Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технический университет

имени Н.Э. Баумана

(национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

Факультет: Робототехника и комплексная автоматизация

Кафедра: Системы автоматизированного проектирования (РК6)

Отчет по лабораторной работе № 1

Группа: РК6-32Б

Студент: Журавлев Н.В.

Москва, 2020 г.

# Задача:

Разработать ООП для компьютерной реализации игры с переходами одностороннего короля на шахматной доске по следующим правилам. Сначала король стоит в поле h8. Два игрока должны по очереди переставлять его по горизонтали, вертикали или диагонали на 1 клетку в направлении поля а1. Проигрывает тот, кто не может сделать свой очередной ход. Эта должна быть реализована в варианте, когда противниками являются компьютер и человек. Человек может делать произвольные ходы по запросу стандартного ввода. При этом любое нарушение правил должно приводить к аварийному прерыванию игры. Компьютер должен отвечать ходами короля по полям с четными смещениями по горизонтали и по вертикали от h8, co стандартным выводом его положения на шахматной доске. Игра должна завершаться диагностикой ее итогового результата.

Входные данные:

Поля шахматной доски.

Описание алгоритма:

1) Переменная класс Desk вызывает метод print\_desk. В нём печатает строчку a b c d e f g h, затем в цикле for выводит цифру и знак х, если номер поля не равен полю в поле класса.

2) Затем заходит в цикл while, где условием выхода является попадание в а1

3) Вызывается метод move для игрока в случае возврата -1 завершает программу

4) В этом методе вызывается cin для переменной типа string, которая проходит проверку на то, чтобы буква была или такой же, или на 1 ближе к а, цифра такая же или -1, затем записывается в переменную класса Desk.

5) Печатается доска

6) Вызывается метод move для компьютера в случае возврата -1 завершает программу

7) В этом методе: если левая клетка свободна, чётная относительно а , то делает ход, иначе по такому же принципу проверяется клетка вниз-влево по диагонали и нижняя, если ход сделать невозможно возвращает -1.

8) На выходе проверяется кто должен делать ход, когда король попал на а1. В зависимости от этого печатается результат.

