|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ** |
| **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** |
| **«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»** |
|  |
| Институт информационных технологий и управления в технических системах |
| (полное название института) |
|  |
| кафедра «Информационные системы» |
| (полное название кафедры) |

**ОТЧЁТ**

по лабораторной работе №4

на тему«МЕТОД АНАЛИЗА ИЕРАРХИЙ»

по дисциплине **«**Основы системного анализа»

Вариант 3

Выполнил

студент ИИТУТС

группы ИС/б-18-2-о

Радыгина Екатерина

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | |
|  | (должность, учёная степень преподавателя) | | | | | | | | |
|  |  | | | | | | | | |
|  | (ФИО преподавателя) | | | | | | | | |
|  | « |  | » |  |  |  | 20 | 20 | г. |
|  |  | | | | | | | | |
|  | (оценка) | | | | | | | | |
|  |  | | | | | | | | |
|  |  | | | | | | | | |

Севастополь 2020

4.1 Цель работы

Углубление теоретических знаний в области системного анализа, приобретение навыков создания и описания иерархических структур, а также изучение понятий цель, критерий, альтернатива.

4.2 Вариант задания

Вариант 3

Фирме нужно купить оборудование для отопления помещений. Нужно выбрать из возможных альтернатив

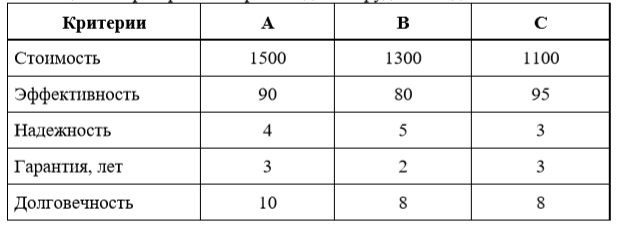


Рисунок 4.1 – Критерии альтернатив для оборудования для отопления

4.3 Ход работы

4.3.1 Решим задание методом МАИ вручную (рисунок 4.2).

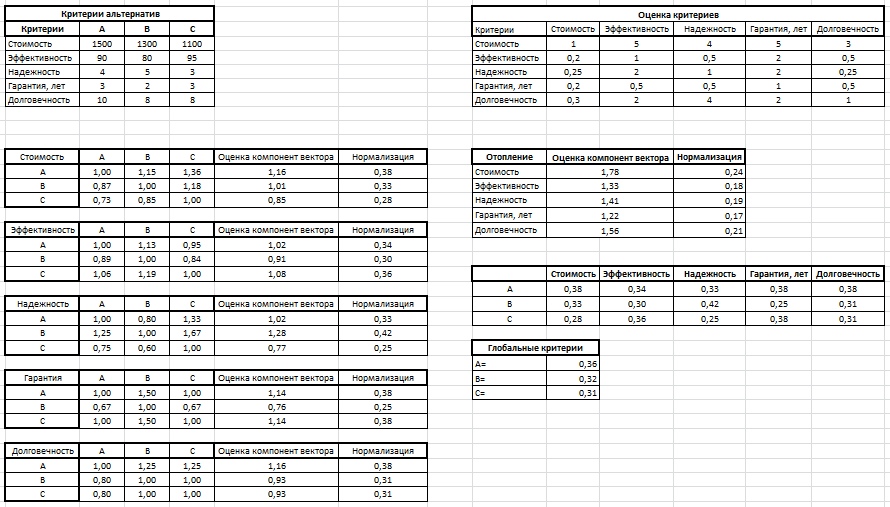


Рисунок 4.2 ­ ­– Решение с помощью метода МАИ в Excel

4.3.2 Напишем программу, которая решает МАИ задачу любой размерности.

#include <iostream>

#include <math.h>

using namespace std;

/\*------------------------------Функция создания матрицы-------------------------------\*/

float \*\*createMatrix(int row, int col){

float\*\* m = new float\* [row];

for (int i = 0; i < row; i++) m[i] = new float[col];

cout << "Введите " << row \* col << " элементов матрицы: " << endl;

for(int i = 0; i < row; i++)

for(int j = 0; j < col; j++)

cin >> m[i][j];

return m;

}

/\*-------------------------Функция создания матрицы сравнения--------------------------\*/

float \*\*priorityCriteria(float \*\*matr, int quant\_alt, int curr\_crit){

float\*\* m = new float\* [quant\_alt];

for(int i = 0; i < quant\_alt; i++) m[i] = new float[quant\_alt];

for(int i = 0; i < quant\_alt; i++)

for(int j = 0; j < quant\_alt; j++)

m[i][j] = matr[curr\_crit][i]/matr[curr\_crit][j];

return m;

}

/\*--------------------------Функция подсчёта приоритетов-------------------------------\*/

float \*calcultPriority(float \*\*m, int quant\_alt){

float sum = 0;

float\* vec = new float[quant\_alt];

for(int i = 0; i < quant\_alt; i++){

float proizv = 1;

for(int j = 0; j < quant\_alt; j++)

proizv = proizv\*m[i][j];

vec[i] = pow(proizv,(1.0/quant\_alt));

sum += vec[i];

