

Федеральное агентство по образованию
Государственное общеобразовательное учреждение высшего профессионального
образования
Пермский государственный технический университет

Лабораторная работа:

Диаграммы SADT(IDF0)

Выполнили:

студенты группы РИС-23-26

Колосов Данил Евгеньевич

Серебряков Егор Константинович

Борисов Никита Андреевич

Проверила:

доцент кафедры ИТАС

Ольга Андреевна Полякова

2024 г.

Этапы построения модели

1. Постановка задачи

Дана шахматная доска размером 8×8 . Необходимо расставить на ней 8 ферзей так, чтобы в каждой строке доски стоял 1 ферзь, при этом расстановка фигур должна быть такой, чтобы ни один ферзь не бил никакого другого.

В соответствии с шахматными правилами ферзь ходит и бьёт по вертикали, горизонтали и диагонали.

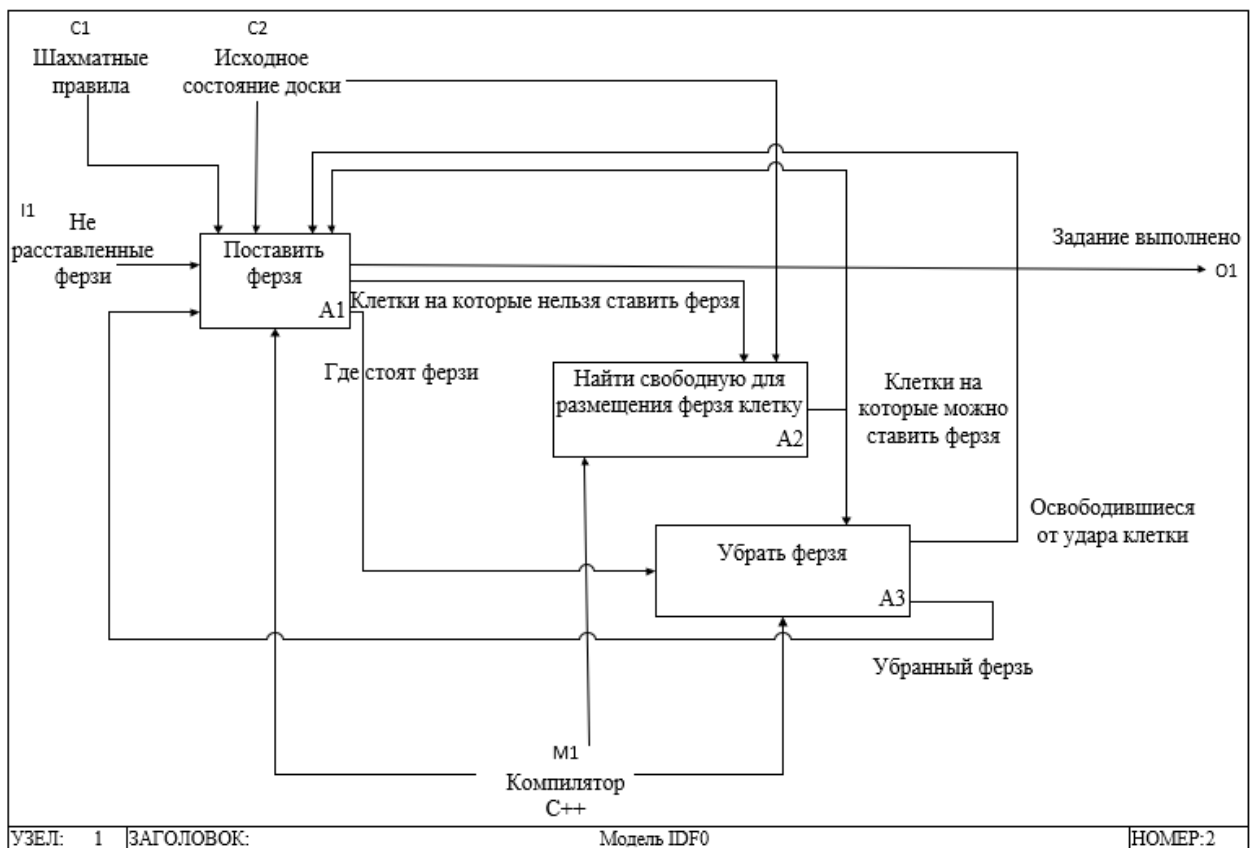
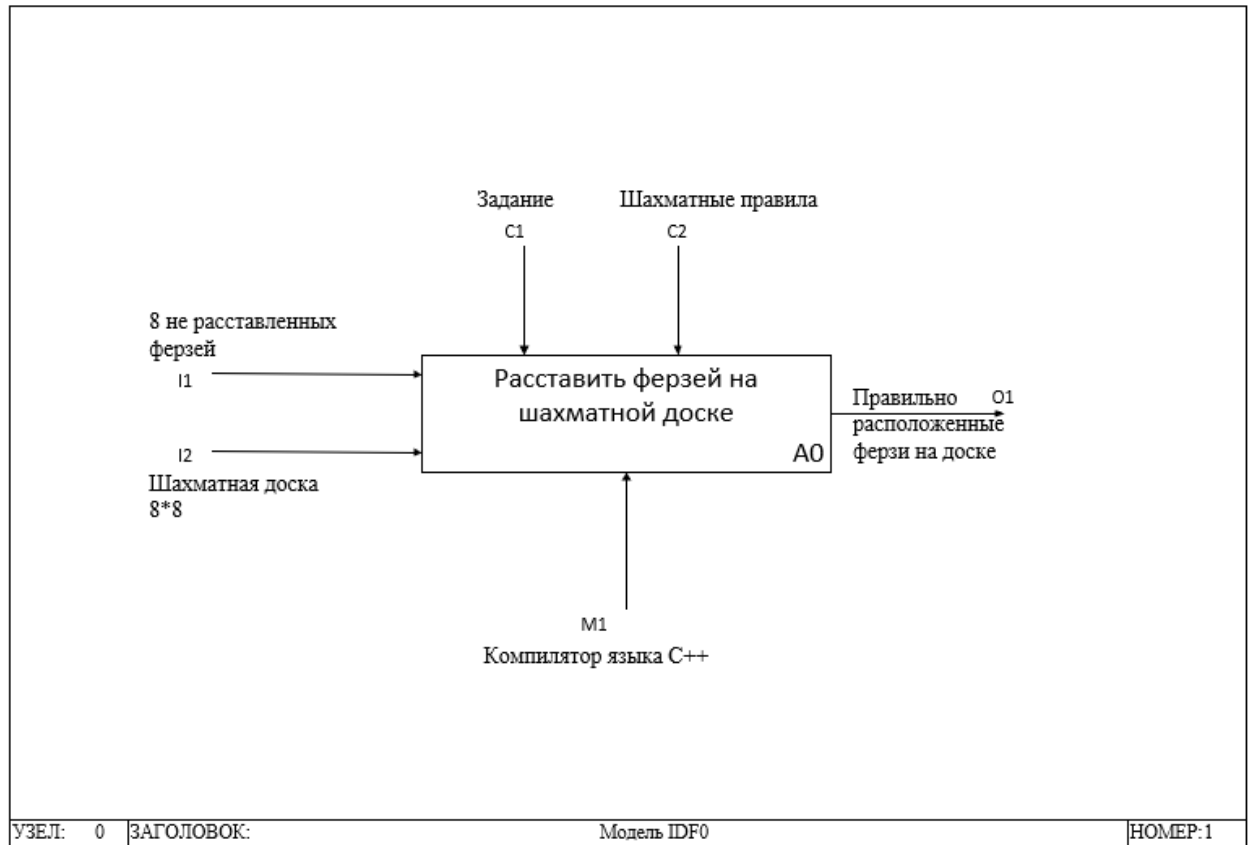
2. Средства реализации задачи

Решение задачи будет выполнено с помощью алгоритма, составленного на языке C++. Алгоритм будет перебирать все возможные значения расстановки ферзей, начиная ставить их в первую встретившуюся в строке шахматной доски клетку, которая удовлетворяет условию задачи.

