НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №2**

з дисципліни **«**Системне програмування 2**»**

Виконала:

студентка 2 курсу гр. ІВ-71

Молчанова В.С.

Перевірив:

Павлов В.Г.

Київ 2019 р.

**Тема:** Побудова і використання об’єктів вузлів деревоподібних та ієрархічних графів.

**Мета**: вивчення методів створення та використання вузлів графів автоматів, а також дерево­по­діб­них та ієрархічних графів, організації доступу до інформації, ре­кон­струкції вхідного тексту та скорочення графів через вилу­чення повторних вузлів.

**Варіант:** 

**Граф внутрішнього подання:**

**A close up of a mans face

Description automatically generated**

**Граф автомата Мура:**

A close up of text on a white background

Description automatically generated

**Лістинг програми:**

class Machine

{

private struct SpecialAction

{

public int startState;

public string signal;

public int endState;

}

private readonly int startState;

private readonly int endState;

private int currentState;

private HashSet<string> knownSignals;

private List<SpecialAction> specialActions;

public Machine(int startState, int endState)

{

this.startState = startState;

this.endState = endState;

currentState = startState;

knownSignals = new HashSet<string>();

specialActions = new List<SpecialAction>();

}

public void StartSimulation(List<string> signals)

{

foreach (var signal in signals)

{

Console.Write($"S{currentState} ({signal})-> ");

ReactToSignal(signal);

}

Console.WriteLine($"S{currentState}");

currentState = startState;

}

public void AddSpecialAction(int startState, string signal, int endState)

{

specialActions.Add(new SpecialAction(){startState = startState, signal = signal, endState = endState});

knownSignals.Add(signal);

}

private void ReactToSignal(string signal)

{

if (ActionIsSpecial(currentState, signal))

{

currentState = specialActions.First(a => a.startState == currentState && a.signal == signal).endState;

}

else if (knownSignals.Contains(signal) && currentState < endState)

{

currentState++;

}

}

private bool ActionIsSpecial(int state, string signal)

{

return specialActions.Any(a => a.startState == state && a.signal == signal);

}

}

static class Lab2

{

public static void DoWork()

{

var machine = new Machine(1, 8);

machine.AddSpecialAction(3, "dlm", 7);

machine.AddSpecialAction(5, "cfr", 5);

machine.AddSpecialAction(4, "ltr", 8);

var sample1 = new List<string>(){ "dlm", "dlm", "dfsfsd", "dlm", "dlm", "dlm" };

var sample2 = new List<string>() { "cfr", "dlm", "ltr", "dlm", "cfr", "cfr", "ltr", "cfr", "ltr"};

var sample3 = new List<string>() { "cfr", "dlm", "ltr", "ltr", "cfr", "ltr", "dlm"};

machine.StartSimulation(sample1);

machine.StartSimulation(sample2);

machine.StartSimulation(sample3);

Console.ReadLine();

}

}

**Контрольні приклади:**

S1 (dlm)-> S2 (dlm)-> S3 (dfsfsd)-> S3 (dlm)-> S7 (dlm)-> S8 (dlm)-> S8

S1 (cfr)-> S2 (dlm)-> S3 (ltr)-> S4 (dlm)-> S5 (cfr)-> S5 (cfr)-> S5 (ltr)-> S6 (cfr)-> S7 (ltr)-> S8

S1 (cfr)-> S2 (dlm)-> S3 (ltr)-> S4 (ltr)-> S8 (cfr)-> S8 (ltr)-> S8 (dlm)-> S8