

문제 정의

두 명의 플레이어가 참여할 수 있는 겜블링 게임을 구현해야 합니다. 먼저 두 명의 플레이어 이름을 설정해 <Enter>키를 누르면 플레이어가 번갈아 가며 0~2의 랜덤 숫자를 생성하고, 세 개의 숫자가 동일할 경우 해당 플레이어가 승리하는 구조입니다.

플레이어를 지정할 클래스와 게임 전체를 관리하는 클래스를 생성하고 두 명의 플레이어를 동적으로 배열을 할당해야한다고 생각했습니다. 동적으로 할당하기 위해서는 포인터 변수와 new/delete 연산자를 이용해 동적 메모리 할당 및 해제를 통해 자원을 관리합니다.

문제 해결 방법 및 알고리즘 설명

Gambling.h 헤더파일에서는 문자열을 처리하기 위해 `#include <string>` 표준 라이브러리를 사용했습니다. Player 클래스 선언부에서는 플레이어의 이름을 저장할 `string name` 멤버변수를 생성하였고, Player 생성자를 만들어 객체가 생성될 때 멤버 변수 `name`을 공백으로 초기화를 하였습니다. `setName` 멤버 함수를 통해 플레이어의 이름을 설정합니다. 또한 `getName` 함수를 이용해 플레이어의 이름을 반환하기 위해 함수를 선언했습니다.

GamblingGame 클래스 선언부에서는 동적으로 메모리를 할당시키기 위해 `Player *p`를 작성해 두 명의 Player 객체를 가리키는 포인터를 작성하였습니다.

또한, 플레이어의 이름을 임시로 저장하기 위한 변수를 위해서 `string name;`을 작성합니다 그 다음 플레이어의 이름을 입력받기 위한 `setPlayer`, 랜덤 숫자를 생성하고 승리 여부를 확인하는 함수 `rand_num`, 게임 진행을 관리하는 함수 `game_start` 함수들을 선언했습니다.

다음으로 Gamebling.cpp 파일에서 Player 클래스 구현부는 생성자, 소멸자를 작성하고 void Player :: setName(string name)로 플레이어의 이름을 설정하는 함수를 작성 및 this ->name = name;으로 매개변수로 받은 이름을 현재 플레이어의 이름으로 설정합니다.이 코드에서 this를 사용한 이유는 setName 함수의 매개변수 name과 클래스의 멤버 변수 name이 동일한 이름을 가지고 있습니다. 이 경우, name이라는 이름을 사용할 때 C++는 매개변수 name을 우선적으로 찾기 때문에. 따라서 멤버 변수에 접근하려면 객체 자신을 가리키는 this를 사용해야 합니다.

GamblingGame 클래스는 두 명의 플레이어 객체를 동적으로 생성시키고 매번 다른 난수 생성을 위해 srand((unsigned)time(0));을 작성합니다. 그 후 게임 시작 메시지를 출력합니다.

GamblingGame 생성자에서 두명의 플레이어를 생성할 때 동적으로 배열을 할당하기 위해 new 연산자를 사용하였고, 또한 객체가 소멸될 때 소멸자에서 동적으로 배열을 할당받았기 때문에 delete [] p 연산자를 사용하였습니다.

플레이어 이름 설정 함수 setPlayer 함수에서는 cin 입력값과 p[i].setName(name);을 이용해 플레이어 이름을 입력받았습니다.

랜덤 숫자 생성 및 승리 여부 판단 함수 rand_num에서는 크기가 3인 배열을 선언하고 반복문을 통해 0,1,2 중 랜덤 숫자를 생성하여 출력하고 세 숫자가 같을 경우 조건문을 사용해 승리 메시지를 출력, 일치하지 않을 경우 메시지를 출력하도록 함수를 구현했습니다.

게임 진행 함수인 game_start 함수는 두 명의 플레이가 번갈아가며 난수를 생성하고, 승리 여부를 판정합니다. 문자열 변수 n과 플레이어의 인덱스를 저장할 i를 선언하고 같은 숫자가 나올 때 까지 while(true) 반복문을 사용해 무한루프를 돌렸습니다. p[i % 2].getName() 코드를 사용해 나머지가 0이면 첫번째 플레이어 1이면 두번째 플레이어의 이름을 출력하게 설정하였습니다. rand_num 함수를 호출하여 0~2의 랜덤 숫자를 생성하고 승리 여부를 판단합니다.