

Институт информационных технологий
Кафедра «Информационные системы»

ОТЧЕТ
по лабораторной работе № 4
«Метод анализа иерархий»
по дисциплине «Методы системного анализа и проектирования
информационных систем»

Выполнил студент группы ИС/б-22-1-о

Донец Н.О.

Проверил доцент

Кудрявченко И.В.

Севастополь

2024

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

Решить задачу методом МАИ вручную: Предприятие для проектируемых изделий должно выбирать операционную систему по заданным критериям указанных в баллах. Нужно выбрать из возможных альтернатив (рисунок 4.1). Написать программу на языке программирования python (или на другом языке) которая решает МАИ задачу любой размерности. Проверить на примере данных по своему варианту.

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

Рисунок 4.1 – Вариант задания

4.3 Ход выполнения работы

Было составлено иерархическое представление задачи, что показано на рисунке 4.2.

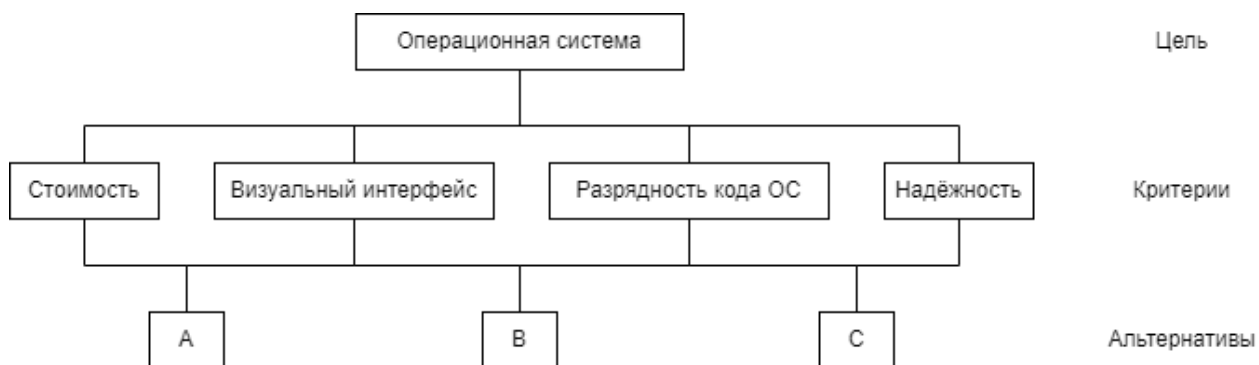


Рисунок 4.2 – Иерархическое представление задачи

В таблице 4.1 представлены оценки экспертов для приоритетов характеристик.

Таблица 4.1 – Экспертные оценки приоритетов характеристик

$W_{21} = 1/3$		
$W_{31} = 3$	$W_{32} = 4$	
$W_{41} = 1/2$	$W_{42} = 2$	$W_{43} = 1/4$

Была составлена матрица попарных сравнений, которая отражена в таблице 4.2, чтобы далее установить приоритеты критериев и оценить по критериям все альтернативы.

Таблица 4.2 – Матрица попарных сравнений

Общее удовлетворение оборудованием	1	2	3	4
Стоимость (1)	1	3	1/3	2
Визуальный интерфейс (2)	1/3	1	1/4	1/2
Разрядность кода ОС (3)	3	4	1	4
Надёжность (4)	1/2	2	1/4	1

Затем были установлены приоритеты критериев и оценена каждая из альтернатив по критериям с помощью формулы среднего арифметического, а затем их сумма:

$$w_1 = \sqrt[4]{C_{11} * C_{12} * C_{13} * C_{14}}$$

$$w_{01} = \sqrt[4]{1 * 3 * \frac{1}{3} * 2} = 1,19$$

$$w_{02} = \sqrt[4]{\frac{1}{3} * 1 * \frac{1}{4} * \frac{1}{2}} = 0,45$$

$$w_{03} = \sqrt[4]{3 * 4 * 1 * 4} = 2,63$$

$$w_{04} = \sqrt[4]{\frac{1}{2} * 2 * \frac{1}{4} * 1} = 0,7$$

$$w_0 = 1,19 + 0,45 + 2,63 + 0,7 = 4,97$$

В таблице 4.3 показан вектор приоритетов для уровня 2.

Таблица 4.3 – Вектор приоритетов для уровня 2

Общее удовлетворение оборудованием	Вектор приоритетов
Стоимость	$x_1 = w_{01} / w_0 = 0,24$
Визуальный интерфейс	$x_2 = w_{02} / w_0 = 0,09$
Разрядность кода	$x_3 = w_{03} / w_0 = 0,53$
Надёжность	$x_4 = w_{04} / w_0 = 0,14$

Далее были составлены пять матриц для третьего уровня по отношению к критериям второго уровня, что показано в таблицах 4.4 – 4.7.