# Выходные данные:

# Результат игры

# Текст программы:

# #include <cstdlib>

# #include <string>

# #include <iostream>

# using namespace std;

# class Desk{

# protected:

# string pos;

# public:

# Desk(string new\_pos) : pos(new\_pos){}

# void print\_desk();

# string get\_pos();

# void set\_pos(string new\_pos);

# };

# class Gambler {

# protected:

# Desk\* pos;

# public:

# Gambler(Desk& new\_pos) {pos = &new\_pos;}

# virtual int move() = 0;

# virtual ~Gambler(){};

# };

# class Man : public Gambler {

# public:

# Man (Desk& new\_pos) : Gambler(new\_pos){}

# virtual int move();

# };

# class Comp : public Gambler {

# public:

# Comp (Desk& new\_pos) : Gambler(new\_pos){}

# virtual int move();

# };

# void Desk::set\_pos(string new\_pos){

# pos = new\_pos;

# }

# string Desk::get\_pos() {

# return pos;

# }

# void Desk::print\_desk() {

# int i;

# int j;

# char s[2];

# cout << " a b c d e f g h\n";

# for(i = 8; i > 0; i--) {

# cout << i << ' ';

# s[1] = '0' + i;

# for(j=0; j < 8; j++) {

# s[0] = 'a' + j;

# if (pos[0] == s[0] and pos[1] == s[1]) {

# cout << 'k' << ' ';

# } else {

# cout << 'x' << ' ';

# }

# }

# cout << i << "\n";

# }

# cout << " a b c d e f g h\n\n";

# }

# int check\_move(string new\_pos, string locate) {

# if (((locate[0] == new\_pos[0] + 1 || locate[0] == new\_pos[0])

# && (locate[1] == new\_pos[1] + 1 || locate[1] == new\_pos[1])

# && new\_pos.length() == 2 && new\_pos[0] >= 'a' && new\_pos[0] <= 'h'

# && new\_pos[1] >= '1' && new\_pos[1] <= '8') && !(locate[1] == new\_pos[1] && locate[0] == new\_pos[0]))

# return 0;

# else

# return 1;

# }

# int Man::move() {

# cout << "Ваш ход: ";

# string new\_pos;

# cin >> new\_pos;

# string locate = pos->get\_pos();

# if (check\_move(new\_pos, locate) == 0)

# pos->set\_pos(new\_pos);

# else

# return -1;

# cout << "\n";

# return 0;

# }

# int Comp::move() {

# string new\_pos = " ";

# string locate = pos->get\_pos();

# while (check\_move(new\_pos, locate)) {

# new\_pos[0] = locate[0] - 1;

# new\_pos[1] = locate[1];

# if (!(new\_pos[0] % 2 == 0 && new\_pos[1] % 2 == 0) || check\_move(new\_pos, locate)) {

# new\_pos[0] = locate[0] - 1;

# new\_pos[1] = locate[1] - 1;

# if (!(new\_pos[0] % 2 == 0 && new\_pos[1] % 2 == 0) || check\_move(new\_pos, locate)) {

# new\_pos[0] = locate[0];

# new\_pos[1] = locate[1] - 1;

# if (!(new\_pos[0] % 2 == 0 && new\_pos[1] % 2 == 0) || check\_move(new\_pos, locate)) {

# new\_pos[0] = locate[0] - 1;

# new\_pos[1] = locate[1];

# if (check\_move(new\_pos, locate)) {

# new\_pos[0] = locate[0];

# new\_pos[1] = locate[1] - 1;

# if (check\_move(new\_pos, locate)) {

# return -1;

# }

# }

# }

# }

# }

# }

# cout << "Ход компьютера: ";

# pos->set\_pos(new\_pos);

# cout << new\_pos << "\n";

# return 0;

# }

# int main() {

# string pos = "h8";

# char lose = '0';

# Desk desk(pos);

# Gambler \*g[] = {new Man(desk), new Comp(desk)};

# pos = "a1";

# desk.print\_desk();

# while(desk.get\_pos() != pos) {

# if (g[0]->move() != -1) {

# desk.print\_desk();

# if (g[1]->move() == -1) {

# lose = '1';

# } else {

# desk.print\_desk();

# }

# } else {

# cout << "Неверный ход\n ";

# delete g[0];

# delete g[1];

# return -1;

# }

# }

# if (lose == '0') {

# cout << "Победил компьютер\n";

# } else {

# cout << "Чудом победил пользователь";

# }

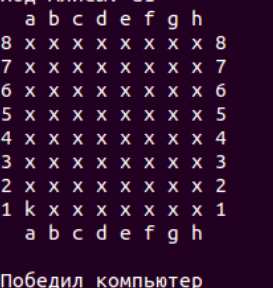
# delete g[0];

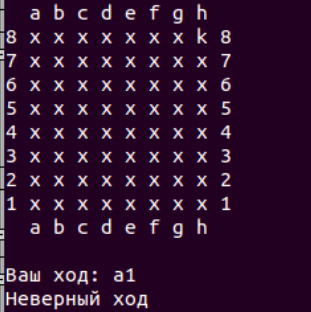
# delete g[1];

# return 0;

# }}

# Результаты тестирование:





# Список Литература:

1) База и Генератор Образовательных Ресурсов [Электронный ресурс]. – Режим доступа: [http://bigor.bmstu.ru/?cnt/?doc=VU/base.cou](https://vk.com/away.php?to=http%3A%2F%2Fbigor.bmstu.ru%2F%3Fcnt%2F%3Fdoc%3DVU%2Fbase.cou&cc_key=), свободный – (05.04.2020).

2) Т.М. Волосатова, С.В. Родионов Автоматизация проектирования лексических анализаторов. Учебное пособие для практических занятий по курсу Лингвистическое Обеспечение САПР Москва Издательство МГТУ им. Н.Э. Баумана 2005