|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Задание по практической работе №1**

по дисциплине «Моделирование программных систем»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-06-22 | Абаренова Д.Д.  Ляхов Т.А.  . |
| **Проверил:** | Образцов В.М. |

2024 г.

**Оглавление**

[Введение 3](#_Toc159621139)

[Постановка задачи 4](#_Toc159621140)

[Ход работы 5](#_Toc159621141)

[1 Значения характеристик анализируемого ПО 5](#_Toc159621142)

[2 Реализация методов 6](#_Toc159621143)

[2.1 Метод ELECTRE 6](#_Toc159621144)

[2.2 Метод SAW 9](#_Toc159621145)

[2.3 Метод TOPSIS 10](#_Toc159621146)

[3 Выбор оптимального ПО 11](#_Toc159621147)

[Вывод 12](#_Toc159621148)

[Список источников 13](#_Toc159621149)

# **Введение**

В современном мире программное обеспечение играет ключевую роль в профессиональной сфере, обеспечивая эффективное функционирование различных организаций и предприятий. Однако, для обеспечения высокого качества работы и достижения поставленных целей необходимо проводить оценку качества программного обеспечения и осуществлять обоснованный выбор оптимального варианта.

В данной работе будет рассмотрена методика оценки качества программного обеспечения, а также представлен анализ программного обеспечения.

# **Постановка задачи**

Провести оценку качества программного обеспечения и осуществить обоснованный выбор варианта программного обеспечения, применяемого в профессиональной сфере в соответствии с указанной методикой.

В качестве программного обеспечения были выбраны сервисы для прослушивая музыки – Spotify, Apple Music, YouTube Music, Yandex Music. В качестве методов принятия управленческих решений были выбраны методы ELECTRE, TOPSIS, SAW.

# **Ход работы**

## **Значения характеристик анализируемого ПО**

В качестве характеристик для оценки программного обеспеченья выберем:

1. Защищенность
2. Восстанавливаемость
3. Удобство использования
4. Удовлетворенность
5. Стоимость

Составим таблицу оценки анализируемого программного обеспеченья

данными характеристиками. В таблице представлен вес для каждого критерия, критерии в столбцах и оцениваемые сервисы для прослушивания музыки в строках.

Таблица 1. Оценка ПО

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Веса** | 0,2 | 0,2 | 0,3 | 0,1 | 0,2 |
| **Характе- ристики**  **Мессен- джеры** | **Защищен-**  **ность** | **Восстанав-**  **ливаемость** | **Удобство использования** | **Удовлетво-**  **ренность** | **Стои- мость** |
| Spotify | 0,8 | 0,88 | 0,8 | 0,9 | 0,9 |
| Apple Music | 0,8 | 0,7 | 0,85 | 0,75 | 0,9 |
| YouTube Music | 0,9 | 0,88 | 0,5 | 0,7 | 0,7 |
| Yandex Music | 0,9 | 0,9 | 0,88 | 0,9 | 0,7 |

## **Реализация методов**

### **Метод ELECTRE**

Метод ELECTRE (Elimination and Choice Expressing Reality) — это один из методов многокритериального принятия решений, который позволяет сравнивать и классифицировать альтернативы на основе нескольких критериев. Основная идея метода ELECTRE заключается в том, что он учитывает не только количественные значения критериев, но также их взаимосвязи и предпочтения принимающей стороны.

Метод ELECTRE позволяет учитывать неопределенность и различные предпочтения принимающей стороны при принятии решений. Он широко используется в различных областях, где необходимо принимать сложные решения на основе множества критериев.

