2022학년도 2학기 출석수업대체(과제물)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **교과목명** | : | C++프로그래밍 |
|  | **학번** | : | 202234-153799 |
|  | **성명** | : | 한승환 |
|  | **연락처** | : | 010-2862-0200 |

목차

1. C++를 이용한 TIC-TAC-TOE 프로그램 작성

* 클래스의 멤버함수 및 용도
* 클래스의 데이터멤버 및 용도
* 멤버함수 설명

1. 매개변수의 자료형 및 용도
2. 처리내용 설명
3. 소스 프로그램
4. **C++를 이용한 TIC-TAC-TOE 프로그램 작성**

C++를 이용하여 TIC-TAC-TOE라는 게임을 프로그래밍 했으며 아래는 프로그래밍시 사용되었던 클래스의 멤버함수와 데이터멤버 그리고 그 용도를 나타내고 있다.

* **클래스의 멤버함수 및 용도**

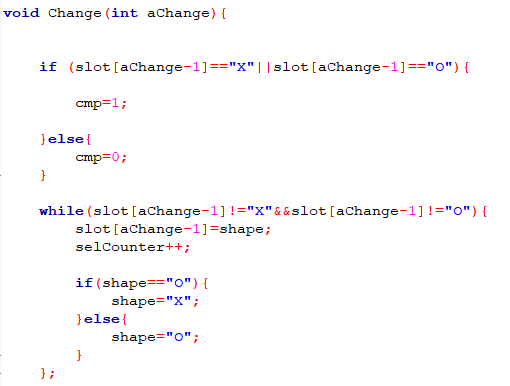
|  |  |
| --- | --- |
| **메소드** | **용도** |
| void StartingShape( ) | 먼저 시작할 모양을 선택 |
| void Grid( ) | TIC-TAC-TOE 게임판을 디스플레이 |
| void WinOrLose( ) | 승패 또는 무승부를 판별 |
| void Select( ) | 말을 둘 위치를 선택 |
| void Change(int aChange) | 선택된 위치에 해당 말을 둔다 |

* **클래스의 데이터멤버 및 용도**

|  |  |
| --- | --- |
| **속성** | **용도** |
| string slot[9] | 게임판의 각각 총 9개 칸을 지정 |
| bool OW | ‘O’말의 승리 판독 |
| bool XW | ‘X’말의 승리 판독 |
| bool D | 무승부 판독 |
| int select | 게임판의 말을 둘 칸 선택 |
| int selCounter | 번갈아가며 말을 주고받은 횟수 카운트 |
| string shape | 현재 말의 모양 지정 |
| bool cmp | 중복된 칸을 선택했을 경우 판독 |

* **멤버함수 설명**

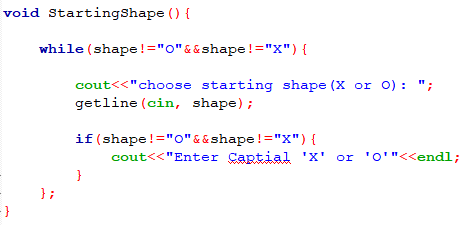
1. **매개변수의 자료형 및 용도**



위는 클래스 함수중 하나인 void Change(int aChange)의 내용이다. 이 함수는 클래스의 함수 중 유일하게 매개변수를 사용하고 있는데, 해당 매개변수는 정수형(int)자료형을 사용하고 있다. 그 용도는 말을 둘 칸을 말의 모양으로 채워넣기위해 선택된 칸의 번호(정수형 번호)를 의미하고 있다.