Таблица 4.4 – Сравнение вариантов с точки зрения стоимости

Стоимость	А	В	С
А	1	5/7	1
В	7/5	1	7/5
С	1	5/7	1

Таблица 4.5 – Сравнение вариантов с точки зрения визуального интерфейса

Визуальный интерфейс	А	В	С
А	1	7/8	7/6
В	8/7	1	8/6
С	6/7	6/8	1

Таблица 4.6 – Сравнение вариантов с точки зрения надёжности

Надёжность	А	В	С
А	1	1	5/9
В	1	1	5/9
С	9/5	9/5	1

Таблица 4.7 – Сравнение вариантов с точки зрения разрядности кода ОС

Разрядность кода ОС	А	В	С
А	1	1/2	1/4
В	2	1	1/2
С	4	2	1

По примеру из предыдущего пункта для всех полученных таблиц были рассчитаны вектора приоритетов. Для таблицы 4.4:

$$w_{11} = \sqrt[3]{1 * \frac{5}{7} * 1} = 0,9$$

$$w_{12} = \sqrt[3]{\frac{7}{5} * 1 * \frac{7}{5}} = 1,25$$

$$w_{13} = \sqrt[3]{1 * \frac{5}{7} * 1} = 0,9$$

$$w_1 = 0,9 + 1,25 + 0,9 = 3,05$$

Для таблицы 4.5:

$$w_{21} = \sqrt[3]{1 * \frac{7}{8} * \frac{7}{6}} = 1$$

$$w_{22} = \sqrt[3]{\frac{8}{7} * 1 * \frac{8}{6}} = 1,15$$

$$w_{23} = \sqrt[3]{\frac{6}{7} * \frac{6}{8} * 1} = 0,86$$

$$w_2 = 1 + 1,15 + 0,86 = 3,01$$

Для таблицы 4.6:

$$w_{41} = \sqrt[3]{1 * 1 * \frac{5}{9}} = 0,82$$

$$w_{42} = \sqrt[3]{1 * 1 * \frac{5}{9}} = 0,82$$

$$w_{43} = \sqrt[3]{\frac{9}{5} * \frac{9}{5} * 1} = 1,48$$

$$w_4 = 0,82 + 0,82 + 1,48 = 3,12$$

Для таблицы 4.7:

$$w_{31} = \sqrt[3]{1 * \frac{1}{2} * \frac{1}{4}} = 0,5$$

$$w_{32} = \sqrt[3]{2 * 1 * \frac{1}{2}} = 1$$

$$w_{33} = \sqrt[3]{4 * 2 * 1} = 2$$

$$w_3 = 0,5 + 1 + 2 = 3,5$$

Матрица со всеми векторами приоритетов для 3 уровня приведена в таблице 4.9, а ее расчет в таблице 4.8.

Таблица 4.8 – Расчет вектора приоритетов для уровня 3

Вариант решения	Стоимость	Визуальный интерфейс	Разрядность кода ОС	Надёжность
А	0,9/3,05	1/3,01	0,5/3,5	0,82/3,12
В	1,25/3,05	1,15/3,01	1/3,5	0,82/3,12
С	0,9/3,05	0,86/3,01	2/3,5	1,48/3,12

Таблица 4.9 – Расчет вектора приоритетов для уровня 3

Вариант решения	Стоимость	Визуальный интерфейс	Разрядность кода ОС	Надёжность
А	0,3	0,33	0,14	0,26
В	0,41	0,38	0,29	0,26
С	0,3	0,29	0,57	0,47

Далее с помощью полученных значений были вычислены глобальные приоритеты путем перемножения локальных приоритетов уровня 3 и соответствующих критериев уровня 2 и суммирования полученных значений:

$$K_A = x_{11}x_1 + x_{12}x_2 + x_{13}x_3 + x_{14}x_4$$

$$K_B = x_{21}x_1 + x_{22}x_2 + x_{23}x_3 + x_{24}x_4$$

$$K_C = x_{31}x_1 + x_{32}x_2 + x_{33}x_3 + x_{34}x_4$$

$$K_A = 0,3*0,24 + 0,33*0,09 + 0,14*0,53 + 0,26*0,14 = 0,24$$

$$K_B = 0,41*0,24 + 0,38*0,09 + 0,29*0,53 + 0,26*0,14 = 0,3$$

$$K_C = 0,3*0,24 + 0,29*0,09 + 0,57*0,53 + 0,47*0,14 = 0,44$$

По полученным данным можно сделать вывод, что вариант С, имеющий самый наивысший вес, является более оптимальным.

4.3.5 Для решения подобных задач была написана программа на языке python, код которой показан в листинге 4.1.

