

2021-1 C++ 프로그래밍 실습과제 06

학번	2020136129	이름	최수연
----	------------	----	-----

(1) 각 문제에 대한 분석과 및 해결 방법

1. 실습과제 3에서 구현한 두 가지 게임 중에서 하나를 골라 클래스 버전으로 수정하라.

- (1) 게임 클래스를 만든다. 이를 위해 6.6절의 내용을 참고하라.
(2) main() 함수를 수정하여 실습과제 3에서와 같이 게임을 할 수 있도록 하라. 이를 위해 게임 객체를 만들고 play() 등의 메소드를 호출하여야 한다.

[문제분석 및 해결방법]

- 클래스명을 SpeedGame(스피드게임)으로 수정한다.(스피드 구구단게임과 스피드 덧셈게임)
- 정적 전역변수 static으로 선언한 NumGames, NumWins, Score는 모두 클래스의 private으로 선언한다.
- 정적함수 static으로 선언한 GuguOnce(), SumOnce()도 private으로 선언하여 개발자만 볼 수 있도록 한다.
- double 함수 playSpeedGugu(), playSpeedSum()은 public으로 하여 사용자에게도 보일 수 있도록 한다.
- 멤버함수 playSpeedGugu()는 Gugu로 playSpeedSum()는 Sum으로 이름을 단순화한다. 스피드게임이라는 것을 클래스명 자체에서 알 수 있으므로 멤버함수의 이름을 단순화하여 본 멤버함수가 구구단이라는 것과 덧셈이라는 것만 구분되도록 하였다.
- 이에, private으로 선언한 멤버함수의 이름도 단순화하였다.
(playGuguOnce()-> GuguOnce(), playSumOnce() -> SumOnce())
- 스피드게임의 메인함수에서 클래스 객체를 선언하였다. (클래스명 객체명; -> SpeedGame a;)
- 메인함수에서 클래스의 public에 있는 멤버함수(Gugu(), Sum())와 멤버변수(tElapsed)를 호출한다. 메인함수에서 클래스를 호출할 때, 반드시 항목연산자 '.'을 사용하여, "객체명.멤버함수(매개변수)"와 "객체명.멤버변수" 형식을 지켜야한다.
- 메인함수에서 매개변수로 받은 nPlay(게임 횟수)와 game(게임의 종류: 스피드구구단게임 or 3~9자리 수 스피드덧셈게임)를 이용하여 게임을 진행하고, score(점수)와 tElapsed(게임소요시간) 반환 후 화면에 출력한다.

2. 교재 291~294쪽의 코드를 참고하여 자신만의 행맨 게임을 구현하라.

- (1) 자신만의 영어 단어를 100개 이상 준비하고, 이를 파일에 저장하라. 예를 들어, 인터넷에서 가장 많이 사용되는 영어 단어 1000개를 찾아 파일에 저장할 수 있다. (예: google에서 "top english word list" 등의 키워드로 검색)

[문제분석 및 해결방법]

Word200.txt 파일을 생성하여 영어 단어 200개를 한 줄당 한 단어씩 저장한다.

- (2) 6.8절의 행맨 프로그램을 수정하여 파일에서 모든 단어를 읽어들이고, 무작위로 임의의 단어가 선택되고 이를 맞추는 게임으로 수정한다.

[문제분석 및 해결방법]

먼저 난수 생성을 위해 헤더파일에 전처리기 <ctime>, <cstdlib>를 선언하고 난수를 초기화한다.

클래스 private에 string을 사용하여 영어단어를 읽어 들일 문자열의 배열 English[200]을 선언한다. 이때, 배

열의 크기는 영어 단어 수에 맞게 200으로 설정한다.

새로운 word() 함수를 클래스 private으로 선언하여 파일 "Word200.txt"에서 문자열을 읽도록 한다.

영단어 문자열을 읽은 후 English에 저장되어있는 문자열 중에서 난수를 생성하여 그 중 하나의 문자열이 선택되면 이를 기존 problem(정답 단어를 보관하는 문자열)에 저장한다.

마지막 play() 함수에서 word()를 한번 호출한다. 이때 play함수의 매개변수 prob를 없애고, 이를 problem (정답 단어)으로 보내는 코드를 삭제한다.

(3) 게임이 진행될 때 마다 출력되는 그림을 자신만의 그림으로 수정하라. 이를 위해 HangmanProgress.txt 파일을 수정하면 된다. 하나씩 틀릴 때 마다 어떤 그림이 나타날 것인가를 먼저 설계하고, 이를 파일에 저장하라. 예를 들어, 아스키 아트를 사용할 수도 있다.

[문제분석 및 해결방법]

"HangmanProgress.txt" 파일을 vs2019로 열어 그림을 수정하였다. 그림의 크기인 80*37 = 2960에 맞게 string으로 progress[2960](각 단계별 화면 출력용 배열)을 선언한다. load() 함수에서 2960줄의 텍스트 파일 문자열을 읽어들인다. print() 함수에서는 각 단계별로 화면을 출력하기 위해 반복문을 통해 행의 크기인 37만큼 반복하여 출력하도록 설정한다.

