**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра Автоматизированных систем управления

Лабораторная работа № 1

По ОБЪЕКТО-ОРИЕНТИРОВАННОМУ ПРОГРАММИРОВАНИЮ

«Объекты и классы. Инкапсуляция»

Студент \_\_\_\_\_\_\_\_\_\_\_\_ Бутусова В.М.

Группа ПМ-21-2

Руководитель

Доцент \_\_\_\_\_\_\_\_\_\_\_\_ Кургасов В. В.

Липецк 2021г.

**Цель работы:** научиться описывать объекты реального мира с использованием классов.

**Задание кафедры:** используя механизм инкапсуляции, описать объект реального мира. Добавить к имеющимся методам и атрибутам ещё по три произвольных пункта.

Вариант 3:

|  |  |  |
| --- | --- | --- |
| 3 | Крестики-нолики | Поместить крестик на поле в позицию X  Поместить нолик в позицию X  Проверить победителя |

**Исходный код:**

#include <iostream>

using namespace std;

class X0

{

private://в private будут записаны массив поля и все методы, вызываемые из метода igra

char pole[4][4]; //создание массива для поля

//делаем ход

void shag(int x, int y, int i)

{

if (i % 2) //если iтый номер делится на 2 с остатком, значит ход второго игрока

pole[x][y] = '0';

else //инче - ход первого игрока

pole[x][y] = 'X';

pokazaty(); //выводим поле на экран

}

//вывод поля

void pokazaty()

{

cout << endl << "Поле:" << endl;

cout << " " << 1 << " " << 2 << " " << 3 << endl; //выод номеров координат по горизонтали

for (int i = 1; i < 4; i++)

{

for (int j = 1; j < 4; j++)

{

if (j == 1) // вывод номеров координат по вертикали

cout << i << ' ';

if ((pole[i][j] == 'X') || (pole[i][j] == '0')) //проверка условия, что элемент массива заполнен 'X' или '0', и вывод поля с добавленным символом

cout << pole[i][j] << ' ';

else //иначе в пустых клетках ставим '.'

cout << '.' << ' ';

}

cout << endl;

}

}

//проверка победителя

int proverka()

{

//перебор всевозможных вариантов

if (((pole[1][1] == 'X') && (pole[1][2] == 'X') && (pole[1][3] == 'X')) || ((pole[1][1] == '0') && (pole[1][2] == '0') && (pole[1][3] == '0')))

{

return 0;

}

if (((pole[2][1] == 'X') && (pole[2][2] == 'X') && (pole[2][3] == 'X')) || ((pole[2][1] == '0') && (pole[2][2] == '0') && (pole[2][3] == '0')))

{

return 0;

}

if (((pole[3][1] == 'X') && (pole[3][2] == 'X') && (pole[3][3] == 'X')) || ((pole[3][1] == '0') && (pole[3][2] == '0') && (pole[3][3] == '0')))

{

return 0;

}

if (((pole[1][1] == 'X') && (pole[2][2] == 'X') && (pole[3][3] == 'X')) || ((pole[3][1] == '0') && (pole[3][2] == '0') && (pole[3][3] == '0')))

{

return 0;

}

if (((pole[3][1] == 'X') && (pole[2][2] == 'X') && (pole[1][3] == 'X')) || ((pole[3][1] == '0') && (pole[2][2] == '0') && (pole[1][3] == '0')))

{

return 0;

}

if (((pole[1][1] == 'X') && (pole[2][1] == 'X') && (pole[3][1] == 'X')) || ((pole[1][1] == '0') && (pole[2][1] == '0') && (pole[3][1] == '0')))

{

return 0;

}

if (((pole[1][2] == 'X') && (pole[2][2] == 'X') && (pole[3][2] == 'X')) || ((pole[1][2] == '0') && (pole[2][2] == '0') && (pole[3][2] == '0')))

{

return 0;

}

if (((pole[1][3] == 'X') && (pole[2][3] == 'X') && (pole[3][3] == 'X')) || ((pole[1][3] == '0') && (pole[2][3] == '0') && (pole[3][3] == '0')))

{

return 0;

}

else

return 1;

}

public: //чтобы пользователь не мог вызывать другие методы самостоятельно, нарушая ход игры, в public оставим только метод, реализующий механизм игры

void igra()

{

pokazaty(); //вызываем метод, который выводит пустое поле на экран

for (int i = 0; i < 9; i++) //начинаем цикл шагов

{

cout << endl << i%2+1 << " игрок, ведите координаты: ";

int x, y;

cin >> x >> y;

shag(x, y, i);

int result = proverka(); //в rezult сохраняем возвращенноё значение метода proverka()

if (result == 0) //если result 0, выводим на экран победителя и останавливаем цикл

{

cout << "Победил " << i % 2 + 1 << " игрок." << endl;

break;

}

if (i == 8) //если было сделано 9 шагов и никто не победил, сообщаем о ничье

{

cout << "Победила дружба! :)" << endl;

}

}

}

};

void main()

{

setlocale(LC\_ALL, "rus");

X0 pt;

pt.igra();

}

**Результаты работы программы:**

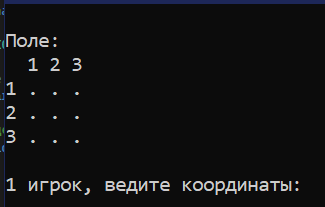
****

Рисунок 1 – начало выполнения программы. Первому игроку предлагается ввести координаты ‘X’.

