

```
if play_rect.collidepoint(event.pos):
    running = True
    play_rect.center = (320, 448)
else:
    running = False
screen.fill((0, 0, 0))
screen.blit(platform, platform_rect)
screen.blit(obstacle, obstacle_rect)
if running:
    screen.blit(ball, ball_rect)
    if move_ball(ball_rect, platform_rect):
        ball_rect.y += 2
    move_obstacle(obstacle_rect)
    move_platform(platform_rect)
pygame.display.flip()
pygame.quit()
```

В этом примере мы создаем игру, в которой мячик прыгает по платформе и избегает препятствия. Код использует библиотеку Pygame для создания экрана, загрузки изображений и обработки событий мыши. Функции `move_ball`, `move_obstacle` и `move_platform` отвечают за движение объектов на экране. Основной цикл игры проверяет события мыши и обновляет экран.

Заключение

Разработка игр на Python – это простой и эффективный способ создания игр. Python имеет большое сообщество разработчиков, большое количество библиотек и фреймворков, что делает его идеальным для разработки игр. Примеры использования Python для разработки игр включают создание 2D-игр, 3D-игр и многопользовательских игр.

Список использованных источников:

1. Разумников С.В. Алгоритм и программное обеспечение для построения агрегированных рейтингов / С.В. Разумников // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2021. – Т. 24, № 2. – С. 39–44.
2. Разумников С.В. Разработка программного обеспечения для построения агрегированных рейтингов на основе метода порогового агрегирования / С.В. Разумников // Вестник ВГУ, серия: Системный анализ и информационные технологии. – 2021. – № 2. – С. 138–152.
3. Разумников С.В. Планирование развития облачной стратегии на основе применения многокритериальной оптимизации и метода STEM / С.В. Разумников // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2020. – Т. 23, № 1. – С. 53–61.
4. Разумников С.В. Модели, алгоритмы и программное обеспечение поддержки принятия стратегических решений к переходу на облачные технологии : монография / С.В. Разумников. – Томск : Изд-во Томского политехнического университета, 2020. – 176 с.

ПРОГРАММИРОВАНИЕ МАТРИЦЫ ПОПАРНЫХ СРАВНЕНИЙ НА ЯЗЫКЕ C#

К.К. Первалов², ученик 11 класса

Научный руководитель: Разумников С.В.^{а1}, к.т.н. доц.

¹Юргинский технологический институт (филиал)

Национального исследовательского Томского политехнического университета

652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26

²МБОУ «Гимназия города Юрги», 652057, Кемеровская обл., г. Юрга, ул. Московская, 48

E-mail: ^arazumnikov@tpu.ru

Аннотация: В статье рассматривается инструмент в области принятия решений – матрица попарных сравнений. Предложен пример программной реализации на языке C#. Представлено диалоговое окно программы для проведения расчетов.

Ключевые слова: матрица попарных сравнений, программирование, язык C#.

Abstract: The article discusses a tool in the field of decision making – a matrix of pairwise comparisons. An example of software implementation in the C# language is proposed. A dialog box of the program for carrying out calculations is presented.

Keywords: pairwise comparison matrix, programming, C# language.

Введение

Матрица попарных сравнений – это важный инструмент в области принятия решений, который используется для анализа и сравнения альтернативных вариантов. Она представляет собой таблицу, в которой каждая строка и столбец соответствуют одному из альтернативных вариантов. В ячейках таблицы указываются значения, которые отражают предпочтения одного варианта по отношению к другому.

В данной статье мы рассмотрим, как можно реализовать матрицу попарных сравнений на языке программирования C#. Мы рассмотрим основные понятия и алгоритмы, которые используются для создания и обработки матрицы попарных сравнений.

Основные понятия

Матрица попарных сравнений – это таблица, в которой каждая строка и столбец соответствуют одному из альтернативных вариантов.

Предпочтения – это значения, которые указываются в ячейках матрицы попарных сравнений и отражают предпочтения одного варианта по отношению к другому.

Отношение предпочтения – это отношение, которое устанавливается между двумя вариантами на основе значений в ячейках матрицы попарных сравнений.

