АНО ПО “ИТ ХАБ”  
  
  
  
  
  
  
  
**Локальная разработка. Линейные алгоритмы**

Работу сделали студенты группы 3ИБ2

Мастов Б.И. , Романова А. А. , Пикулин И. А.

Работу принял: Сухов А.А.  
Москва 2024

Оглавление

[Задача: 3](#_Toc157698747)

[Вербальная модель решения: 3](#_Toc157698748)

[Математическая модель решения: 3](#_Toc157698749)

[Блок-схема: 5](#_Toc157698750)-6

[Проверка: 7](#_Toc157698751)

[Заключение: 7](#_Toc157698752)

[Выводы: 7](#_Toc157698753)

Задача:  
На шахматной доске стоят черный король и три белые ладьи (ладья бьет по горизонтали и вертикали).

Вербальная модель решения:  
Нужно определить угрожают ли ладьи королю  
Для это нужно:  
1) Проверить положение на доске короля и ладей по горизонтали  
2) Проверить положение на доске короля и ладей по вертикали

Математическая модель решения:  
Ладья может угрожать королю по горизонтали, если y\_r1 = y\_k, y\_r2 = y\_k или y\_r3 = y\_k.

Ладья может угрожать королю по вертикали, если x\_r1 = x\_k, x\_r2 = x\_k или x\_r3 = x\_k.  
  
Код:  
#include <iostream>

#include <cmath>

// Функция для вывода шахматной доски

void printBoard(int kingX, int kingY, int rook1X, int rook1Y, int rook2X, int rook2Y, int rook3X, int rook3Y) {

std::cout << " a b c d e f g h" << std::endl;

for (int i = 8; i >= 1; i--) {

std::cout << i << " ";

for (int j = 1; j <= 8; j++) {

if ((kingX == j && kingY == i)) {

std::cout << "K ";

}

else if ((rook1X == j && rook1Y == i) || (rook2X == j && rook2Y == i) || (rook3X == j && rook3Y == i)) {

std::cout << "R ";

}

else {

std::cout << ". ";

}

}

std::cout << i << std::endl;

}

std::cout << " a b c d e f g h" << std::endl;

}

// Функция для проверки нахождения фигуры в пределах доски

bool isOnBoard(int x, int y) {

return (x >= 1 && x <= 8 && y >= 1 && y <= 8);

}

// Функция для проверки угрозы королю от каждой ладьи

void threatToKing(int kingX, int kingY, int rookX, int rookY) {

if (kingX == rookX && kingY == rookY) {

std::cout << "king is on the one field with rook." << std::endl;

}

else if (rookX == kingX || rookY == kingY) {

std::cout << "Threat from rook." << std::endl;

}

else {

std::cout << "no threat from rook." << std::endl;

}

}

// Функция для проверки, перекрывает ли одна ладья другую

bool isCovered(int rook1X, int rook1Y, int rook2X, int rook2Y, int rook3X, int rook3Y) {

return ((rook1X == rook2X && rook1Y == rook2Y) || (rook1X == rook3X && rook1Y == rook3Y) || (rook2X == rook3X && rook2Y == rook3Y));

}

int main() {

int kingX, kingY, rook1X, rook1Y, rook2X, rook2Y, rook3X, rook3Y;

// Ввод позиций короля и трех ладей

std::cout << "Pos korolya (x y): ";

std::cin >> kingX >> kingY;

std::cout << "pos rook 1 (x y): ";

std::cin >> rook1X >> rook1Y;

std::cout << "pos rook 2 (x y): ";

std::cin >> rook2X >> rook2Y;

std::cout << "pos rook 3 (x y): ";

std::cin >> rook3X >> rook3Y;

// Проверка нахождения фигур в пределах доски

if (!isOnBoard(kingX, kingY) || !isOnBoard(rook1X, rook1Y) || !isOnBoard(rook2X, rook2Y) || !isOnBoard(rook3X, rook3Y)) {

std::cerr << "Error: Figura ne na doske." << std::endl;

return 1;

