

문제 정의:

이 문제는 두 명의 플레이어가 번갈아가며 무작위로 3개의 숫자를 생성하고, 이 숫자들이 모두 동일하게 나오면 그 플레이어가 승리하는 갬블링 게임을 구현하는 것이다. 플레이어는 이름을 입력해 자신의 차례에 Enter 키를 누르고, 0~2 사이의 랜덤한 숫자 3개를 출력하여 그 숫자가 모두 같으면 게임이 종료되고 승자가 결정된다.

문제 해결 방법:

1. 클래스 구조 설계:
 - a. 각 플레이어는 Player 클래스로 정의되며, 이름을 저장하고 자신의 차례에 랜덤한 숫자를 생성하는 역할을 수행한다.
 - b. 게임 전체는 GamblingGame 클래스로 관리되며, 두 명의 플레이어가 번갈아가며 차례를 진행한다.
2. 게임 흐름 제어:
 - a. 두 명의 플레이어가 번갈아 가면서 자신의 차례에 Enter 키를 누르면, 숫자 3개가 생성되고 출력된다.
 - b. 각 차례에서 생성된 숫자들이 모두 동일하면 승리 메시지를 출력하고 게임이 종료된다. 그렇지 않으면 "아쉽군요!" 메시지가 출력되고 다음 플레이어의 차례로 넘어간다.
3. 랜덤 숫자 생성:
 - a. 숫자는 0부터 2까지의 범위에서 무작위로 선택되며, `rand() % 3`을 통해 이를 구현한다.
4. 반복 구조:
 - a. 승자가 나오기 전까지 게임은 무한히 반복되며, 매번 플레이어가 차례를 마치면 턴을 다른 플레이어에게 넘긴다.

아이디어 평가:

- 플레이어 구조:
 - Player 클래스를 사용해 각 플레이어의 이름과 행동을 관리하는 방식을 통해 게임 내 각 플레이어를 개별적으로 처리할 수 있다.
- 게임 반복 흐름:
 - 게임이 진행되는 동안 `while (true)` 루프를 사용해 턴을 계속 돌리는 방식

은 효율적이다. 승리 조건이 충족되면 루프를 빠져나와 게임을 종료한다.

- **랜덤 숫자 생성:**
 - 숫자를 0~2 범위에서 무작위로 생성하는 방식도 간단하면서 문제 정의에 맞다. 승리 조건인 세 개의 숫자가 동일한 경우를 검사하는 것도 직관적이다.
- **입력 흐름 관리:**
 - `cin.get()`을 사용해 사용자가 Enter 키를 입력할 때만 차례가 진행되도록 한 부분도 사용자와의 상호작용을 자연스럽게 구현했다.

문제를 해결한 키 아이디어 또는 알고리즘 설명:

1. **클래스 기반 설계:** Player 클래스는 각 플레이어의 역할을 캡슐화하여 게임 진행 흐름에서 복잡성을 줄인다. 플레이어의 이름과 행동을 별도로 관리하여 프로그램이 확장 가능하고 유지보수에 좋다.
2. **랜덤 숫자 생성 및 승리 조건 검사:** 플레이어가 Enter 키를 누를 때마다 세 개의 랜덤 숫자가 생성된다. `rand() % 3`을 통해 0~2 사이의 숫자를 무작위로 생성하며, 생성된 숫자들을 배열에 저장하고 이를 비교하여 승리 조건을 확인한다. 세 개의 숫자가 동일한지 비교하는 로직이 간결하며, 이를 통해 빠르고 명확하게 승리 여부를 결정할 수 있다.
3. **턴 관리:** `turn` 변수를 통해 현재 차례가 누구인지 관리하며, 차례가 끝나면 $(turn + 1) \% 2$ 방식으로 다음 플레이어에게 차례를 넘긴다. 이 방식을 통해 플레이어 간의 턴이 반복적으로 번갈아 가며 진행되도록 한다.