}

for(int i = 0; i < quant\_alt; i++)

vec[i] = vec[i] / sum; //нормализация

return vec;

}

/\*-----------------------Функция расчёта глобальных приоритетов------------------------\*/

float\* calcultGlobalPriority(float\*\* m, float\* v, int quant\_crit, int quant\_alt){

float\* global\_prior = new float[quant\_alt];

for (int i = 0; i < quant\_alt; i++) global\_prior[i] = 0;

for (int i = 0; i < quant\_alt; i++)

for (int j = 0; j < quant\_crit; j++)

global\_prior[i] += m[j][i]\*v[j];

cout << "Вес альтернатив А, В и С соответственно: ";

for (int i = 0; i < quant\_alt; i++)

cout << \*&global\_prior[i] << ' ';

return global\_prior;

}

/\*---------------------------------Основная программа----------------------------------\*/

int main(){

setlocale(LC\_ALL, "Rus");

int quant\_crit; //количество критериев

int quant\_alt; //количество альтернатив

cout << "Введите количество критериев: "; cin >> quant\_crit;

cout << "Введите количество альтернатив: "; cin >> quant\_alt;

float\*\* crit\_alt; //Матрица критериев альтернатив

float\*\* eval\_crit; //Матрица оценок критериев

float\*\* cost, \*\* efficiency,

\*\* reliability, \*\* guarantee,

\*\* durability; //Матрицы сравнения критериев

float\*\* priority; //Матрица приоритетов

float\* prioir\_eval; //Вектор приоритетов оценок

float\* global\_crit; //Вектор глобальных приоритетов

cout << "Введите матрицу критериев" << endl;

crit\_alt = createMatrix(quant\_crit, quant\_alt);

cout << "Введите матрицу оценок" << endl;

eval\_crit = createMatrix(quant\_crit, quant\_crit);

//Рассчет приоритетов критериев

cost = priorityCriteria(crit\_alt, quant\_alt, 0);

efficiency = priorityCriteria(crit\_alt, quant\_alt, 1);

reliability = priorityCriteria(crit\_alt, quant\_alt, 2);

guarantee = priorityCriteria(crit\_alt, quant\_alt, 3);

durability = priorityCriteria(crit\_alt, quant\_alt, 4);

prioir\_eval = calcultPriority(eval\_crit, quant\_crit);

priority = new float\* [quant\_crit];

priority[0] = calcultPriority(cost, quant\_alt);

priority[1] = calcultPriority(efficiency, quant\_alt);

priority[2] = calcultPriority(reliability, quant\_alt);

priority[3] = calcultPriority(guarantee, quant\_alt);

priority[4] = calcultPriority(durability, quant\_alt);

//Итог

global\_crit = calcultGlobalPriority(priority, prioir\_eval, quant\_crit, quant\_alt);

/\*-------------------------------Очистка памяти------------------------------------\*/

for (int i = 0; i < quant\_crit; i++) delete[] crit\_alt[i];

delete[] crit\_alt;

for (int i = 0; i < quant\_crit; i++) delete[] eval\_crit[i];

delete[] eval\_crit;

for (int i = 0; i < quant\_alt; i++) delete[] cost[i];

delete[] cost;

for (int i = 0; i < quant\_alt; i++) delete[] efficiency[i];

delete[] efficiency;

for (int i = 0; i < quant\_alt; i++) delete[] reliability[i];

delete[] reliability;

for (int i = 0; i < quant\_alt; i++) delete[] guarantee[i];

delete[] guarantee;

for (int i = 0; i < quant\_alt; i++) delete[] durability[i];

delete[] durability;

for (int i = 0; i < quant\_crit; i++) delete[] priority[i];

delete[] priority;

delete[] prioir\_eval;

delete[] global\_crit;

return 0;

}

4.3.3 Результат работы программы (рисунок 4.3).

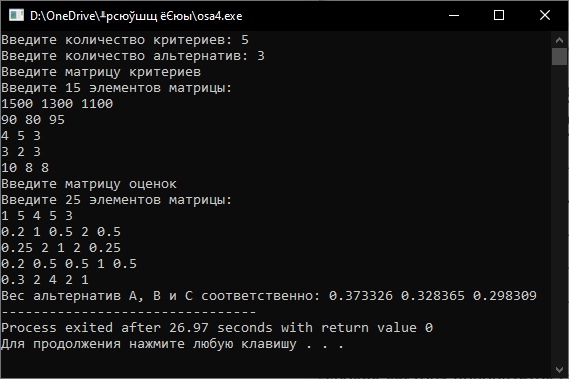


Рисунок 4.3 – Выполнение программы

**Вывод**

В ходе данной работы были получены теоретические знания в области системного анализа, приобретены навыки создания и описания иерархических структур, изучены понятия цель, критерий, альтернатива. Была написана программа, которая решает МАИ задачу любой размерности.

Исходя из расчетов, можно сделать вывод, что альтернатива А является наиболее оптимальной, так как имеет наибольший вес. Не рекомендуется приобретать альтернативу С, так как она имеет наименьший вес.