Листинг 2.1

|  |
| --- |
| def \_\_concordance\_matrix\_\_(criteria\_matrix, weigths):  # Метод составления матрицы согласия  concordance = np.zeros((criteria\_matrix.shape[0], criteria\_matrix.shape[0]))  for i in range(0, concordance.shape[0]):  for j in range(0, concordance.shape[1]):  concordance[i, j] = sum(weigths[k] for k in range(0, criteria\_matrix.shape[1]) if (criteria\_matrix[i, k] >= criteria\_matrix[j, k]))  return concordance  def \_\_discordance\_matrix\_\_(criteria\_matrix):  # Метод составления матрицы несогласия  n = criteria\_matrix.shape[0]  discordance = np.zeros((n, n))  for i in range(n):  for j in range(n):  discordance[i, j] = round(max(criteria\_matrix[j, :] - criteria\_matrix[i, :]), 2)  return discordance  def \_\_dominance\_matrix\_\_(concordance, discordance, c\_lim, d\_lim):  # Метод составления матрицы удовлетворения пороговым значения индексов согласия и несогласия  dominance = np.zeros((concordance.shape[0], concordance.shape[0]))  for i in range(0, dominance.shape[0]):  for j in range(0, dominance.shape[1]):  if concordance[i, j] >= c\_lim and discordance[i, j] <= d\_lim and i != j:  dominance[i, j] = 1  return dominance  def \_\_lim\_concordance\_list\_\_(concordance):  # Метод поиска предельных значений индекса согласия  return [round(min(x), 4) for x in concordance]  def \_\_lim\_discordance\_list\_\_(discordance):  # Метод поиска предельных значений индекса несогласия  return [round(max(x), 4) for x in discordance]  def rank\_alternatives(dominance, alternative\_names):  dominated = [(alternative\_names[i], elem) for i, elem in enumerate(dominance)]  dominated = sorted(dominated, key=lambda k: sum(k[1]), reverse=True)  info = 'Метод ELECTRE.\nРезультаты ранжирования:\n'  for ind, el in enumerate(dominated):  info += f'{ind + 1}. {el[0]}\n'  return info  def electre(criteria\_matrix, weigths, alternative\_names, c\_lim=0.45, d\_lim=0.5):  criteria\_matrix = np.array(criteria\_matrix)  concordance = \_\_concordance\_matrix\_\_(criteria\_matrix, weigths)  discordance = \_\_discordance\_matrix\_\_(criteria\_matrix)  dominance = \_\_dominance\_matrix\_\_(concordance, discordance, c\_lim, d\_lim)  print(rank\_alternatives(dominance, alternative\_names))  return concordance, discordance, dominance |

В данном коде представлены методы для использования метода ELECTRE:

1. Метод \_\_concordance\_matrix\_\_ создает матрицу согласия. Для каждой пары альтернатив вычисляется сумма весов критериев, где первая альтернатива превосходит или равна важности критерия по сравнению с другой альтернативой.

2. Метод \_\_discordance\_matrix\_\_ создает матрицу несогласия. Для каждой пары альтернатив вычисляется максимальная разница между значениями критериев.

3. Метод \_\_dominance\_matrix\_\_ создает матрицу доминирования на основе матриц согласия и несогласия, а также пороговых значений индексов согласия и несогласия.

4. Методы \_\_lim\_concordance\_list\_\_ и \_\_lim\_discordance\_list\_\_ используются для поиска предельных значений индексов согласия и несогласия соответственно.

5. Метод rank\_alternatives используется для вывода ранжирования альтернатив на основе матрицы доминирования.

Внутри самой функции electre вычисляются матрицы согласия и несогласия, затем формируется матрица доминирования, и выводится ранжирование альтернатив.

Вывод программы представлен на рисунке 1.

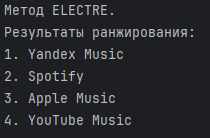


Рис. 1 – Вывод метода ELECTRE

В результате получаем, что Yandex Music на первом месте, Sportify на втором месте, Apple Music на третьем месте, YouTube Music на четвертом месте.

* 1. **Метод SAW**

Метод SAW (Simple Additive Weighting) — это один из простых методов многокритериальной оптимизации, который используется для принятия решений в условиях неопределенности. Основная идея метода SAW заключается в том, что каждый критерий оценивается по шкале от 0 до 1, где 1 обозначает наилучшее значение, а 0 - наихудшее. Затем каждый критерий умножается на его весовой коэффициент и суммируются для каждой альтернативы, чтобы получить итоговую оценку.

Напишем код для подсчёта и получения ранжированного списка анализируемого ПО.

Листинг 2.2. Метод SAW

|  |
| --- |
| def saw(criteria\_matrix, weights, alternatives\_names, criteria\_names):  # Умножение значений критериев на веса и суммирование  weighted\_matrix = criteria\_matrix \* weights  saw\_scores = np.sum(weighted\_matrix, axis=1)  # Ранжирование альтернатив  ranked\_indices = np.argsort(saw\_scores)[::-1] # Сортировка в порядке убывания |

Вывод программы представлен на рис. 2.

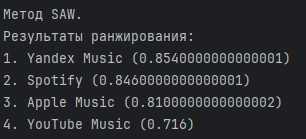


Рисунок 2 – Результат метода SAW

В итоге получится:

1 место - Yandex Music

2 место - Sportify

3 место - Apple Music

4 место – YouTube Music

* 1. **Метод TOPSIS**

Метод TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) — это метод многокритериального принятия решений, который используется для оценки альтернативных вариантов на основе их близости к идеальному и анти-идеальному решениям. Этот метод позволяет ранжировать альтернативы по уровню приоритетности и выбрать оптимальный вариант.