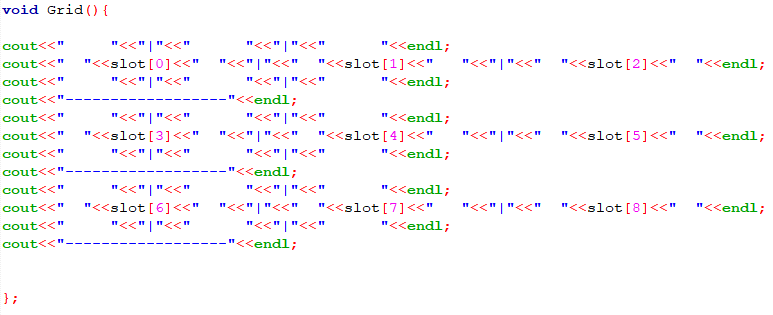
1. **각 메소드의 처리내용**

***void StartingShape( )***



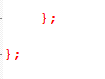
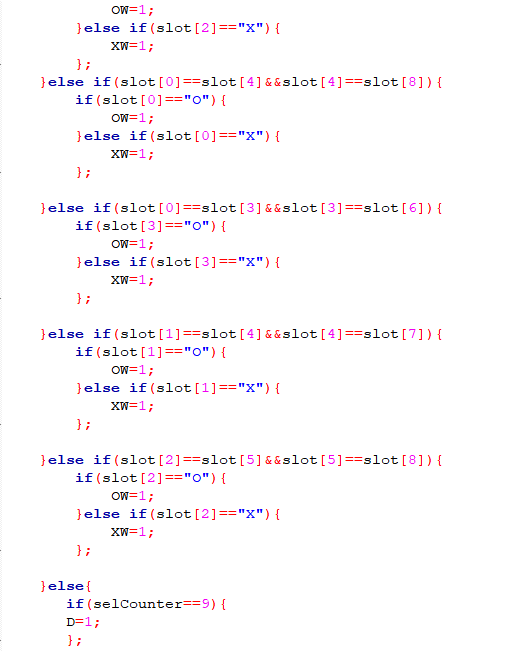
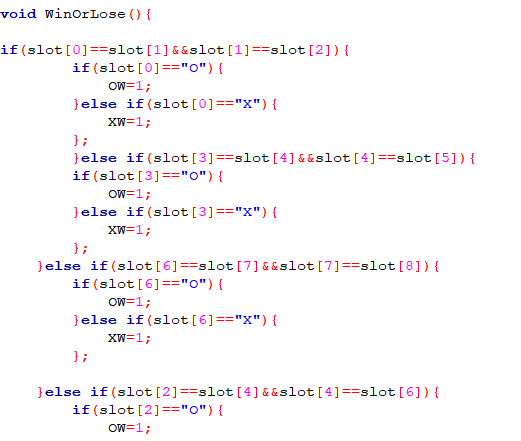
위의 멤버함수는 시작할 말의 모양을 설정하는 메소드이다. ‘shape’는 string형 변수로써 말의 모양인 ‘O’또는 ‘X’를 입력받아 저장한다. while함수를 사용하여 shape에 ‘O’또는 ‘X’값이 입력되지 않으면 계속해서 함수내의 명령문들을 반복하도록 설계되었으며 if 함수로 알맞은 값이 입력되지 않는 경우 코멘트를 표시하여 플레이어가 ‘X’또는 ‘O’를 입력하게끔 유도하고 있다.

***void Grid( )***



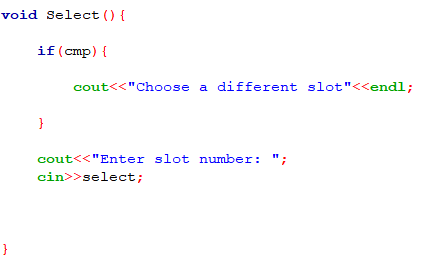
void Grid( )는 틱택토판을 화면상 표시하기위한 메소드로써 cout의 명령어로 이루어져 있다. 중간중간 칸별로 slot배열을 사용하여 칸 별 번호 또는 배열에 들어갈 모양(O/X)을 입력받아 표시한다.

***void WinOrLose( )***



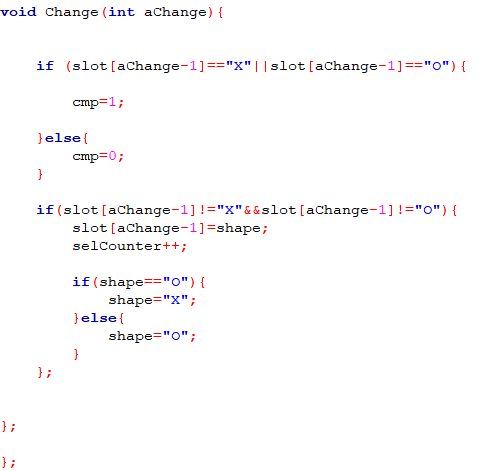
위의 함수는 게임이 진행되면서 각각의 상황에서 승자 또는 무승부를 판독하기 위한 메소드이다. if함수로 이루어져 있으며 모든 승패의 경우의 수의 조건을 확인하고 O의 승리(OW), X의 승리(XW), 무승부(D)를 불대수 변수를 이용하여 판별하고 있다. 또한 무승부는 모든 수를 놓고도 승패가 갈리지 않았을 경우를 selCounter라는 수를 둘때마다 +1씩 올라가는 카운터를 이용하여 무승부를 가려내고 있다.

***void Select( )***



void Select( )는 각 수를 놓을 차례에서 말을 놓을 위치를 정하기 위하여 플레이어가 원하는 위치에 말을 놓도록 안내하고 있다. 안내 문구를 디스플레이하기전에 cmp라는 불대수변수를 이용하여 그것이 참이면 “Choose a different slot”이라는 문구로 다른 칸을 선택하도록 요구하고 있는데, 이는(다음 메소드에서 후술하겠지만) 플레이어가 중복된 칸을 선택하였을 경우 cmp가 1, 즉 참이 되기 때문이다.

***void Change(int aChange)***



void Change(int aChange)는 aChange라는 정수형 매개변수를 전달받아 slot배열에 X또는 O를 저장하는 기능을 하고 있다. 이 때 if문을 사용하여 입력하려는 slot에 이미 X또는 O가 저장되어 있으면 불대수인 cmp가 참이 되도록 하여 중복된 입력을 방지하고 있다. 또한 두번째 if문을 사용하여 입력하려는 슬롯이 X나 O가 아닐때 if문 안의 명령들을 실행한다. 여기서 선택한 슬롯 안에 shape에 저장되어 있는 모양을 입력한다. 그릐고 수를 두었으므로 selCounter에 정수 1을 더하여 카운팅을 한다(무승부 판별용). 그리고 마지막 if문에서는 그 다음 수를 위하여 만약 현재 shape에 O가 저장되어있으면 X로, X가 저장되어 있으면 O로 변환하도록 되어있다.

slot배열에서 aChange에 1을 빼주는데 이는 slot 배열의 각 개체는 0부터 카운팅 되기 때문이다.(예, 1번칸은 slot[0]이다.)

