

Министерство образования и науки Российской Федерации
Московский государственный технический университет имени Н.Э.Баумана
Факультет «Робототехника и комплексная автоматизация»
Кафедра «Системы автоматизированного проектирования»

Отчет по практической работе №3

По курсу «Программное обеспечение САПР»

Выполнил:

Студент Гусаров Аркадий
Группа РК6-33Б

Проверил:

Дата _____
Подпись _____

Москва, 2020 г.

1. Задание:

Разработать ООП для компьютерной реализации игры с покрытием клеток шахматной доски (10x10) фишками домино. При этом за 1 ход можно покрыть любую пару смежных клеток без наложения фишек и нарушения границ доски. Проигрывает тот, кто не может сделать очередной ход. Эта игра должна быть реализована в варианте, когда противниками являются компьютер и человек. Человек может выбирать произвольные ходы, указывая позиции своей фишки по запросу стандартного ввода. При этом любая ошибка ввода должна приводить к аварийному прерыванию партии игры. Компьютер должен отвечать симметричными ходами относительно центра шахматной доски. Все ходы обоих игроков должны отображаться в клетках стандартного вывода шахматной доски. Игра должна завершаться диагностикой ее итогового результата.

Гусаров

Вариант G08

2. Описание входных данных:

В поток стандартного ввода передаются координаты фишки, в которых она располагается.

Пример входных данных:

./a.out

a1 b1

3. Описание работы алгоритма:

В цикле программа поочередно предоставляет ходы компьютеру и человеку. Получая данные от человека, они проверяются на правильность, затем ход делает компьютер, после чего преобразованное поле выводится в поток стандартного вывода. Процедура повторяется. Программа выводит победителя в случае, если дальнейший ход противника невозможен. В случае некорректного ввода координат со стороны человека, происходит аварийное прерывание игры.

4. Описание выходных данных:

Результатом работы является диагностика итогового результата игры.

5. Код программы:

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

class Domino
{
private:
    int sizeofBoard = 10;
    char board[10][10];

public:
    Domino() { boardFill(); };
    void boardFill();
    void printBoard();
```

```

void moveUser();
void movePC(int[], int[]);
char letterToIndex(char);
};

```

```

void Domino::boardFill()
{
    for (int i = 0; i < 10; i++)
        for (int j = 0; j < 10; j++)
            board[i][j] = '.';
}

```

```

void Domino::printBoard()
{
    cout << "\ta b c d e f g h i j\n\n";
    for (int i = 0; i < sizeOfBoard; i++)
    {
        cout << i << "\t";
        for (int j = 0; j < sizeOfBoard; j++)
            cout << board[i][j] << " ";

        cout << "\t" << i << "\n";
    }
    cout << "\n\ta b c d e f g h i j\n";
}

```

```

void Domino::moveUser()
{
    int posStart[2];
    int posEnd[2];
    char posS[2];
    char posE[2];
    char userSymbol = 'u';

    scanf("%c%c %c%c", &posS[0], &posS[1], &posE[0], &posE[1]);
    printf("\n=====Your input: %c%c %c%c\n=====\\n", posS[0],
posS[1], posE[0], posE[1]);

    posStart[0] = int(posS[1]) - 48;
    posStart[1] = letterToIndex(posS[0]);
    posEnd[0] = int(posE[1]) - 48;
    posEnd[1] = letterToIndex(posE[0]);

    if ((posStart[0] == posEnd[0]) && (posStart[1] == posEnd[1]))
    {
        cout << "ERROR: No difference!" << endl;
        exit(3);
    }
}

```

```

    }
    else if ((posStart[0] < 0) || (posStart[0] > 9) || (posEnd[0] < 0) || (posEnd[0] > 9) || (posStart[1] < 0) ||
(posStart[1] > 9) || (posEnd[1] < 0) || (posEnd[1] > 9))
    {
        cout << "ERROR: Off the chessboard!" << endl;
        exit(4);
    }
    else if ((board[posStart[0]][posStart[1]] != '.') || (board[posEnd[0]][posEnd[1]] != '.'))
    {
        cout << "ERROR: Is not free!" << endl;
        exit(5);
    }
    else if ((abs(posStart[0] - posEnd[0]) >= 1) && (abs(posStart[1] - posEnd[1]) >= 1))
    {
        cout << "ERROR: Chips are diagonally!" << endl;
        exit(6);
    }
    else if ((abs(posStart[0] - posEnd[0]) > 1) || (abs(posStart[1] - posEnd[1]) > 1))
    {
        cout << "ERROR: Over-distance!" << endl;
        exit(7);
    }
    else
    {
        board[posStart[0]][posStart[1]] = userSymbol;
        board[posEnd[0]][posEnd[1]] = userSymbol;
    }

    movePC(posStart, posEnd);
    printBoard();
}

```

```

void Domino::movePC(int posStart[], int posEnd[])
{
    int res11, res12, res21, res22;
    char PCsymbol = 'c';

    res11 = 9 - posStart[0];
    res12 = 9 - posEnd[0];
    res21 = 9 - posStart[1];
    res22 = 9 - posEnd[1];

    if ((board[res11][posStart[1]] == '.') && (board[res12][posEnd[1]] == '.'))
    {
        board[res11][posStart[1]] = PCsymbol;
        board[res12][posEnd[1]] = PCsymbol;
    }
    else if ((board[posStart[0]][res21] == '.') && (board[posEnd[0]][res22] == '.'))

```