Процесс реализации метода TOPSIS включает следующие шаги:

1. Формулирование критериев оценки и альтернативных вариантов.

3. Определение идеального (MAX) и анти-идеального (MIN) решений.

4. Расчет расстояний от каждой альтернативы до идеального и анти-идеального решений.

5. Расчет близости каждой альтернативы к идеальному решению (ближе - лучше).

6. Ранжирование альтернатив по уровню близости к идеальному решению.

Листинг 2.3. Метод TOPSIS

|  |
| --- |
| def topsis(criteria\_matrix, weights):  # Умножение на веса  weighted\_matrix = criteria\_matrix \* weights  # Вычисление итоговых оценок для каждой альтернативы  v\_plus = np.max(weighted\_matrix, axis=0)  v\_minus = np.min(weighted\_matrix, axis=0)  # Расчет расстояний до идеального и анти-идеального решений  d\_plus = np.sqrt(((weighted\_matrix - v\_plus) \*\* 2).sum(axis=1))  d\_minus = np.sqrt(((weighted\_matrix - v\_minus) \*\* 2).sum(axis=1))  # Расчет близости к идеальному решению  closeness = d\_minus / (d\_plus + d\_minus)  # Ранжирование альтернатив  ranked\_indices = np.argsort(closeness)[::-1] # Сортировка в порядке убывания |

Вывод программы представлен на рисунке 2.

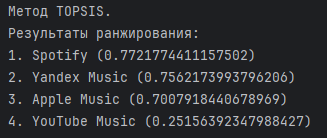


Рис. 2 – вывод метода TOPSIS

В итоге выходит: Spotify на первом месте, Yandex Music на втором месте, Apple Music на третьем месте, YouTube Music на четвертом месте

1. **Выбор оптимального ПО**

На основе результатов выбранных методов получается, что в двух из трёх результатов 1-ое место получило ПО “Yandex Music”. Из чего можно сделать вывод, что данное ПО будет самым оптимальным выбором из просмотренных. На втором месте сервис “Spotify”. На третьем месте - “Apple Music”. На четвертом месте сервис “YouTube Music”.

**Вывод**

Во время выполнения данной работы мы научились:

1. Определять критерии оценки качества программного обеспечения.

2. Устанавливать веса для каждого критерия, чтобы выразить их относительную значимость при выборе программного обеспечения.

3. Применять метод ELECTRE для вычисления матриц согласия и несогласия, а также матрицы доминирования для ранжирования альтернатив и выбора оптимального варианта программного обеспечения.

4. Принимать обоснованные решения по выбору программного обеспечения, основываясь на результате анализа методами ELECTRE, SAW и TOPSIS.

Таким образом, выполнение данной работы позволило нам приобрести навыки по проведению оценки качества программного обеспечения и принятию обоснованных решений при выборе оптимального варианта для использования в профессиональной деятельности.

# **Список источников**

1. А. В. Демидовский. Сравнительный анализ методов многокритериального принятия решений: ELECTRE, TOPSIS и ML-LDM [Электронный ресурс]: Электронная статья: Национальный Исследовательский Университет Высшая Школа Экономики, 2020. – режим доступа: URL: <https://scm.etu.ru/assets/files/2020/scm20/papers/4/234.pdf>, свободный (дата обращения 27.02.2023)
2. Катаржина Халицкая. Выбор технологий с помощью метода TOPSIS [Электронный ресурс]: Электронная статья: Белостокский технический университет, 2020. – режим доступа: URL: <https://foresight-journal.hse.ru/data/2020/03/20/1567702093/6-Халицкая-85-96.pdf>, свободный (дата обращения 27.02.2023)
3. Леонас Устинович, Зенонас Турскис, Галина Шевченко. ПРИМЕНЕНИЕ МЕТОДА SAW ДЛЯ МНОГОКРИТЕРИАЛЬНОГО СРАВНИТЕЛЬНОГО АНАЛИЗА ВАРИАНТОВ РИСКА ИНВЕСТИЦИЙ В СТРОИТЕЛЬСТВЕ [Электронный ресурс]: Электронная статья: Вильнюсский технический университет им. Гедиминаса, 2006. – режим доступа: URL: <https://www.tsi.lv/sites/default/files/editor/science/Research_journals/Tr_Tel/2006/V3/art06-2.pdf>, свободный (дата обращения 27.02.2023)