

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им.
Р.Е.АЛЕКСЕЕВА.

Институт радиоэлектроники и информационных технологий

Отчет

по лабораторной работе № 2

по дисциплине

«Основы теории интеллектуальных вычислительных систем»

Руководитель:

_____ Мисевич П. В.

Студент:

_____ Игнаков К. М.

_____ Тихонов В. Д.

группа 19-ВМ

Работа защищена «__» _____

С оценкой _____

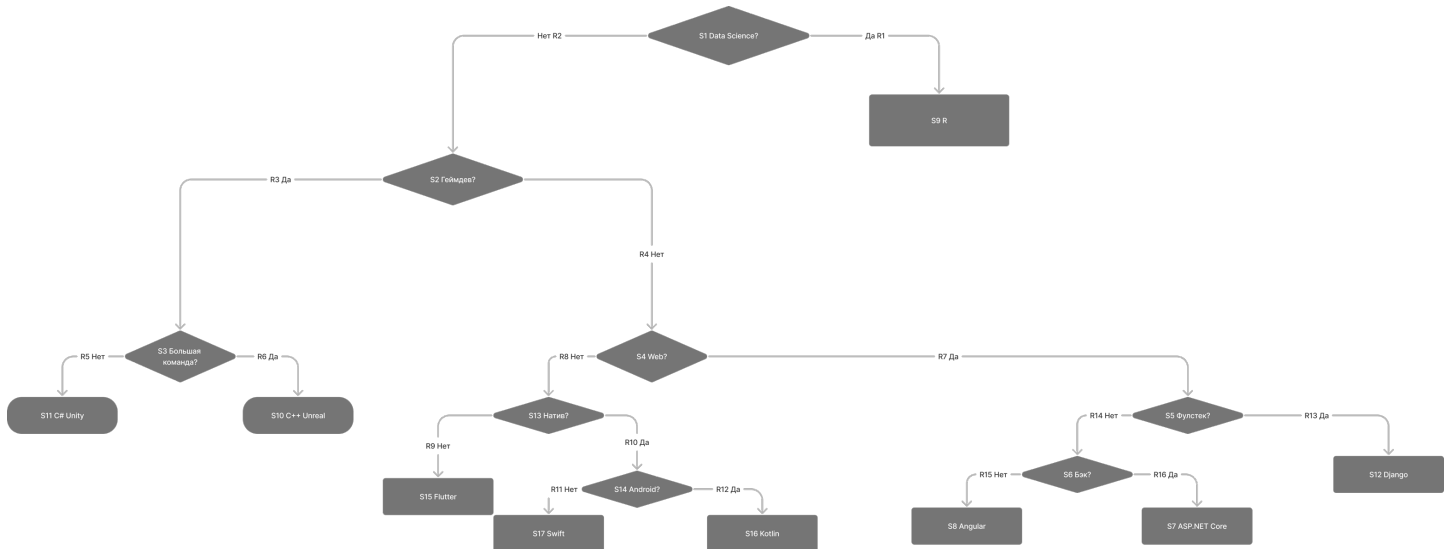
Нижний Новгород 2022

Постановка задачи

Реализация графа вопрос-ответ программой с типовыми элементами ЭС.

Ход работы

Дерево вопрос-ответ:



Код программы (формы, тк в ней располагаются вся логика для инкапсуляции):

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;
```

```
namespace LP2_Client_Server
```

```
{
```

```
public partial class Form1 : Form
```

```
{
```

```
    public Form1()
```

```
    {
```

```
        InitializeComponent();
```

```
    }
```

```
    int[] STOP_;//
```

```
    string[] S_;
```

```
    int[,] rools_;
```

```
    private void Form1_Load(object sender, EventArgs e)
```

```
    {
```

```
        // массив событий порождающего сценария множества S
```

```
        S_ = new string[100];// параметр: настроенное на 100 проодукционных правил с 3 исходами
```

```
        S_[1] = "Data Science ?";
```

```
        S_[2] = "Gamedev ?";
```

```
        S_[3] = "Big company?";
```

```
        S_[4] = "Web?";
```

```
        S_[5] = "Fullstack?";
```

```
        S_[6] = "Back-end?";
```

```
        S_[7] = "ASP.Net Core";
```

```
        S_[8] = "Angular";
```

```
        S_[9] = "R";
```

```
        S_[10] = "Unreal Engine";
```

```
        S_[11] = "C# Unity";
```

```
        S_[12] = "Python Django";
```

```
        S_[13] = "Native?";
```

```
        S_[14] = "Android?";
```

```
        S_[15] = "Flutter";
```

```

S_[16] = "Kotlin";
S_[17] = "Swift";