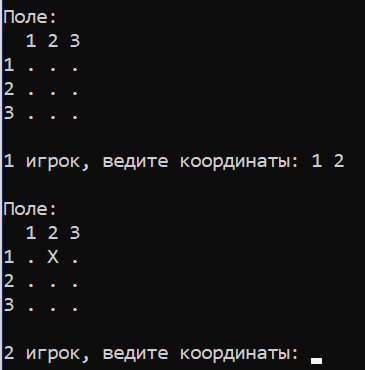


Рисунок 2 – первый игрок сделал ход. Второй игрок должен ввести координаты ‘0’.

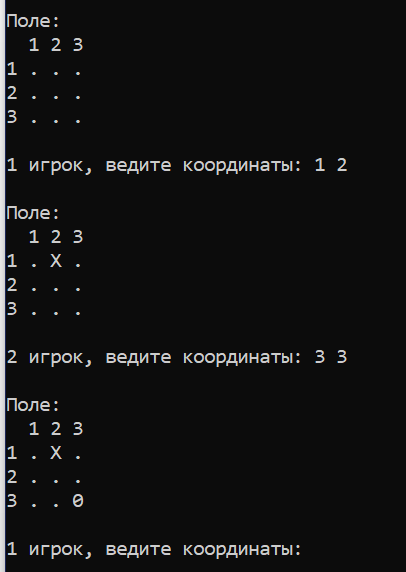


Рисунок 3 – второй игрок сделал ход.

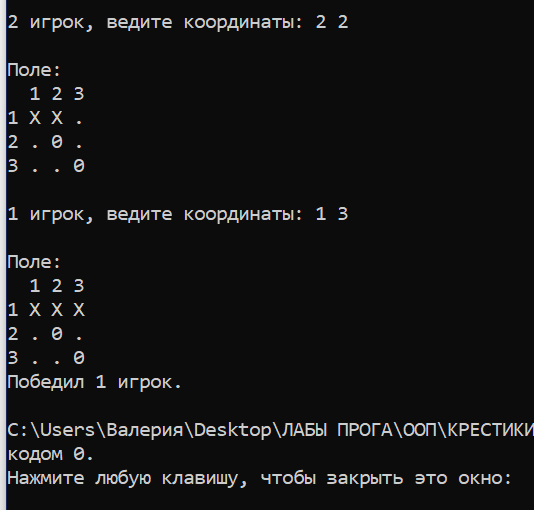


Рисунок 4 – исход 1: победил первый игрок.

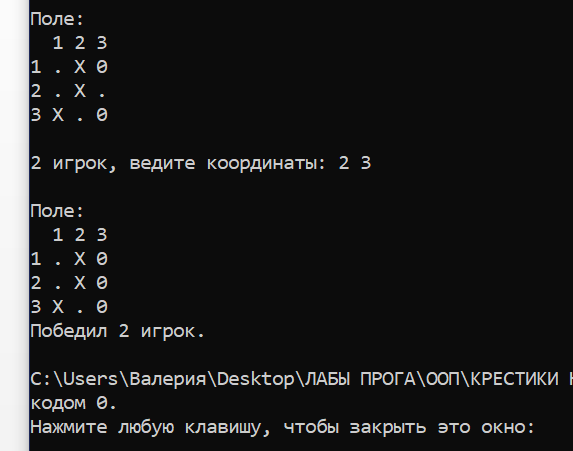


Рисунок 5 – исход 2: победил второй игрок.

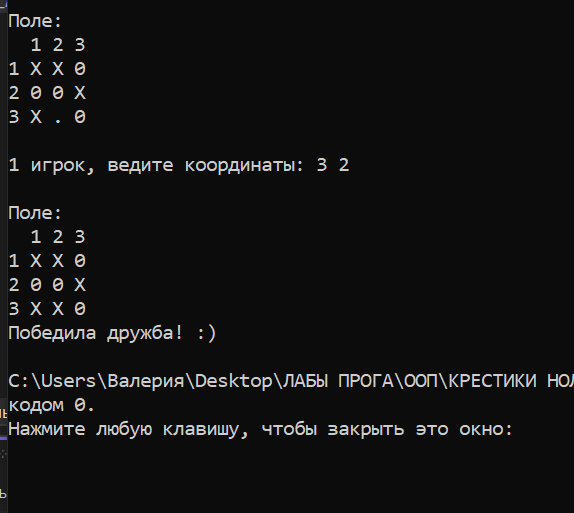


Рисунок 6 – исход 3: ничья.

**Вывод:** я научилась описывать объекты реального мира с использованием классов.

**Контрольные вопросы.**

1. Что такое класс?

Класс —модель для создания объектов определённого типа, описывающая их структуру (набор полей и их начальное состояние) и определяющая алгоритмы (функции или методы) для работы с этими объектами.

1. Чем отличается экземпляр класса от класса?

Класс - это некий шаблон, который задаёт структуру и поведение для любого экземпляра класса.  
Экземпляр класса - это объект, который сделан по шаблону класса.

1. Что такое свойство и метод класса?

Методы класса — это его функции.  
Свойства класса — его переменные.

1. Какие существуют уровни доступа к свойствам и методам класса?

**Private** — объявляет метод или свойство доступным только в том классе в котором он присутствует. То есть к private методам и свойствам мы не можем обращаться ни из объектов, ни из дочерних классов.  
**Protected** — объявляет метод или свойство защищенными. То есть такими, которые не могут быть доступны из объекта, реализующего класс, но вполне может быть использовано в дочерних классах.  
**Public** — публичный. Классы и методы, объявленные public, могут быть доступны как внутри самого класса, так и в дочерних классах, и в объектах, реализовавших класс.

1. Что означает понятие инкапсуляция?

Инкапсуляция (encapsulation) - это механизм, который объединяет данные и код, манипулирующий этими данными, а также защищает и то, и другое от внешнего вмешательства или неправильного использования.