Листинг 4.1 – Код программы для метода иерархий

```
import numpy as np

def calculate_geometric_means(matrix):
    return np.prod(matrix, axis=1) ** (1 / matrix.shape[1])

def normalize_vector(vector):
    return vector / np.sum(vector)

def create_comparison_matrix(data):
    size = data.shape[0]
    matrix = np.zeros((size, size))
    for i in range(size):
        for j in range(size):
            matrix[i, j] = data[i] / data[j]
    return matrix
```

```

def calculate_alternative_priorities(data):
    alternative_priorities = {}
    for key, matrix in data.items():
        alt_priority = calculate_geometric_means(matrix)
        alternative_priorities[key] = normalize_vector(alt_priority)
    return alternative_priorities

# Матрица попарных сравнений для критериев
criteria_comparison = np.array([
    [1, 3, 1/3, 2],
    [1/3, 1, 1/4, 1/2],
    [3, 4, 1, 4],
    [1/2, 2, 1/4, 1],
])

# Вычисление и нормализация средних геометрических для критериев
priority_vector =
normalize_vector(calculate_geometric_means(criteria_comparison))

# Данные ОС
alternatives = np.array([
    [5, 7, 5, 16], # Оборудование А
    [7, 8, 5, 32], # Оборудование В
    [5, 6, 9, 64]  # Оборудование С
])

# Составление матриц попарных сравнений для альтернатив
criteria = ["Стоимость", "Интерфейс", "Надежность", "Разрядность"]
matrices = {criteria[i]: create_comparison_matrix(alternatives[:, i])
for i in range(alternatives.shape[1])}

# Вычисление приоритетов для каждой альтернативы
alternative_priorities = calculate_alternative_priorities(matrices)

# Вектор приоритетов для критериев уровня 2
criteria_priorities = np.array([0.24, 0.09, 0.53, 0.14])

# Вектор приоритетов для альтернатив уровня 3

```



```

        alternative_matrix = np.array([alternative_priorities[criterion] for
criterion in criteria]).T

# Расчет итоговых оценок для альтернатив
final_scores = np.dot(alternative_matrix, criteria_priorities)

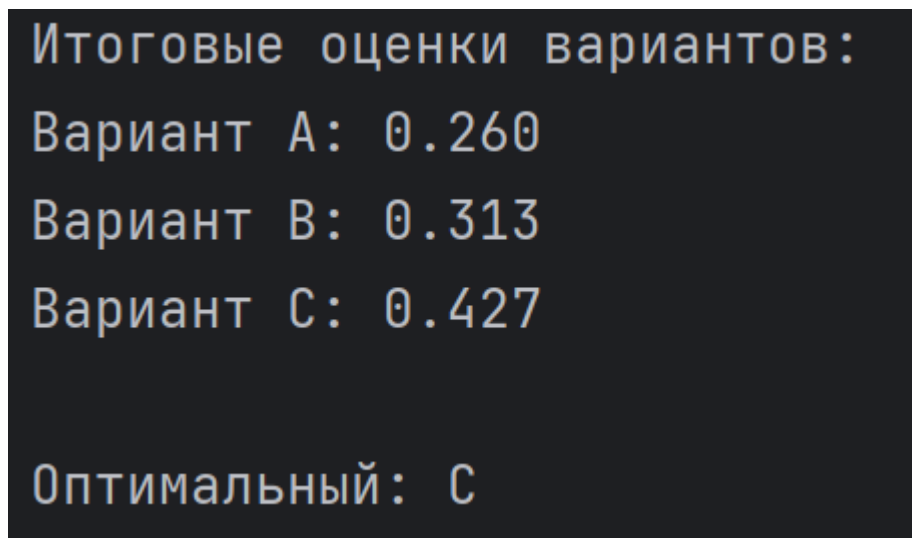
# Вывод итоговых оценок
print("Итоговые оценки вариантов:")
for i, score in enumerate(final_scores):
    print(f"Вариант {chr(65+i)}: {score:.3f}")

max_index = np.argmax(final_scores)

print(f"\nОптимальный: {chr(65+max_index)}")

```

Вычисления, полученные программным методом, совпали с вычислениями, сделанными аналитическим методом, что показано в выводе программы на рисунке 4.3.



```

Итоговые оценки вариантов:
Вариант A: 0.260
Вариант B: 0.313
Вариант C: 0.427

Оптимальный: C

```

Рисунок 4.3 – Вывод программы

Выводы

В ходе лабораторной работы был исследован метод анализа иерархий, который применяется, чтобы структурировать сложную проблему принятия решений в виде иерархии, сравнить и выполнить количественную оценку альтернативных вариантов решения. Для системы, полученной по варианту, был построена иерархическая структура. Далее были установлены приоритеты критериев и оценена каждая из альтернатив по критериям. Для количественного определения сравнительной важности факторов и проблемной ситуации была составлена матрица попарных сравнений. Затем были составлены пять матриц для третьего уровня по отношению к критериям второго уровня, для всех из которых были построены вектора приоритетов. После вычисления глобальных приоритетов был сделан вывод о том, что вариант С, имеющий самый наивысший вес, является более оптимальным. В последнюю очередь была написана программа на языке python, которая применима для решения подобных задач методом анализа иерархий. В конце выполнения лабораторной работы был написан отчет.