Министерство науки и высшего образования РФ

Севастопольский государственный университет

Кафедра информатики и управления в технических системах

ОТЧЕТ

по лабораторной работе №4

МЕТОД АНАЛИЗА ИЕРАРХИЙ

по дисциплине «Основы системного анализа»

Выполнил:

Студент группы ИВТ/б 22-о

Черняев Н.Г.

Проверил:

Токарев А.И.

г. Севастополь 2019

**Цель работы**

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

**Вариант №13(1)**

Решить задачу методом МАИ вручную. Написать программу, которая решает МАИ задачу любой размерности.

Предприятие для проектируемых изделий должно выбирать операционную систему по заданным критериям указанных в баллах.

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерии** | **Windows 7** | **Windows 8** | **Windows 10** |
| Стоимость, балл | 5 | 7 | 5 |
| Визуальный интерфейс | 7 | 8 | 6 |
| Надежность | 5 | 5 | 9 |
| Разрядность кода ОС | 16 | 32 | 64 |

**Ход работы**

Составление матрицы попарных сравнений для уровня 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Общее удовлетворение ОС | 1 | 2 | 3 | 4 |
| Стоимость, балл(1) | 1 | 5 | 4 | 5 |
| Визуальный интерфейс(2) | 1/5 | 1 | 1/2 | 2 |
| Надежность(3) | 1/4 | 2 | 1 | 2 |
| Разрядность кода ОС(4) | 1/5 | 1/2 | 1/2 | 1 |

Составление матрицы попарных сравнений для уровня 3:

Сравнение вариантов с точки зрения стоимости:

|  |  |  |  |
| --- | --- | --- | --- |
| Стоимость | Windows 7 | Windows 8 | Windows 10 |
| Windows 7 | 1 | 5/7 | 5/5 |
| Windows 8 | 7/5 | 1 | 7/5 |
| Windows 10 | 5/5 | 5/7 | 1 |

Сравнение вариантов с точки зрения визуального интерфейса:

|  |  |  |  |
| --- | --- | --- | --- |
| Визуальный интерфейс | Windows 7 | Windows 8 | Windows 10 |
| Windows 7 | 1 | 7/8 | 7/6 |
| Windows 8 | 8/7 | 1 | 8/6 |
| Windows 10 | 6/7 | 6/8 | 1 |

Сравнение вариантов с точки зрения надежности:

|  |  |  |  |
| --- | --- | --- | --- |
| Надежность | Windows 7 | Windows 8 | Windows 10 |
| Windows 7 | 1 | 5/5 | 5/9 |
| Windows 8 | 5/5 | 1 | 5/9 |
| Windows 10 | 9/5 | 9/5 | 1 |

Сравнение вариантов с точки зрения разрядности кода ОС:

|  |  |  |  |
| --- | --- | --- | --- |
| Разрядность кода ОС | Windows 7 | Windows 8 | Windows 10 |
| Windows 7 | 1 | 16/32 | 16/64 |
| Windows 8 | 32/16 | 1 | 32/64 |
| Windows 10 | 64/16 | 64/32 | 1 |

Синтез приоритетов для уровня 2:

|  |  |
| --- | --- |
| Общее удовлетворение ОС | Вектор приоритетов |
| Стоимость, балл | 0.6 |
| Визуальный интерфейс | 0.13 |
| Надежность | 0.19 |
| Разрядность кода ОС | 0.08 |

Синтез приоритетов для уровня 3:

Для 3 уровня вектора приоритетов считаются аналогичным образом.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант решения | Стоимость, балл | Визуальный интерфейс | Надежность | Разрядность кода ОС |
| Windows 7 | 0.29 | 0.33 | 0.26 | 0.14 |
| Windows 8 | 0.42 | 0.38 | 0.26 | 0.29 |
| Windows 10 | 0.29 | 0.29 | 0.48 | 0.57 |

Вычислим глобальные приоритеты. Для этого локальные приоритеты уровня 3 перемножаются на приоритеты соответствующих критериев уровня 2 и суммируются.

На основе проведенных вычислений вариант выбора Windows 8 получает наивысший вес и, следовательно, является наиболее оптимальным.

