Дана игра крестики-нолики с полем 10 на 10. Нужно усложнить игру так, чтобы в ней присутствовали функции объектно-ориентированного программирования или немного переделать, но чтобы задача была примером ООП. Код задан ниже

#include "stdafx.h"

#include <iostream>

#include <windows.h>

#include <time.h>

#define LINE 5 //длина выигрышной линии

#define SIZE 10 //размер поля

using namespace std;

int checkLine(int i0, int j0, int di, int dj);

void startGame();

void printField();

int fieldValidation();

void movePlayer();

void movePC();

int field[SIZE][SIZE];

int main(void)

{

 int choice;

 while (1)

 {

 cout << "Tic Tac Toe" << endl << "Play? (1-yes, 0 - no)" << endl;

 cin >> choice;

 switch (choice)

 {

 case 1:

 for (int i = 0; i<SIZE; i++)

 for (int j = 0; j<SIZE; j++)

 field[i][j] = 0;

 startGame();

 break;

 case 0:

 return 0;

 default:

 cout << "Invalid command" << endl;

 break;

 }

 }

}

//подсчет длины линии одинаковых клеток из клетки (i0,j0) с шагом по координатам (di,dj)

int checkLine(int i0, int j0, int di, int dj)

{

 int i = i0; //координаты

 int j = j0; //начальной точки

 for (int l = 0; l<LINE;) //цикл проверки линии выигрышной длины

 {

 if (field[i][j] != field[i0][j0]) return l; //в клетке другое заполнение

 l++; //увеличение длины линии

 i+= di; //вычисление координат

 j+= dj; //следующей клетки

 if ((i < 0) || (i >= SIZE) || (j < 0) || (j >= SIZE)) return l; //выход за пределы поля

 }

 return LINE; //найдена линия одинаковых клеток выигрышной длины

}

void startGame()

{

 int k = 1;

 printField();

 while (k)

 {

 movePlayer();

 printField();

 k = fieldValidation();

 if (k)

 {

 Sleep(1000);

 movePC();

 printField();

 k = fieldValidation();

 }

 }

}

void printField()

{

 cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

 cout << " 0 1 2 3 4 5 6 7 8 9" << endl;

 for (int i = 0; i<SIZE; i++)

 {

 cout << i << " ";

 for (int j = 0; j<SIZE; j++)

 {

 switch (field[i][j])

 {

 case 0:

 cout << "\_ ";

 break;

 case 1:

 cout << "X ";

 break;

 case 2:

 cout << "O ";

 break;

 }

 }

 cout << endl;

 }

 cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

}

void movePlayer()

{

 int i, j;

 while (1)

 {

 cout << "Enter numbers of cell" << endl;

 cin >> i;

 cin >> j;

 if (field[i][j] == 0)

 {

 field[i][j] = 1;

 return;

 }

 else

 cout << "Cell is busy" << endl;

 }

}

void movePC()

{

//попытка поставить выигрышную линию O

 for (int i = 0; i<SIZE; i++)

 for (int j = 0; j<SIZE; j++)

 if (field[i][j] == 0) //клетка пустая, проверить, что поставив в нее O получим линию длиной LINE

 {

 field[i][j] = 2; //временно ставим в клетку O

 if ((checkLine(i,j,0,1) == LINE) || //проверка по горизонтали вправо

 (checkLine(i,j,0,-1) == LINE) || //проверка по горизонтали влево

 (checkLine(i,j,1,0) == LINE) || //проверка по вертикали вниз

 (checkLine(i,j,-1,0) == LINE) || //проверка по вертикали вверх

 (checkLine(i,j,1,1) == LINE) || //проверка по диагонали вправо-вниз

 (checkLine(i,j,-1,1) == LINE) || //проверка по диагонали вправо-вверх

 (checkLine(i,j,1,-1) == LINE) || //проверка по диагонали влево-вниз

 (checkLine(i,j,-1,-1) == LINE)) //проверка по диагонали влево-вверх

 {

 return; //построена выигрышная линия, оставляем O в клетке

 }

 field[i][j] = 0; //удаляем временно поставленный O из клетки

 }

//попытка помешать построить на следующем ходу выигрышную линию X

 for (int i = 0; i<SIZE; i++)

 for (int j = 0; j<SIZE; j++)

 if (field[i][j] == 0) //клетка пустая, проверить, что поставив в нее X получим линию длиной LINE

