

문제 정의 :

갬블링 게임

1. 선수 이름을 초기에 입력을 받는다.
2. 두 플레이어가 번갈아 가면서 Enter 를 칠 때 랜덤의 값을 가진 3 개의 수가 생성된다.
3. 모두 같은 수가 나오면 게임에서 이긴다.
4. 숫자의 범위는 0~2
5. 선수는 Player 클래스로 작성, 2 명의 선수는 배열로 구성
6. 게임은 GamblingGame 클래스로 작성

문제 해결 방법 :

GamblingGame 객체는 main 이 실행될 때 같이 실행되도록 만든다.

GamblingGame 클래스에는 두 플레이어의 이름 변수와, game 메소드를 사용한다.

객체 생성 후에 플레이어의 이름 변수를 cin 해주면서, Player 객체를 생성한다.

GamblingGame 에 game 메소드는 두 플레이어의 순서를 차례로 넘기며 게임에서 이긴 플레이어가 있는지 체크한다.

게임에서 이긴 플레이어가 있다면, Player 객체를 삭제하고 게임을 종료한다.

Player 클래스는 동적 클래스로 객체를 생성할 때는 객체 배열을 사용해 생성한다.

Player 클래스의 생성자는 기본 생성자로 만들고, 게임이 끝날 때 객체를 삭제해 주기 위해 소멸자도 구현해준다.

Player 클래스에는 3 개의 숫자 변수와, 1 개의 결과 bool 변수가 존재한다. 3 개의 숫자를 랜덤으로 생성해주는 메서드와 생성된 숫자를 출력해주는 메서드, 3 개의 숫자를 비교해 결과를 저장하는 메서드와 결과를 출력해주는 메서드 4 가지 메서드가 존재한다.

적절히 this 포인터까지 써볼 수 있으면 좋을 것 같다.

아이디어 평가 :

문제 해결 방법에서 예상했던 부분들에 많은 부분들이 비슷하게 흘러갔다.

다른 부분이 있다면, Player 클래스를 구성할 때 플레이어 명이 추가되었고, 모든 변수들을 private 로 설정했기 때문에 GamblingGame 클래스에서 사용하기 위한 변수들이 많았기에 get, set 함수들을 만들어서 메소드가 생각보다 많이 필요해졌다.

난수 생성과 객체 배열과 포인터에 대해서 사용하기는 했으나 아직 이해도가 많이 부족하다는 생각을 했다.

This 에 관해서는 객체 내부 변수들을 초기화할 때 사용해 주었다.

알고리즘 설명:

GamblingGame.h

GamblingGame 클래스에는 player1, player2, result 변수를 private 로, setPlayer 와 Game 을 public 메서드로 생성한다.

player1, player2 변수는 플레이어명을 넣어주기 위한 변수이다.

Result 변수는 게임의 결과를 넣어주기 위한 변수이다.

setPlayer 는 player 명을 받아서 변수에 초기화해주는 함수이다.

Game 메서드는 게임 시작과 진행 종료의 역할을 하는 메서드이다.

Player.h

Player 클래스에는 playerName, num1, num2, num3, result 변수를 private 로 Player 생성자와 setPlayername, getPlayername, setRandNum, setResult, getResult, outResult 메서드를 생성한다.

playerName 변수는 객체의 플레이어 명을 넣어주기 위한 변수이다.

num1, num2, num3 변수는 0~2 의 랜덤의 값을 넣어주기 위한 변수이다.

Result 변수는 플레이어의 각 라운드의 결과를 넣어주기 위한 변수이다.

Player 생성자는 player 를 인수로 받아서 playerName 을 초기화 해준다.

getPlayername 메서드는 외부에서 playerName 변수에 접근하기 위한 메소드이다.

setRandNum 메소드는 랜덤의 숫자를 num 변수들에 초기화해주는 메소드이다.

setResult 메소드는 초기화된 num 변수들을 비교해 결과를 result 에 초기화해주는 메소드이다.

getResult 메소드는 result 변수를 외부에서 접근하기 위한 메소드이다.

OutResult 메소드는 결과값을 통해서 숫자와 결과값을 cout 로 출력해주는 메소드이다.

Main.cpp

Main 메소드를 실행시켜주는 파일

플레이어 변수 명 2 개를 선언하고, gamblingGame 객체를 생성해준다.

게임 시작과 플레이어명 변수를 초기화해주기 위한 cout 과 cin 을 적어준다.
gamblingGame 객체의 setPlayer 와 Game 메소드를 차례로 실행시켜 준다.

Test.cpp

앞서 소개했던 gamblingGame 와 player 헤더파일에 있는 함수들을 선언해주는 파일이다.

Set 이 앞에 붙어있는 메소드들은 this 를 사용해서 조건이 있다면 조건을 붙여서 변수를 초기화해준다.

Player::setRandNum 메소드는 rand() 메소드를 사용해 랜덤의 숫자를 넣어준다.

Player::setResult 메소드는 num1, num2, num3 의 숫자를 비교하여 결과값을 넣어준다.

Get 이 앞에 붙어있는 메소드들은 조건에 맞는 변수들을 외부에서 사용하기 위해서 알맞은 값으로 return 해준다.

Player::outResult 메소드는 결과값에 따라 다른 값을 출력해준다.

그 외에는 마지막 GamblingGame::Game 메소드가 있다.

이 메소드는 player 객체를 포인터를 사용해 2 개의 값을 갖는 배열의 객체로 생성해준다.

각각 객체들을 Player 생성자로 초기화해준다. 그 값에 각각 cin 으로 받아온 값들을 생성자 인수로 넣어준다. 배열 값을 돌려가며 넣어주기 위한 숫자 변수 num 을 선언해준다.

while 문을 통해 GamblingGame 객체의 result 조건으로 돌린다.

While 문 내에는 num 변수 숫자에 맞는 player 객체 배열의 인덱스를 가진 객체에 대한 player 를 설정해주고, cout, cin 값을 통해 엔터를 치면 결과가 나오도록 변경해준다.

그리고 player 객체의 setRandNum, setResult, outResult 메서드들을 실행시켜 주고, GamblingGame 객체의 result 값을 getResult 을 사용해 초기화해준다. 그 후 num 값을 변경해 준다.

While 문이 끝나면 delete 를 사용해 player 객체 반환해 준다.