3. Определение приоритетов блоков модели SADT

В приоритете будет непосредственная установка фигуры на шахматную доску. Далее, если ещё не все ферзи находятся на доске, то рассматриваются оставшиеся свободными для размещения фигур клетки, если их нет, то следующий блок отвечает за удаление последнего поставленного ферзя. В следующий раз этот ферзь будет стоять на другом, доступном ему месте.

Диаграмма SADT



Алгоритм решения задачи на языке C++

```
#include <iostream>
```

```
using namespace std;
```

```
int board[8][8]; // Шахматная доска (размерность 8x8)
```

```
void showBoard() { // Функция для визуализации изменений на доске
```

```
    for (int i = 0; i <= 7; ++ i) {
```

```
        for (int j = 0; j <= 7; ++ j) {
```

```
            if (board[i][j] >= 0) cout << " " << board[i][j] << " ";
```

```
            else cout << board[i][j] << " ";
```

```
        }
```

```
        cout << endl;
```

```
    }
```

```
    // Для удобства просмотра в консоли по шагам
```

```
    cout << "Enter to continue...";
```

```
    string pause;
```

```
    getline(cin, pause);
```

```
}
```

```
void setQueen(int i, int j) { // Функция принимает на вход ту клетку, куда можно ставить ферзя
```

```
    for (int x = 0; x <= 7; ++ x) { // значения x перебираются 8 раз потому что этого достаточно, чтобы пройти по всей доске в любом направлении
```

// Действия в цикле служат для обозначения клеток под боем по разным направлениям

// Во все клетки под боем надо добавить 1

++ board[x][j]; // По столбцу сверху вниз

++ board[i][x]; // По строке слева вправо

int foo; // Задаёт направление двух диагоналей так, что они проходят через ферзя и их координата по строке будет симметрична относительно ферзя

foo = j - i + x;

if (foo >= 0 && foo <= 7) ++ board[x][foo]; // Сверху вниз по диагонали, направленной вправо

foo = j + i - x;

if (foo >= 0 && foo <= 7) ++ board[x][foo]; // Сверху вниз по диагонали, направленной влево

}

board[i][j] = -1; // Ферзь поставлен

showBoard();

}

void resetQueen(int i, int j) { // На вход функции принимается клетка, в которой на данный момент стоит ферзь

for (int x = 0; x <= 7; ++ x) {

// Цикл убирает все клетки под боем ферзя по разным направлениям

// Из всех таких клеток вычитается 1

-- board[x][j]; // По столбцу сверху вниз

-- board[i][x]; // По строке слева вправо

int foo; // Задаёт направление двух диагоналей так, что они проходят через ферзя и их координата по строке будет симметрична относительно ферзя

foo = j - i + x;

if (foo >= 0 && foo <= 7) -- board[x][foo]; // Сверху вниз по диагонали, направленной вправо

foo = j + i - x;

if (foo >= 0 && foo <= 7) -- board[x][foo]; // Сверху вниз по диагонали, направленной влево

}

board[i][j] = 0; // Ферзь был убран

showBoard();

}

bool tryQueen(int i) { // Функция принимает на вход индекс строки

bool result = false; // Флажок, знаменующий конец расстановки

for (int j = 0; j <= 7; ++j) { // Осуществляется проход по всем клеткам строки

if (board[i][j] == 0) { // Проверка на пустую клетку (если -1, то в ней ферзь, если > 0, то под боем)

setQueen(i, j); // Установка ферзя в пустую клетку

if (i == 7) result = true; // Когда индекс строки будет равен 7, это значит, что все ферзи были успешно расставлены

else {

result = tryQueen(i + 1); // Показатель возможности постановки следующего ферзя

```
        if (result == false) resetQueen(i, j); // Если следующего  
        ферзя нельзя поставить, меняется позиция текущего
```

```
    }
```

```
    }
```

```
}
```

```
return result;
```

```
}
```

```
int main() {
```

```
    for (int i = 0; i <= 7; ++ i) { // Создаём пустую доску,  
    заполненную нулями
```

```
        for (int j = 0; j <= 7; ++ j) {
```

```
            board[i][j] = 0;
```

```
        }
```

```
    }
```

```
    tryQueen(0); // Начинаём расставлять ферзей
```

```
    return 0;
```