고창석 : <ENTER>
      1      1      0      아쉽군요 !
김연재 : <ENTER>
      0      1      0      아쉽군요 !
고창석 : <ENTER>
      0      2      0      아쉽군요 !
김연재 : <ENTER>
      0      0      0      김연재님 승리 !!

C:\Users\USER\source\Cpp-HW-2\x64\Debug\Cpp-HW-2.exe(프로세스 17668)이(가) 0 코드(0x0)
와 함께 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요 ...|
```

- 소스 구현 설명

- 문제 정의

랜덤 함수를 사용해서 두 플레이어에게 랜덤한 3개의 수를 생성해서 주고, 모두 동일한 수가 나온 사람을 승자로 하는 갬블링 게임을 제작한다.

■ 문제 해결 방법, 문제를 해결한 키 아이디어 또는 알고리즘 설명

```
1  ∨ #include <iostream>
2    | #include <string>
3    | #include <random>
4
5    | using namespace std;
6
7  ∨ class Player {
8    |     int card[3];
9    |     string name;
10   | public:
11   |     Player() :Player("플레이어") { }
12   |     Player(string name) { this->name = name; }
13   |     string getName() { return name; }
14   |     bool playGambling();
15   | };
16  ∨ bool Player::playGambling() {
17  |     random_device rd;
18  |     uniform_int_distribution<int> distribution(0, 100);
19
20  |     ∨ for (int i = 0; i < 3; i++) {
21  |     |     card[i] = distribution(rd) % 3;
22  |     |     cout << "Wt" << card[i];
23  |     | }
24  |     ∨ for (int i = 0; i < 2; i++) {
25  |     |     ∨ if (card[i] != card[i + 1]) {
26  |     |     |     return false;
27  |     |     | }
28  |     | }
29  |     return true;
30  | }
31 }
```

◆ Player 클래스

Player() : 기본 생성자로, 이름을 플레이어로 초기화한다.

Player(String name) : 플레이어의 이름을 사용자가 지정할 수 있는 생성자

string getName : 플레이어의 이름을 반환

bool playerGambling() : 갬블링을 수행하는 부분.

먼저 무작위 숫자 생성기로 0, 1, 2 중에서 3개를 뽑아 card[3] 배열에 저장한다.

책이 없어 문제 3번의 힌트를 참조하지 못해서, 랜덤한 숫자를 위해 웹서핑을 통해 C++11 에서 추가된 random 라이브러리를 사용했다.

```

32  class GamblingGame {
33      Player player[2];
34      bool isGameRun = true;
35  public:
36      GamblingGame();
37      void play() {
38          int i = 0;
39          while (isGameRun) {
40              cout << player[i % 2].getName() << ":<ENTER>";
41
42              char input = cin.get();
43
44              if (input != 'Wn') {
45                  cout << "게임이 강제 종료되었습니다!" << endl;
46                  return;
47              }
48
49              if (player[i % 2].playGambling()) {
50                  isGameRun = false;
51                  cout << "Wt" << player[i % 2].getName() << "님 승리!!" << endl;
52              }
53              else {
54                  cout << "Wt아쉽군요!" << endl;
55              }
56              i++;
57          }
58      }
59  };

```

◆ GamblingGame 클래스

bool isGameRun : 게임이 종료되었는지를 나타내는 플래그. false면 게임을 계속하고, true면 게임이 종료된다.

play() 메서드 : while문을 사용해서 isGameRun이 true일 때, 계속 작동하고, 엔터가 아닌 다른 키를 입력하면 false를 넣어 강제 종료를 시키도록 했다.

문자입력은 char형 변수 input을 get()함수로 입력받아서 if조건문을 이용해 입력받은 변수 input이 엔터가 아니라면 강제 종료 되도록 구현하였다.

숫자가 같지 않으면 false를 return받으므로 일반적으로 else문이 계속 실행되고

숫자가 같으면 true를 return받고 isGameCompleted는 false를 받기때문에 다음 while문은 실행되지 않고 종료되고 승리자를 말하는 구문이 실행된다.

```

61  ▾ GamblingGame::GamblingGame() {
62  |     cout << "***** 갠블링 게임을 시작합니다. *****" << endl;
63  |     string name;
64  |
65  |     while (true) {
66  |         cout << "첫번째 선수 이름>>";
67  |         getline(cin, name);
68  |
69  |         if (name.empty()) {
70  |             cout << "이름은 빈 문자열일 수 없습니다. 다시 입력하세요." << endl;
71  |         }
72  |         else {
73  |             player[0] = Player(name);
74  |             break;
75  |         }
76  |     }
77  |
78  |     while (true) {
79  |         cout << "두번째 선수 이름>>";
80  |         getline(cin, name);
81  |
82  |         if (name.empty()) {
83  |             cout << "이름은 빈 문자열일 수 없습니다. 다시 입력하세요." << endl;
84  |         }
85  |         else {
86  |             player[1] = Player(name);
87  |             break;
88  |         }
89  |     }
90  | }

```

GamblingGame() 생성자 : 게임을 시작할 때 호출된다. 두 명의 플레이어 이름을 입력받고, 빈 문자열을 입력하면 오류를 출력한다.

```

92  ▾ int main() {
93  |     GamblingGame game;
94  |     game.play();
95  |
96  |     return 0;
97  | }

```

main() 함수 : GamblingGame 클래스의 객체 game을 만들고 이 객체로 play() 메서드를 실행하는 코드이다.

■ 아이디어 평가

김연재 - 플레이어 이름에 공백이 들어갔을 때 오류를 출력하도록 만들었는데, 예외 처리를 넣어서 안정성을 올린 점이 좋았다.

고창석 - play() 메서드에서 엔터키가 아닌 다른 키를 입력했을 때 즉시 종료되도록 케이스를 구분하고 싶었는데 이를 해결할 방법을 찾지 못해서 다른 문자를 입력받았을 때 종료되도록 예외 케이스를 구성했다. 좀 더 연습해서 코드를 개선하고 싶다.