```

{
    board[posStart[0]][res21] = PCSymbol;
    board[posEnd[0]][res22] = PCSymbol;
}
else if ((board[res11][res21] == '.') && (board[res12][res22] == '.'))
{
    board[res11][res21] == PCSymbol;
    board[res12][res22] == PCSymbol;
}

else
{
    printBoard();
    cout << "You are winner!" << endl;
    exit(2);
}
}

```

```

char Domino::letterToIndex(char pos)

```

```

{
    switch (pos)
    {
    case 'a':
        return 0;
    case 'b':
        return 1;
    case 'c':
        return 2;
    case 'd':
        return 3;
    case 'e':
        return 4;
    case 'f':
        return 5;
    case 'g':
        return 6;
    case 'h':
        return 7;
    case 'i':
        return 8;
    case 'j':
        return 9;
    default:
        return -1;
    }
}

```

```

int main(int argc, char *argv[])

```

```

{
    if (argc != 1)
    {
        cout << "Incorrect number of arguments" << endl;
        return 1;
    }

    Domino U;
    U.printBoard();

    printf("Input: ");
    while (getchar() != '\0')
    {
        U.moveUser();
    }

    return 0;
}

```

6. Результаты тестирования:

```

=====
Your input: a1 b1
=====
      a b c d e f g h i j
0      . . . . . 0
1      u u . . . . . 1
2      . . . . . 2
3      . . . . . 3
4      . . . . . 4
5      . . . . . 5
6      . . . . . 6
7      . . . . . 7
8      c c . . . . . 8
9      . . . . . 9
      a b c d e f g h i j

```

```

=====
Your input: d3 d4
=====
      a b c d e f g h i j
0      . . . . . 0
1      u u . . . . . 1
2      . . . . . 2
3      . . . u . . . . 3
4      . . . u . . . . 4
5      . . . c . . . . 5
6      . . . c . . . . 6
7      . . . . . 7
8      c c . . . . . 8
9      . . . . . 9
      a b c d e f g h i j

```

```

      a b c d e f g h i j
0      . . . . . 0
1      u u . . . . . 1
2      . . . . . 2
3      . . . u . . . . 3
4      . . . u . . . . 4
5      . . . c . . . . 5
6      . . . c . . . . 6
7      . . . . . 7
8      c c . . . . . 8
9      . . . . . 9
      a b c d e f g h i j
g0 g0
=====
Your input: g0 g0
=====
ERROR: No difference!

```

```

      a b c d e f g h i j
0      . . . . . 0
1      u u . . . . . 1
2      . . . . . 2
3      . . . . . 3
4      . . . . . 4
5      . . . . . 5
6      . . . . . 6
7      . . . . . 7
8      c c . . . . . 8
9      . . . . . 9
      a b c d e f g h i j
a0 a1
=====
Your input: a0 a1
=====
ERROR: Is not free!

```

```

      a b c d e f g h i j
0      u . . . . . . . . 0
1      u . . . . . . . . 1
2      . . . . . . . . . 2
3      . . . . . . . . . 3
4      . . . . . . . . . 4
5      . . . . . . . . . 5
6      . . . . . . . . . 6
7      . . . . . . . . . 7
8      c . . . . . . . . 8
9      c . . . . . . . . 9

      a b c d e f g h i j
c2 d4

=====
Your input: c2 d4
=====
ERROR: Chips are diagonally!

```

```

=====
Your input: d4 d5
=====
      a b c d e f g h i j
0      . . c . . . . u u . 0
1      u u c . u u . . u u 1
2      u u . u u . c . u u 2
3      u u u c c . c u . . 3
4      u u u u . u u u c c 4
5      u u c u . c c c c c 5
6      c c c u u . u c . . 6
7      c c . c c . u . c c 7
8      c c u . c c . . c c 8
9      . . u . . . . c c . 9

      a b c d e f g h i j
You are winner!

```

Список литературы:

1. Программирование на языке высокого уровня (базовый курс)
<http://bigor.bmstu.ru/?cnt/?doc=VU/base.cou>
2. Конспекты с лекций по курсу «Объектно-ориентированное программирование».
3. Прикладное программирование на языке C++ : учебное пособие / Т.М. Волосатова, С.В. Родионов, Д.Т. Шварц. – Москва : Издательство МГТУ им. Н. Э. Баумана, 2015. – 146, [2] с. : ил.