Алгоритмы

Алгоритмы, которые используются для создания и обработки матрицы попарных сравнений, включают:

1. Алгоритм создания матрицы попарных сравнений. Этот алгоритм используется для создания матрицы попарных сравнений на основе предпочтений.

2. Алгоритм определения отношения предпочтения. Этот алгоритм используется для определения отношения предпочтения между двумя вариантами на основе значений в ячейках матрицы попарных сравнений.

3. Алгоритм обработки матрицы попарных сравнений. Этот алгоритм используется для обработки матрицы попарных сравнений, например, для нахождения наиболее предпочтительного варианта.

Пример кода

Вот пример кода на языке программирования C#, который реализует матрицу попарных сравнений:

```
public class Matrix
{
    private int[,] matrix;
    public Matrix(int rows, int columns)
    {
        matrix = new int[rows, columns];
    }
    public void SetValue(int row, int column, int value)
    {
        matrix[row, column] = value;
    }
    public int GetValue(int row, int column)
    {
        return matrix[row, column];
    }
    public int GetMaxValue(int row)
    {
        int maxValue = matrix[row, 0];
        for (int column = 1; column < matrix.GetLength(1); column++)
        {
            if (matrix[row, column] > maxValue)
```

```
    {  
        maxValue = matrix[row, column];  
    }  
}  
return maxValue;  
}  
}
```

В этом примере мы создаем класс `Matrix`, который представляет матрицу попарных сравнений. Класс содержит методы для установки и получения значений в ячейках матрицы, а также для нахождения максимального значения в строке. На рис. 1 представлена форма программы для проведения расчетов по матрице попарных сравнений.

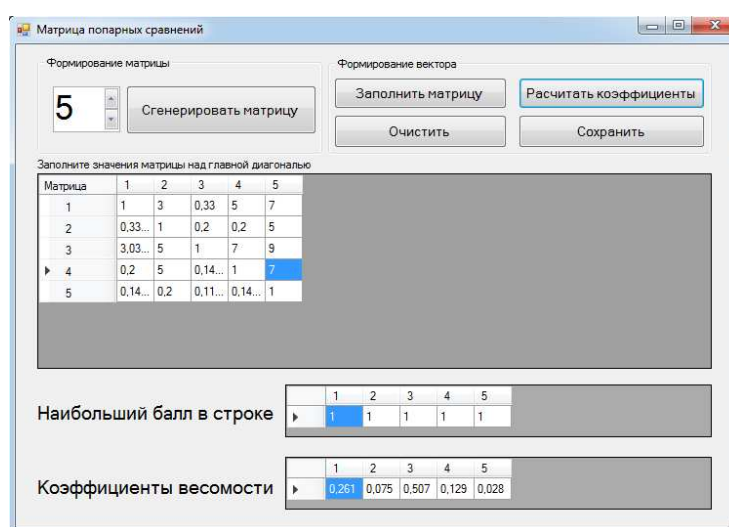


Рис. 1. Форма программы для расчетов по матрице попарных сравнений

Заключение

Программирование матрицы попарных сравнений на языке программирования C# позволяет создавать и обрабатывать матрицы попарных сравнений. Алгоритмы, которые используются для создания и обработки матрицы попарных сравнений, позволяют определить отношение предпочтения между вариантами и найти наиболее предпочтительный вариант.

Список использованных источников:

1. Разумников С.В. Алгоритм и программное обеспечение для построения агрегированных рейтингов / С.В. Разумников // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2021. – Т. 24, № 2. – С. 39–44.
2. Разумников С.В. Разработка программного обеспечения для построения агрегированных рейтингов на основе метода порогового агрегирования / С.В. Разумников // Вестник ВГУ, серия: Системный анализ и информационные технологии. – 2021. – № 2. – С. 138–152.
3. Разумников С.В. Планирование развития облачной стратегии на основе применения многокритериальной оптимизации и метода STEM / С.В. Разумников // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2020. – Т. 23, № 1. – С. 53–61.
4. Разумников С.В. Модели, алгоритмы и программное обеспечение поддержки принятия стратегических решений к переходу на облачные технологии: монография / С.В. Разумников. – Томск : Изд-во Томского политехнического университета, 2020. – 176 с.