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

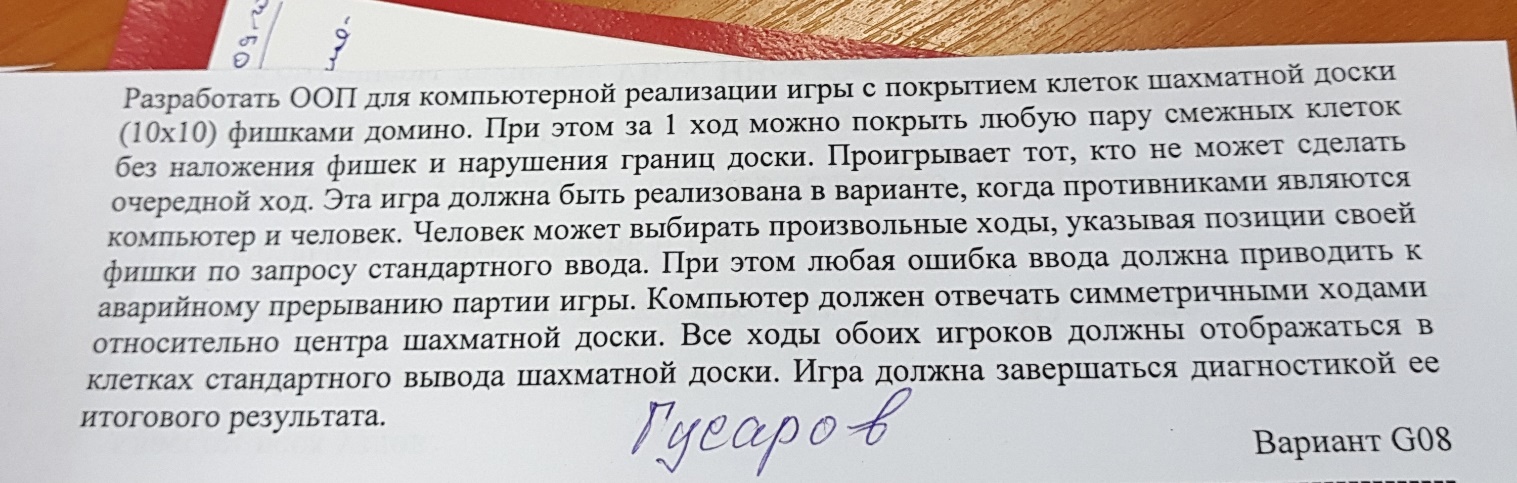
Проверил:

Дата

Подпись

Москва, 2020 г.

1. **Задание:**



1. **Описание входных данным:**

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

Пример входных данных:

./a.out a1 b1

1. **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.

1. **Описание выходных данных:**

Результатом работы является диагностика итогового результата игры.

1. **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=================\nYour 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')

{

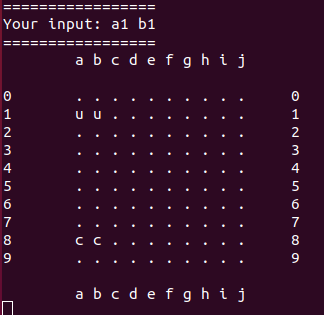
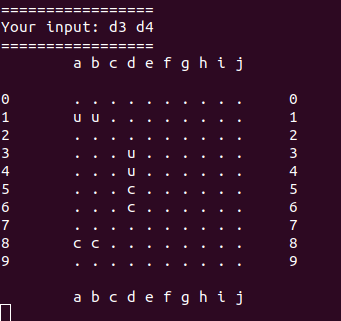
U.moveUser();

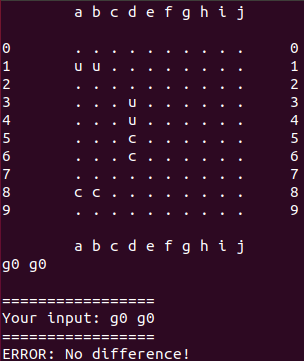
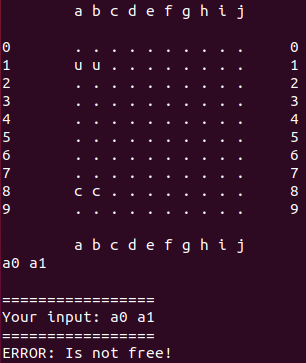
}

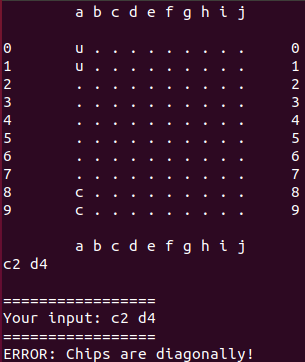
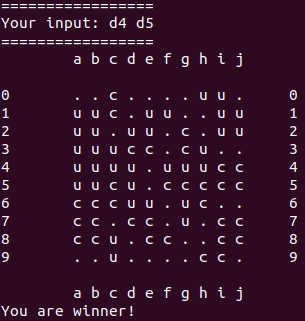
return 0;

}

# Результаты тестирования:

**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 С++ : учебное пособие / Т.М. Волосатова, С.В. Родионов, Д.Т. Шварц. – Москва : Издательство МГТУ им. Н. Э. Баумана, 2015. – 146, [2] с. : ил.