// закодированные логические правила: продукционная БЗ начало

//int[,] rools_ = new int[100, 2];// параметр: настроенное на 100 проодукционных правил с 3
исходами

rools_ = new int[100, 2];// параметр: настроенное на 100 проодукционных правил с 3 исходами
rools_[1, 1] = 9;
rools_[1, 0] = 2;
rools_[2, 1] = 3;
rools_[2, 0] = 4;
rools_[3, 1] = 10
rools_[3, 0] = 11
rools_[4, 1] = 5;
rools_[4, 0] = 13
rools_[5, 1] = 12
rools_[5, 0] = 6;
rools_[6, 1] = 7;
rools_[6, 0] = 8;
rools_[13, 1] = 14;
rools_[13, 0] = 15;
rools_[14, 1] = 16;
rools_[14, 0] = 17;

// закодированные логические правила: продукционная БЗ окончание

//подготавливаем хранилище индексов конечных событий начало

STOP_ = new int[100];// параметр: настроенное на 100 проодукционных правил с 3 исходами

// при отрицательном значении массива STOP_ событие конечное

// метим конечные события
for (int i = 0; i < 100; i++)
{
    STOP_[i] = 7;
}

```

```
STOP_[9] = -7;
```

```
STOP_[11] = -7;
```

```
STOP_[10] = -7;
```

```
STOP_[15] = -7;
```

```
STOP_[17] = -7;
```

```
STOP_[16] = -7;
```

```
STOP_[8] = -7;
```

```
STOP_[7] = -7;
```

```
STOP_[12] = -7;
```

```
// метим конечные события
```

```
//подготавливаем хранилище индексов конечных событий окончание
```

```
// инициализация стартового события
```

```
textBox1.Text = S_[1];
```

```
textBox3.Text = "1";
```

```
}
```

```
private void textBox2_TextChanged(object sender, EventArgs e)
```

```
{
```

```
}
```

```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
    textBox2.Text = "1";
```

```
}
```

```
private void button2_Click(object sender, EventArgs e)
```

```
{
```

```

        textBox2.Text = "0";
    }

private void button3_Click(object sender, EventArgs e)
{

    //База фактов из 2-х переменных начало

    int SDB = 0; // антецедент импликации  $S_i$  - левая часть правила
    int RDB = 0; // антецедент импликации  $R_j$  - левая часть правила
    int S_cos_ = 0; // косеквент импликации

    //наполнение БФ информацией о состоянии предметной области начало

    SDB = Int32.Parse(textBox3.Text);

    RDB = Int32.Parse(textBox2.Text);

    //наполнение БФ информацией о состоянии предметной области окончание

    //База фактов из 2-х переменных окончание


    // процедуры событий !! события отделены от правил и процедур вывода!

    // интерпретатор начало // в явном виде нет цикла

    S_cos_ = rools_[SDB, RDB];

    // textBox2.Text = S_cos_.ToString();

    textBox4.Text = Convert.ToString(S_cos_);


    // интерпретатор окончание

    // button4_Click(sender, e);

}

```

```

private void button4_Click(object sender, EventArgs e)
{
    //string[] S_ = new string[100];// параметр: настроенное на 100 проодукционных правил с 3
    исходами

    //***** КОД КНОПКИ

    int S_cos_ = 0;// косеквент импликации

    //наполнение БФ информацией о состоянии предметной области начало

    S_cos_ = Int32.Parse(textBox4.Text);
    textBox1.Text = S_[S_cos_];
    textBox3.Text = Convert.ToString(S_cos_);

    if (STOP_[S_cos_] < 0)
    {
        MessageBox.Show("концевое событие");
        Environment.Exit(0);
    }
}
}
}
}

```

Протокол решения задачи:

Data Science ?

IF S AND R THEN S

Да Нет

по правилу определить следующее событие

Выполнить событие

Gamedev ?

IF S AND R THEN S

Да Нет

по правилу определить следующее событие

Выполнить событие

Web?

IF S AND R THEN S

Да Нет

по правилу определить следующее событие

Выполнить событие

Fullstack?

IF S AND R THEN S

Да Нет

по правилу определить следующее событие

Выполнить событие

The screenshot shows a web application titled "ASP.NET Core". It features a rule configuration interface with three input fields: "IF S" containing "7", "AND R" containing "1", and "THEN S" containing "7". Below these fields are two buttons labeled "Да" (Yes) and "Нет" (No). Underneath these buttons is a button labeled "по правилу определить следующее событие" (determine the next event by rule). At the bottom of the main interface is a button labeled "Выполнить событие" (execute event). A modal dialog box is open on the right side, titled "концевое событие" (end event) with a close button (X). It contains an "OK" button.

Вывод

Данный подход в решении задачи нуждается в небольшой справке для реализации, но гораздо удобнее и проще расширяемый для небольших деревьев, по сравнению с примером, выполненным в лабораторной работе №1. Минусы данного метода в том, что для большой структуры вопрос-ответ данный метод становится сложным в поддержке.