1. **소스 프로그램**

**cpp 소스파일 내용**

#include <iostream>

using namespace std;

class TICTACTOE{

public:

string slot[9]={"1","2","3","4","5","6","7","8","9"};

bool OW=0, XW=0;

bool D=0;

int select;

int selCounter=0;

string shape;

bool cmp=0;

void StartingShape(){

while(shape!="O"&&shape!="X"){

cout<<"choose starting shape(X or O): ";

getline(cin, shape);

if(shape!="O"&&shape!="X"){

cout<<"Enter Captial 'X' or 'O'"<<endl;

}

};

}

void Grid(){

cout<<" "<<"|"<<" "<<"|"<<" "<<endl;

cout<<" "<<slot[0]<<" "<<"|"<<" "<<slot[1]<<" "<<"|"<<" "<<slot[2]<<" "<<endl;

cout<<" "<<"|"<<" "<<"|"<<" "<<endl;

cout<<"------------------"<<endl;

cout<<" "<<"|"<<" "<<"|"<<" "<<endl;

cout<<" "<<slot[3]<<" "<<"|"<<" "<<slot[4]<<" "<<"|"<<" "<<slot[5]<<" "<<endl;

cout<<" "<<"|"<<" "<<"|"<<" "<<endl;

cout<<"------------------"<<endl;

cout<<" "<<"|"<<" "<<"|"<<" "<<endl;

cout<<" "<<slot[6]<<" "<<"|"<<" "<<slot[7]<<" "<<"|"<<" "<<slot[8]<<" "<<endl;

cout<<" "<<"|"<<" "<<"|"<<" "<<endl;

cout<<"------------------"<<endl;

};

void WinOrLose(){

if(slot[0]==slot[1]&&slot[1]==slot[2]){

if(slot[0]=="O"){

OW=1;

}else if(slot[0]=="X"){

XW=1;

};

}else if(slot[3]==slot[4]&&slot[4]==slot[5]){

if(slot[3]=="O"){

OW=1;

}else if(slot[3]=="X"){

XW=1;

};

}else if(slot[6]==slot[7]&&slot[7]==slot[8]){

if(slot[6]=="O"){

OW=1;

}else if(slot[6]=="X"){

XW=1;

};

}else if(slot[2]==slot[4]&&slot[4]==slot[6]){

if(slot[2]=="O"){

OW=1;

}else if(slot[2]=="X"){

XW=1;

};

}else if(slot[0]==slot[4]&&slot[4]==slot[8]){

if(slot[0]=="O"){

OW=1;

}else if(slot[0]=="X"){

XW=1;

};

}else if(slot[0]==slot[3]&&slot[3]==slot[6]){

if(slot[3]=="O"){

OW=1;

}else if(slot[3]=="X"){

XW=1;

};

}else if(slot[1]==slot[4]&&slot[4]==slot[7]){

if(slot[1]=="O"){

OW=1;

}else if(slot[1]=="X"){

XW=1;

};

}else if(slot[2]==slot[5]&&slot[5]==slot[8]){

if(slot[2]=="O"){

OW=1;

}else if(slot[2]=="X"){

XW=1;

};

}else{

if(selCounter==9){

D=1;

};

};

};

void Select(){

if(cmp){

cout<<"Choose a different slot"<<endl;

}

cout<<"Enter slot number: ";

cin>>select;

}

void Change(int aChange){

if (slot[aChange-1]=="X"||slot[aChange-1]=="O"){

cmp=1;

}else{

cmp=0;

}

if(slot[aChange-1]!="X"&&slot[aChange-1]!="O"){

slot[aChange-1]=shape;

selCounter++;

if(shape=="O"){

shape="X";

}else{

shape="O";

}

};

};

};

int main(){

int a=0;

string confirm;

TICTACTOE tic;

tic.StartingShape();

cout<<"====GAME START===="<<endl;

while(tic.XW==0&&tic.OW==0&&tic.D==0){

cout<<tic.shape<<"'s turn"<<endl;

tic.Grid();

tic.Select();

tic.Change(tic.select);

tic.WinOrLose();

cout<<"=============================="<<endl;

};

cout<<"=============================="<<endl;

cout<<"=============================="<<endl;

cout<<"<<Final Result>>"<<endl;

tic.Grid();

if(tic.XW){

cout<<"Winner: X"<<endl;

cout<<"Winner: X"<<endl;

cout<<"Winner: X"<<endl;

}else if(tic.OW){

cout<<"Winner: O"<<endl;

cout<<"Winner: O"<<endl;

cout<<"Winner: O"<<endl;

}else{

cout<<"DRAW"<<endl;

cout<<"DRAW"<<endl;

cout<<"DRAW"<<endl;

}

return 0;

}