Министерство образования и науки Российской Федерации

МГТУ им .Н.Э Баумана

Факультет «Робототехника и комплексная автоматизация»

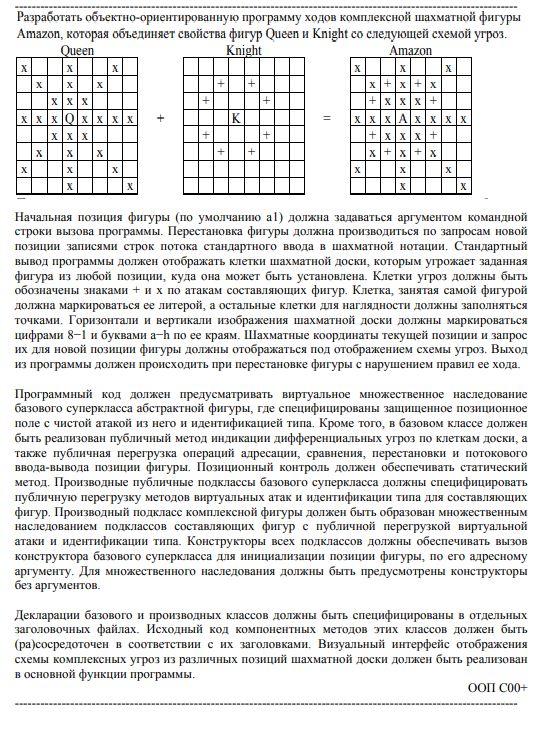
Кафедра «РК6»

**Отчет по лабораторной работе**

По курсу «Объектно-ориентированное программирование»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Выполнил: |  |  | Студент | Журавлев Н.В. |
|  |  |  | Группа | РК6-22Б |
|  |  |  |  |  |
| Проверил: |  |  |  | Козов А.В. |
|  |  |  | Дата | 13.05.2020 |
|  |  |  | Подпись | \_\_\_\_\_\_\_\_\_\_\_ |

**Домашнее задание №1**



**Описание алгоритма**

1)Проверка вводимого значения из командной строки.

2) Объявление фигуры типа Amazon.

3) Проверяем каждую клетку и ставим марку(.,+,х).

4) Сначала ставим метки первой фигуры(+),если она соответствует условию(x + 1 == y || y + 1 == x) && (x + y == 3) (где х и у позиции относительно начальной точки расположении).  
5)Аналогично для знака х,с условием x == 0 || y == 0 || x==y || x + y < 3

6) Ввод с последующей обработкой с пункта 3, пока не будет введено неверное значение.

**Текст программы**

**chess.cpp**

#include <iostream>

#include "chesslib.h"

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

if(argc < 2) {

argv[1] = (char\*)"nil";

}

if(Figure::deskout(argv[1]) || argv[1][3] == '\0') {

std::cout << "Correct " << argv[1];

std::cout << " to " << (argv[1] = (char\*)"a1") << " Default\n";

std::cout << "Usage example: chess+ e5\n";

}

Amazon f(argv[1]);

do{

f.danger();

std::cout << f << '-' << f.isa();

} while (std::cin >> f);

return(0);

}

**chesslib.cpp**

#include "chesslib.h"

int Knight::attack(char\* p)

{

if(deskout(p) > 0)

return(0);

int x = p[0] - pos[0];

int y = p[1] - pos[1];

if(x < 0) x = -x;

if(y < 0) y = -y;

if((x + 1 == y || y + 1 == x) && (x + y == 3))

return(1);

return(0);

}

int Queen::attack(char\* p)

{

if(deskout(p) > 0)

return(0);

int x = p[0] - pos[0];

int y = p[1] - pos[1];

if(x < 0) x = -x;

if(y < 0) y = -y;

if(x == 0 || y == 0 || x==y || x + y < 3)

return(2);

return(0);

}

int Amazon::attack(char\* s)

{

if(Knight::attack(s) > 0)

return(1);

if(Queen::attack(s) > 0)

return(2);

return(0);

}

**chessman.cpp**

#include "chessman.h"

using std::cout;

Figure::Figure(char\* p)

{

pos[0] = p[0]; pos[1] = p[1]; pos[2] = pos[3] = '\0';

}

Figure::operator char\*()

{

return pos;

}

int Figure::deskout(char\* p)

{

return((p[0] > 'h') || (p[0] < 'a') || (p[1] < '1') || (p[1] > '8'));

}

Figure& Figure::operator=(char\* p)

{

pos[0] = p[0]; pos[1] = p[1];

return(\*this);

}

int Figure::operator==(char\* p)

{

return((pos[0] == p[0]) && (pos[1] == p[1]));

}

int Figure::operator!=(char\* p)

{

return((pos[0] != p[0]) || (pos[1] != p[1]));

}

std::ostream& operator<<(std::ostream& out, Figure& f)

{

return out << f.isa() << f.pos[0] << f.pos[1];

}

std::istream& operator>>(std::istream& in, Figure& f)

{

char s[2];

std::cin.unsetf(std::ios::skipws);

in >> s[0] >> s[1];

in.ignore(64, '\n');

if((f.attack(s) == 0) || (f == s))

in.clear(std::ios::failbit | in.rdstate());

f = s;

return in;

}

int Figure::danger()

{

int i, j;

char s[2];

char \*mark =(char\*)".+x";

char m;

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;

m = (\*this != s) ? mark[attack(s)] : isa();

cout << m << ' ';

}

cout << i << "\n";

}

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

return(0);

**chessman.h**

#ifndef CHESSMAN

#define CHESSMAN

#include <iostream>

class Figure

{

protected:

char pos[4];

public:

Figure(char\*);

Figure() {};

virtual int attack(char\*)=0;

virtual char isa()=0;

operator char\*();

static int deskout(char\*);

int danger();

Figure& dodge(Figure\*);

int city(char\*);

Figure& operator=(char\*);

int operator==(char\*);

int operator!=(char\*);

friend std::ostream& operator<<(std::ostream&, Figure&);

friend std::istream& operator>>(std::istream&, Figure&);

};

#endif

**chesslib.h**

#ifndef CHESSLIB

#define CHESSLIB

#include "chessman.h"

class Knight : virtual public Figure

{

public:

Knight(char\* p) : Figure(p) {};

Knight() {};

char isa() { return 'K'; };

int attack(char\*);

};

class Queen : virtual public Figure

{

public:

Queen(char\* p) : Figure(p) {};

Queen() {};

char isa() { return 'Q'; };

int attack(char\*);

};

class Amazon : public Knight, public Queen

{

public:

Amazon(char\* p) : Figure(p) {};

Amazon() {};

char isa() { return 'A'; };

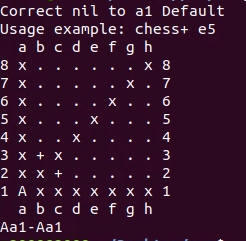
int attack(char\*);

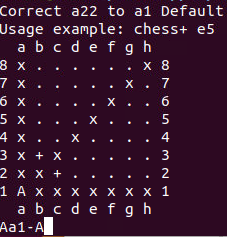
};

#endif

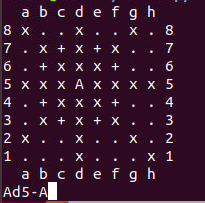
**Тесты**

**Ввод Вывод**

Пустые данных



а22

d5

**Результат**

Шахматная доска, положение фигуры.

**Списки литературы**

1)Волосатова Т. М., Родионов С. В. «Объектно-ориентированное программирование на С++» 2)http://bigor.bmstu.ru/?cnt/?doc=VU/base.cou