I.문제 분석 및 정의

명품 C++ 프로그래밍 278페이지 14번 문제는 두 사람이 참여하는 겜블링 게임을 구현하는 것입니다. 게임의 목표는 각 플레이어가 자신의 차례에 <Enter> 키를 눌러 0 이상 2 이하의 무작위 숫자 3개를 생성하고, 세 숫자가 모두 동일한 조합을 얻으면 승리하는 것입니다.

II.요구사항

1. 두 명의 플레이어가 게임에 참여합니다.

2. 각 플레이어의 이름을 초기에 입력받습니다.

3. 각 플레이어는 자신의 차례에 <Enter> 키를 눌러 무작위 숫자 3개를 생성합니다.

4. 세 숫자가 모두 동일한 조합을 얻으면 해당 플레이어가 승리합니다.

5. 플레이어는 `Player` 클래스로 작성하고, 2명의 플레이어는 배열로 구성합니다.

6. 게임은 `GamblingGame` 클래스로 작성합니다.

III.설계

1.클래스 다이어그램

+------------------+

| Player |

+------------------+

| - name: string |

+------------------+

| + Player(name) |

+------------------+

+---------------------------+

| GamblingGame |

+---------------------------+

| - players: Player[2] |

+---------------------------+

| + GamblingGame(p1, p2) |

| + play() |

+---------------------------+

2. 파일 구조

- Player.hpp

- Player.cpp

- GamblingGame.hpp

- GamblingGame.cpp

- main.cpp

3. 구현

○ Player.hpp

#ifndef PLAYER\_HPP

#define PLAYER\_HPP

#include <string>

#include <iostream>

// Player 클래스 선언

class Player {

public:

std::string name; // 플레이어의 이름을 저장하는 멤버 변수

Player(const std::string& name); // 생성자 선언

};

#endif // PLAYER\_HPP

```

\*\*설명\*\*: `Player.hpp` 파일은 `Player` 클래스의 선언을 포함합니다. 이 클래스는 각 플레이어의 이름을 저장하는 역할을 합니다. `Player` 클래스는 생성자를 통해 이름을 초기화합니다.

○ Player.cpp

#include "Player.hpp"

// Player 클래스의 생성자 구현

Player::Player(const std::string& name) : name(name) {}

```

\*\*설명\*\*: `Player.cpp` 파일은 `Player` 클래스의 구현을 포함합니다. 생성자는 전달된 이름을 멤버 변수 `name`에 저장합니다.

○ GamblingGame.hpp

#ifndef GAMBLINGGAME\_HPP

#define GAMBLINGGAME\_HPP

#include "Player.hpp"

#include <cstdlib>

#include <ctime>

// GamblingGame 클래스 선언

class GamblingGame {

private:

Player players[2]; // 두 명의 플레이어를 저장하는 배열

public:

GamblingGame(const std::string& player1, const std::string& player2); // 생성자 선언

void play(); // 게임을 진행하는 메서드 선언

};

#endif // GAMBLINGGAME\_HPP

```

\*\*설명\*\*: `GamblingGame.hpp` 파일은 `GamblingGame` 클래스의 선언을 포함합니다. 이 클래스는 두 명의 플레이어를 저장하고, 게임을 진행하는 역할을 합니다. `GamblingGame` 클래스는 생성자를 통해 두 플레이어의 이름을 받아 초기화합니다.

○ GamblingGame.cpp

#include "GamblingGame.hpp"

#include <iostream>

// GamblingGame 클래스의 생성자 구현

GamblingGame::GamblingGame(const std::string& player1, const std::string& player2)

: players{ Player(player1), Player(player2) } {

std::srand(std::time(0)); // 난수 생성을 위한 시드 설정

}

// 게임을 진행하는 메서드 구현

void GamblingGame::play() {

char c;

while (true) {

for (int i = 0; i < 2; ++i) {

std::cout << players[i].name << ">> : ";

std::cin.get(c); // 플레이어의 입력을 기다림

if (c == '\n') { // <Enter> 키가 눌렸을 때

std::cout << "<Enter>" << std::endl;

int numbers[3];

for (int j = 0; j < 3; ++j) {

numbers[j] = std::rand() % 3; // 0부터 2까지의 난수 생성

std::cout << numbers[j] << " ";

}

std::cout << std::endl;

if (numbers[0] == numbers[1] && numbers[1] == numbers[2]) { // 세 숫자가 모두 동일한 경우

std::cout << players[i].name << " wins!" << std::endl;

return; // 게임 종료

}

}

}

}

}

```

\*\*설명\*\*: `GamblingGame.cpp` 파일은 `GamblingGame` 클래스의 구현을 포함합니다. 생성자는 두 플레이어의 이름을 받아 `Player` 객체를 초기화하고, 난수 생성을 위한 시드를 설정합니다. `play` 메서드는 게임을 진행하며, 각 플레이어가 <Enter> 키를 눌러 무작위 숫자를 생성하고, 세 숫자가 모두 동일한 경우 승리 메시지를 출력합니다.

○ main.cpp

#include "GamblingGame.hpp"

#include <iostream>

int main() {

std::string player1, player2;

std::cout << "Enter name for player 1: ";

std::cin >> player1; // 첫 번째 플레이어의 이름 입력

std::cout << "Enter name for player 2: ";

std::cin >> player2; // 두 번째 플레이어의 이름 입력

GamblingGame game(player1, player2); // 게임 객체 생성

game.play(); // 게임 시작

return 0;

}

\*\*설명\*\*: `main.cpp` 파일은 프로그램의 진입점입니다. 두 플레이어의 이름을 입력받고, `GamblingGame` 객체를 생성하여 게임을 시작합니다.

IV.테스트 계획

1. 프로그램을 컴파일하고 실행합니다.

2. 두 플레이어의 이름을 입력합니다.

3. 각 플레이어가 번갈아 가며 <Enter> 키를 눌러 무작위 숫자를 생성합니다.

4. 세 숫자가 모두 동일한 조합이 나올 때까지 게임을 진행합니다.

5. 승리 메시지가 출력되는지 확인합니다.