}

// Вывод шахматной доски

printBoard(kingX, kingY, rook1X, rook1Y, rook2X, rook2Y, rook3X, rook3Y);

// Проверка угрозы королю от каждой ладьи

threatToKing(kingX, kingY, rook1X, rook1Y);

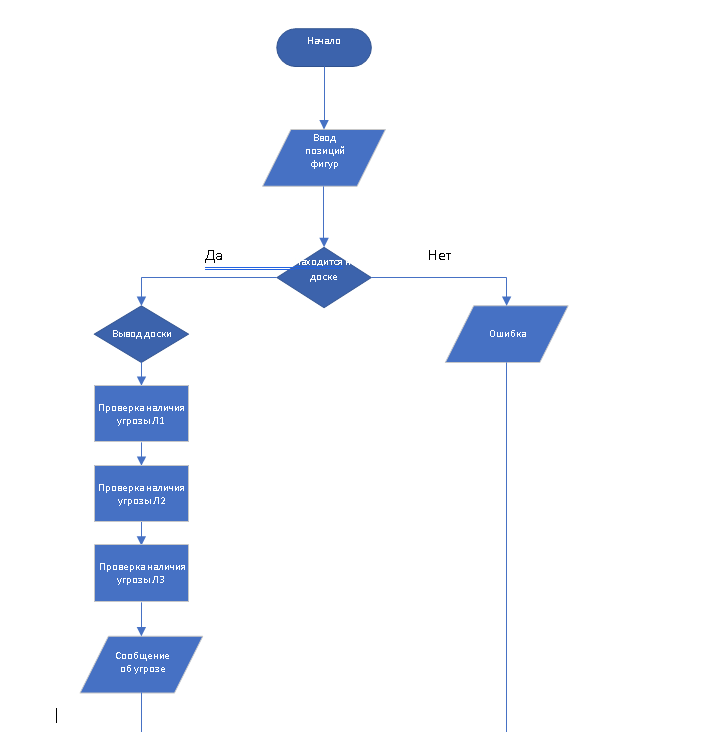
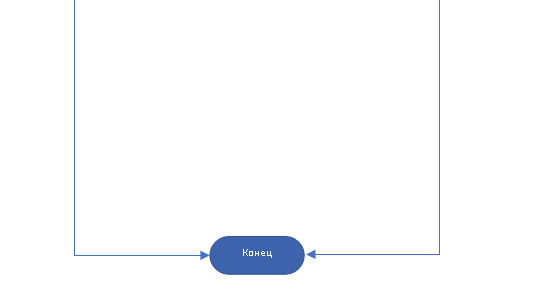
threatToKing(kingX, kingY, rook2X, rook2Y);

threatToKing(kingX, kingY, rook3X, rook3Y);

return 0;

}

Блок-схема:



Проверка:  
Возьмем положение как X: 3 Y:5

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Ладья 1(x,y) | 1. Ладья 2 (x,y) | 1. Ладья 3 (x,y) | Результат |
| 3 4 | 3 7 | 3 6 | Король под угрозой от Ладьи 1 |
| 3 5 | 2 5 | 5 5 | Король под угрозой от Ладьи 1 и 3 |
| 3 5 | 3 12 | 15 2 | Некорректный ввод данных |
| 1 2 | 1 1 | 1 4 | Нет угроз |

Заключение:  
  
Исходя из произведенных тестов, мы смогли подтвердить корректность вербальной и математической модели решения через код на C++. Количество произведенных тестов – 4, все 4 верны.

Выводы:  
После проведения тестов, мы пришли к следующим выводам.

1. Вербальная и математическая модели, которые мы использовали в коде , верны.
2. Тесты помогли найти ошибки в коде и исправить их, следовательно, тесты – неотъемлемая часть данной работы.