Open Challenge

한글 끝말잇기 게임

목적 여러 개의 클래스 만들기, 동적 객체 배열 할당 및 반환, 객체 포인터 이용, string 클래스 다루기 연습

n명이 하는 한글 끝말잇기 게임을 작성해보자. 아래의 결과와 같이 선수의 수를 입력받고, 선수 각 사람의 이름을 입력받아 게임을 시작한다. 난이도 6

```
끝말 잇기 게임을 시작합니다
게임에 참가하는 인원은 몇명입니까?3
참가자의 이름을 입력하세요. 빈칸 없이>>황기태
참가자의 이름을 입력하세요. 빈칸 없이>>한원선
참가자의 이름을 입력하세요. 빈칸 없이>>손연재
시작하는 단어는 아버지입니다
황기태>>지우개
한원선>>개나리
손연재>>리본
황기태>>본죽
한원선>>죽집
손연재>>잡수리
```

빈칸 없이 입력

힌트 Hint

(1) 3개의 모듈로 나누어라.

- WordGame 클래스 – 끝말잇기 게임 전체를 운영하는 클래스
- Player 클래스 – 선수를 표현하는 클래스
- main() 함수 – WordGame 객체를 생성하고 게임을 시작하는 함수

(2) 한글 문제

한글 문자열을 저장하기 위해 string 클래스를 이용하라. 한글은 2바이트 코드로 저장되므로, string 객체에 저장된 한글 문자열에서 끝 글자를 비교하려면 두 바이트를 함께 비교하여야 한다. 예를 들면 다음과 같다.

```
string a="아버지"; // 6개의 바이트에 저장. a.size()는 6. 3이 아님
string b="지우개"; // 6개의 바이트에 저장
if(a.at(4) == b.at(0) &&
   a.at(5) == b.at(1)) { // "아버지"의 '지'와 "지우개"의 '지'를 비교
   //끝말잇기 성공한 경우 처리 코드
}
else {
   // 실패한 경우 처리 코드
}
```

P204 한글 끝말잇기 게임 1/10 main.cpp


```
#include <iostream>
#include "WordGame.h"
using namespace std;

int main() {
        WordGame *game;
        game = new WordGame("끝말 잇기", "아버지");
        game->run();
        delete game;
}
```

한글 끝말잇기 게임 2/10 Player.cpp


```
#include <iostream>
#include <string>
using namespace std;

#include "Player.h"

string Player::sayWord() {
        cout << name + ">>";
        cin >> word;
        return word;
}
```

한글 끝말잇기 게임 3/10


```
// lastWord와 참가자가 말한 word를 비교하여 끝말잇기가 잘되었는지 판단.
// 성공하였으면 true 리턴
bool Player::succeed(string lastWord) {
        int lastIndex = lastWord.length()-2; // 최종 단어의 맨 마지막 문자의 인덱스

        // 최종 단어의 맨 마지막 문자와 참가자가 말한 단어의 첫 문자가 같은지 비교
        if(lastWord.at(lastIndex) == word.at(0) &&
                lastWord.at(lastIndex+1) == word.at(1))
                return true;
        else
                return false;
}
```

한글 끝말잇기 게임 4/10 WordGame.cpp


```
#include <iostream>
#include "WordGame.h"
#include "Player.h"
using namespace std;

WordGame::WordGame(string title, string startWord) {
        this->title = title;
        this->startWord = startWord;
        this->players = NULL;
}

WordGame::~WordGame() {
        if(this->players != NULL)
                delete [] this->players;
}
```

한글 끝말잇기 게임 5/10


```
bool WordGame::createPlayers() {
        int n;
        cout << title << " 게임을 시작합니다" << endl;
        cout << "게임에 참가하는 인원은 몇명입니까?";
        cin >> n;

        if(n < 2) {
                cout << "인원수는 2 이상입니다 " << endl;
                return false;
        }
        nPlayers = n;
        players = new Player[n];
        if(!players)
                return false;
```

한글 끝말잇기 게임 6/10


```
        // 각 참여자의 이름을 입력받아 Player 객체 생성
        string playerName;
        for(int i=0; i<nPlayers; i++) {
                cout << "참가자의 이름을 입력하세요. 빈칸 없이>>";
                cin >> playerName;
                players[i].setName(playerName);
        }

        return true;
}
```

한글 끝말잇기 게임 7/10


```
void WordGame::run() {
        if(!createPlayers()) // 참가자 수가 2보다 적4던지 문제가 생긴 경우
                return;

        string lastWord = startWord;
        cout << "시작하는 단어는 "+lastWord+ "입니다" << endl;
        int next =  0; // 참가자들의 순서대로 증가
```

한글 끝말잇기 게임 8/10


```
// 게임이 끝날 때까지 루프
        while(true) {
                string newWord = players[next].sayWord(); // next 참가자가 단어를 말하도록 한다.
                if(!players[next].succeed(lastWord)) { // next 참가자가 성공하였는지 검사.
                        cout << players[next].getName() + "이 졌습니다.";
                        break; // 게임을 종료한다.
                }
                next++; // 다음 참가자
                next %= nPlayers; // next가 참가자의 개수보다 많게 증가하는 것을 막는다.
                lastWord = newWord;
        }
}
```

한글 끝말잇기 게임 9/10 player.h


```
#include <string>
using namespace std;


class Player {
        string name; // 게임 참가자의 이름
        string word; // 참가자가 말한 단어
public:
        void setName(string name) {
                this->name = name;
        }
        string getName() { return name; }
        string sayWord();
        bool succeed(string lastWord);
};
```

한글 끝말잇기 게임 10/10 WordGame.h


```
#include <string>
using namespace std;

class Player;

class WordGame {
private:
        string title;
        string startWord;
        int nPlayers;
        Player *players;
        bool createPlayers();
public:
        WordGame(string title, string startWord);
        void run();
        ~WordGame();
};
```