**Текст программы**

**import java.math.\*;**

**import java.util.\*;**

**public class Main {**

**public static void main (String[] args){**

**Solver ob1 = new Solver();**

**Scanner in = new Scanner(System.in);**

**System.out.println("Enter amount of criteria");**

**ob1.setRow(in.nextInt());**

**System.out.println("Enter amount of options");**

**ob1.setColumn(in.nextInt());**

**System.out.println("Enter matr");**

**ob1.setMatr();**

**System.out.println("Enter expertmatr");**

**ob1.setExpertmatr();**

**System.out.println(ob1.solution());**

**}**

**}**

**class Solver {**

**private int row,column;**

**private float[][] matr = new float[15][15];**

**private float[][] expertmatr = new float[15][15];**

**public void setColumn(int column) {**

**this.column = column;**

**}**

**public void setRow(int row) {**

**this.row = row;**

**}**

**public void Showmatr() {**

**for (int i = 0; i <= this.row; i++){**

**for (int j = 0; j <= this.column; j++) {**

**System.out.print(this.matr[i][j] + " ");**

**}**

**System.out.println();**

**}**

**}**

**public void setMatr(float[][] matr) {**

**this.matr = matr;**

**}**

**public void setExpertmatr(float[][] expertmatr) {**

**this.expertmatr = expertmatr;**

**}**

**public void setMatr() {**

**Scanner in = new Scanner(System.in);**

**for (int i = 0; i < this.row; i++){**

**System.out.println("Enter " + (i+1) + " line");**

**for (int j = 0; j < this.column; j++){**

**this.matr[i][j] = in.nextFloat();**

**}**

**}**

**}**

**public void setExpertmatr() {**

**Scanner in = new Scanner(System.in);**

**for (int i = 0; i < this.row; i++){**

**System.out.println("Enter " + (i+1) + " line");**

**for (int j = 0; j < this.row; j++){**

**this.expertmatr[i][j] = in.nextFloat();**

**}**

**}**

**}**

**private float avggeom(float size, float[] matr){**

**float result = 1;**

**for (int i = 0; i < size; i++) {**

**result \*= matr[i];**

**}**

**return (float) Math.pow(result, 1/size);**

**}**

**public String solution(){**

**float[] lvl2 = solutionlvl2();**

**float[][] lvl3 = solutionlvl3();**

**ArrayList<Float> koef = new ArrayList<Float>(this.column);**

**for (int i = 0; i < this.column; i++) {**

**float tmp = 0;**

**for (int j = 0; j < this.row; j++) {**

**tmp += lvl3[i][j] \* lvl2[j];**

**}**

**koef.add(tmp);**

**}**

**float maxkoef = 0;**

**for (int i = 0; i < koef.size(); i++) {**

**maxkoef = Math.max(maxkoef, koef.get(i));**

**}**

**return "The best option is " + (koef.indexOf(Collections.max(koef))+1) + " K = " + Collections.max(koef);**

**}**

**private float[] solutionlvl2 (){**

**float[] vectpriorlvl2 = new float[this.row];**

**float[] tmp = new float[row];**

**float avgtmp = 0;**

**for (int i = 0; i < this.row; i++) {**

**tmp[i] = avggeom(this.row,expertmatr[i]);**

**avgtmp += tmp[i];**

**}**

**for (int i = 0; i < this.row; i++) {**

**vectpriorlvl2[i] = tmp[i]/avgtmp;**

**}**

**return vectpriorlvl2;**

**}**

**private float[][] solutionlvl3 (){**

**float[][] vectpriorlvl3 = new float[this.column][this.row];**

**for (int i = 0; i < this.row; i++) {**

**float[][] tmpmatr = new float[this.column][this.column];**

**for (int j = 0; j < this.column; j++) {**

**for (int k = 0; k < this.column; k++) {**

**tmpmatr[j][k] = this.matr[i][j]/this.matr[i][k];**

**}**

**}**

**float[] tmp = new float[row];**

**float avgtmp = 0;**

**for (int j = 0; j < this.column; j++) {**

**tmp[j] = avggeom(this.column, this.matr[j]);**

**avgtmp += tmp[j];**

**}**

**for (int j = 0; j < this.column; j++) {**

**vectpriorlvl3[j][i] = tmp[j]/avgtmp;**

**}**

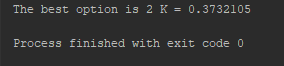
**}**

**return vectpriorlvl3;**

**}**

**}**

**Результат работы программы**



Результат работы программы практически полностью совпадает с результатом рассчетов вручную.

**Выводы**

В данной лабораторной работе были углублены теоретические знания в области системного анализа. В ходе работы были приобретены навыки создания и описания иерархических структур, а также изучены следующие понятия: цель, критерий, альтернатива.