(2) 자신이 구현한 주요 코드

```
string English[200];
void word(const char* Word = "Word200.txt") {
    ifstream fs(Word);
    if (fs) {
        for (int i = 0; i < 200; i++)
            getline(fs, English[i]);
    }
    int num = rand() % 199;
    problem = English[num];
}

void play() {
    word();
...};
```

먼저 string을 사용하여 단어를 저장할 문자열을 선언한다. "Word200.txt" 파일을 const char*형으로 받아, 읽기 위한 파일을 연다. 파일 내의 한 줄당 한 단어씩의 문자열을 읽어 들이기 위해 for문을 사용해 English[0]부터 English[199]까지 한 단어씩 저장한다. 정수형 변수를 선언하여 0~199까지 총 200개의 숫자 중 하나의 수를 랜덤으로 입력받는다. 입력받은 수를 문자열 English의 배열 크기로 지정한 후, problem에 대입한다.

(3) 다양한 입력에 대한 테스트 결과

1. 실습과제 3에서 구현한 두 가지 게임 중에서 하나를 골라 클래스 버전으로 수정하라.

[문제 1]: $9 \times 3 = 27$
[문제 2]: $8 \times 4 = 32$
[문제 3]: $5 \times 7 = 35$

[문제 1]: $191 + 298 = 489$
 [문제 2]: $134 + 565 = 699$
 [문제 3]: $192 + 904 = 1096$

점수 = 50.8점 (총 7.4초)

점수 = 67.1점 (총 19.8초)

2. 교재 291~294쪽의 코드를 참고하여 자신만의 행맨 게임을 구현하라.

<Hangman Game>

$$\overline{I}$$
[illegible]

until

a.c.ef..ijkl.n.....tu.....
정답!! 휴~ 살았다!

<Hangman Game>

```

      (o
      oo      oooooo      #      oooooooo*
      oooooooooooooo      ooooooooooooooooooooooooooooo
      oooooo      oo      ooooo*      ooooooooooooooooooooooooooooo
      oooooooo      oooooooo      ooooooooooooooooooooooooooooooooooooo
      oooooooooooooooooooooo      %oooooooooooooooooooo#      ooooooooooooooooooooooooooooo
      ooooooooooooooooooooooooooooooooooooooooooooooooooooo      oooooo      ooooo
      #o      ooooooooooooooooooooooooooooooooooooo      /      ooooo      ooooo
      ooooooooooooo*      &oooooooooooooooooooooooooooooooooooo      oo      .....      ooooo
      oo      .....      ooooooooooooooooooooooooooooooooooooo      oo      .....      ooooo
      ooooo      ooooooooooooooooooooooooooooooooooooo      oo&      .....      &oooo
      ooooooooooooooooooooo      .....      ooooooooooooooooooooo      o      .....      ooooo
      o      .....      *oooooooooooooooooooo      .....      oo      .....      /oo
      ooooooooooooooooooooooooooooooooooooooooooooooooooooo      oo      .....      ooooo
      o      .....      ooooooooooooooooooooooooooooooooooooo      ooooo*      .....      oo
      /oo      /ooooooooo(      *oooo      .....      (oo      ooooo      .....      o%
      oo(      ooooooooooooooooooooooooooooo      .....      ooooooooooooo      .....      o
      o      .....      ooooooooooooooooooooooooooooooooooooooooooooooooooooo      .....      ooooooooo
      ooooooooooooooooooooooooooooooooooooooooooooooooooooo      .....      oo      .....      oo
      ooooooooooooooooooooooooooooooooooooooooooooo      .....      ooooooooooooo      .....      oo
      oooooooooo      .....      ooooo      .....      #oooooooooooo      .....      oooooo      .....      o
      ooooooooooooo      .....      oooooo      .....      *oo/      .....      ooooo      .....      o
      oooo      o      .....      ooooo      .....      &o      .....      ooooo      .....      o
      %oooo#      .....      ooooo      .....      oo      .....      oo      .....      oo      .....      o
      oooo      oo&      .....      *oooo      .....      #oooooooo      .....      oo      .....      oo      .....      #oooo
      oo      .....      %oooo*      .....      oo      .....      ooooo      .....      #oo*      .....      oo
      ooooo      ooooo*oooo(      .....      ooooo(      .....      ooooo      .....      oo*      .....      oo%      .....      oo
      oo*      .....      ooooooo*      .....      *oooo*/oooo*oooooooooooooooooooooooooooooooooooooooooooooooooooo
      o      .....      *      .....      ***      (oooo      &*****oooo*oooooooooooooooooooooooooooooooooooooooooooooooooooo
      oo*****oooooo      .....      *****oooo*      .....      *****      .....      ooooo
      o      .....      ooooooooooooo(      .....      ooooo      .....      *****      .....      ooooooooooooooooooooooooooooo
      ooooooooooooooooooooooooooooo

```

s---n-

abc.....i...l.n....s....x.z
평!!!! 실패..

<Hangman Game>

southern

```
.....e..hi..l.nop.rstu.....
      정답!! 휴~ 살았다!
```

(4) 코드에 대한 설명 및 해당 문제에 대한 고찰

(5) 이번 과제에 대한 느낀점

이번 과제에서는 랜덤 단어를 입력받는 것도 재밌었지만 자신만의 그림을 만드는 것도 매우 재밌었다. 이 두 가지 기능 모두 잘 활용한다면 더 재미있는 게임을 만들 수 있을 것 같다는 생각이 들었다. 또한 전에 했던 코드를 클래스 형식으로 다시 바꿔보니까, 클래스로 바꾼 코드들이 훨씬 보기도 편하고 정리가 잘 된 코드라는 느낌이 들었다. 앞으로도 클래스를 자주 사용할 수 있도록 습관을 만들어 나가야겠다.

(6) 궁금한 점이나 건의사항