 {

 field[i][j] = 1; //временно ставим в клетку X

 if ((checkLine(i,j,0,1) == LINE) || //проверка по горизонтали вправо

 (checkLine(i,j,0,-1) == LINE) || //проверка по горизонтали влево

 (checkLine(i,j,1,0) == LINE) || //проверка по вертикали вниз

 (checkLine(i,j,-1,0) == LINE) || //проверка по вертикали вверх

 (checkLine(i,j,1,1) == LINE) || //проверка по диагонали вправо-вниз

 (checkLine(i,j,-1,1) == LINE) || //проверка по диагонали вправо-вверх

 (checkLine(i,j,1,-1) == LINE) || //проверка по диагонали влево-вниз

 (checkLine(i,j,-1,-1) == LINE)) //проверка по диагонали влево-вверх

 {

 field[i][j] = 2; //ставим в клетку O, чтобы помешать получить выигрышную линию из X

 return;

 }

 field[i][j] = 0; //удаляем временно поставленный X из клетки

 }

//попытка помешать построить на следующем ходу выигрышную линию-1 X

 for (int i = 0; i<SIZE; i++)

 for (int j = 0; j<SIZE; j++)

 if (field[i][j] == 0) //клетка пустая, проверить, что поставив в нее X получим линию длиной LINE-1

 {

 field[i][j] = 1; //временно ставим в клетку X

 if ((checkLine(i,j,0,1) == LINE-1) || //проверка по горизонтали вправо

 (checkLine(i,j,0,-1) == LINE-1) || //проверка по горизонтали влево

 (checkLine(i,j,1,0) == LINE-1) || //проверка по вертикали вниз

 (checkLine(i,j,-1,0) == LINE-1) || //проверка по вертикали вверх

 (checkLine(i,j,1,1) == LINE-1) || //проверка по диагонали вправо-вниз

 (checkLine(i,j,-1,1) == LINE-1) || //проверка по диагонали вправо-вверх

 (checkLine(i,j,1,-1) == LINE-1) || //проверка по диагонали влево-вниз

 (checkLine(i,j,-1,-1) == LINE-1)) //проверка по диагонали влево-вверх

 {

 field[i][j] = 2; //ставим в клетку O, чтобы помешать получить (выигрышную линию-1) из X

 return;

 }

 field[i][j] = 0; //удаляем временно поставленный X из клетки

 }

//выполняем ход O в случайную свободную клетку поля

 int i, j;

 srand(time(0));

 i = rand() % (SIZE); //генерация случайных координат клетки

 j = rand() % (SIZE);

 while (1)

 {

 if (field[i][j] == 0)

 {

 field[i][j] = 2;

 return;

 }

 else //если клетка занята, снова генерировать случайные координаты клетки

 {

 i = rand() % (SIZE);

 j = rand() % (SIZE);

 }

 }

}

int fieldValidation()

{

 int full = 1, win = 0;

 for (int i = 0; i<SIZE; i++)

 for (int j = 0; j<SIZE; j++)

 if (field[i][j] != 0) //клетка не пустая, проверить, что из нее выходит линия длиной LINE

 if ((checkLine(i,j,0,1) == LINE) || //проверка по горизонтали вправо

 (checkLine(i,j,1,0) == LINE) || //проверка по вертикали вниз

 (checkLine(i,j,1,1) == LINE) || //проверка по диагонали вправо-вниз

 (checkLine(i,j,1,-1) == LINE)) // //проверка по диагонали влево-вниз

 {

 win = field[i][j];

 }

 if (win)

 {

 switch (win)

 {

 case 1:

 cout << "Player win!" << endl << "Congratulations!!!!" << endl;

 return 0;

 case 2:

 cout << "PC win!" << endl << "Try again." << endl;

 return 0;

 }

 }

 for (int i = 0; i<SIZE; i++)

 for (int j = 0; j<SIZE; j++)

 if (field[i][j] == 0) // Проверка, на наличие свободных ячеек в поле

 full = 0;

 if (full)

 {

 cout << "Full field!" << endl;

 return 0;

 }

 return 1;

}