Міністерство освіти і науки України

Національний авіаційний університет

Кафедра прикладної інформатики

Лабораторна робота №8

з дисципліни: “Моделювання систем”

на тему: “Дослідження систем мережеми Петрі”

Виконав:

студент групи ТП-214

Прийнято

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2018р.

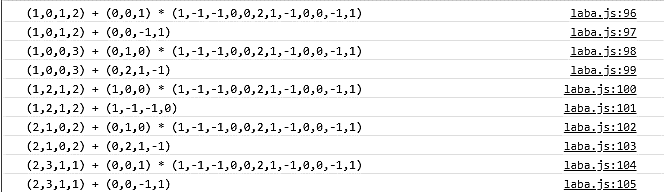
Київ 2018

**Мережі Петрі**- математичний апарат для [моделювання](https://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) динамічних дискретних систем. Вперше описані [Карлом Петрі](https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D0%BB_%D0%90%D0%B4%D0%B0%D0%BC_%D0%9F%D0%B5%D1%82%D1%80%D1%96) у [1962](https://uk.wikipedia.org/wiki/1962) році.

МП використовуються для моделювання асинхронних систем, що функціонують як сукупність паралельних взаємодіючих процесів. Аналіз МП дозволяє отримати інформацію про структуру та динамічну поведінку модельованої системи. Причинно-наслідковий зв'язок подій в асинхронних системах задається [множиною](https://uk.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%B0) відношень вигляду "умови-події". У МП умови - це позиції, а події - переходи. Відповідно до цього граф МП є [дводольним орієнтованим мультиграфом](https://uk.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D1%87%D0%B0%D1%81%D1%82%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9_%D0%B3%D1%80%D0%B0%D1%84). Орієнтовані дуги можуть сполучати лише позиції і переходи в прямому і зворотному напрямі. МП є мультиграфом, оскільки допускається кратність дуг між позиціями і переходами.

**Скріни виконання програми:**





**Код програми:**

const button = document.getElementById('btn');  
button.addEventListener('click', getInput);  
const Arr1 = [];  
function getInput() {  
  
 const Dmilus = [[0,1,1,0],[0,0,0,1],[0,0,1,0]];  
 const Dplus = [[1,0,0,0],[0,2,1,0],[0,0,0,1]];  
 const Mi = [1,0,1,2];  
  
 let D = [];  
  
 for (let i = 0; i < Dmilus.length; i++) {  
 D[i] = [];  
 for (let j = 0; j < Dmilus[0].length; j++) {  
 D[i][j] = Dplus[i][j] - Dmilus[i][j];  
 }  
 }  
   
 let Estart = [0,0,1];  
  
 function MGenerate(Estart) {  
 let M = [0,0,0,0];  
 for (let i = 0; i < D[0].length; i++) {  
 for (let j = 0; j < D.length; j++) {  
 M[i] += D[j][i] \* Estart[j];  
 }  
 }  
 return(M)  
 }  
  
 let M = MGenerate(Estart);  
  
  
  
  
 function MAddition(Mi, M){  
 let MiPlus = [];  
 for (let i = 0; i <Mi.length; i++) {  
 MiPlus[i] = Mi[i] + M[i];  
 }  
 return(MiPlus);  
 }  
 let M1Plus = MAddition(Mi,M);  
 // console.log(M1Plus);  
 ///////////////////////////////////////////////  
 let E2start = [0,1,0];  
 let M1 = MGenerate(E2start);  
 let M2Plus = MAddition(M1Plus, M1);  
 // console.log(M2Plus);  
 ////////////////////////////////////////////  
 let E3start = [1,0,0];  
 let M2 = MGenerate(E3start);  
 let M3Plus = MAddition(M2Plus, M2);  
 // console.log(M3Plus);  
 ///////////////////////////////////////////  
 let E4start = [0,1,0];  
 let M3 = MGenerate(E4start);  
 let M4Plus = MAddition(M3Plus, M3);  
 // console.log(M4Plus);  
 ////////////////////////////////////////////  
 let E5start = [0,0,1];  
 let M4 = MGenerate(E5start);  
 let M5Plus = MAddition(M4Plus, M4);  
 // console.log(M5Plus);  
  
  
 let Rezult1 = document.getElementById('1it');  
 let Rezult2 = document.getElementById('2it');  
 let Rezult3 = document.getElementById('3it');  
 let Rezult4 = document.getElementById('4it');  
 let Rezult5 = document.getElementById('5it');  
 let Rezult6 = document.getElementById('6it');  
  
 Rezult1.innerText = `{${Mi}}`;  
 Rezult2.innerText = `{${M1Plus}}`;  
 Rezult3.innerText = `{${M2Plus}}`;  
 Rezult4.innerText = `{${M3Plus}}`;  
 Rezult5.innerText = `{${M4Plus}}`;  
 Rezult6.innerText = `{${M5Plus}}`;  
  
 let start1 = document.getElementById('t2');  
 let start2 = document.getElementById('t3');  
 let start3 = document.getElementById('t4');  
 let start4 = document.getElementById('t5');  
 let start5 = document.getElementById('t6');  
  
  
 start1.innerText = `e(k) = {${Estart}}`;  
 start2.innerText = `e(k) = {${E2start}}`;  
 start3.innerText = `e(k) = {${E3start}}`;  
 start4.innerText = `e(k) = {${E4start}}`;  
 start5.innerText = `e(k)= {${E5start}}`;  
 console.log(`(${Mi}) + (${Estart}) \* (${D})`);  
 console.log(`(${Mi}) + (${M})`);  
 console.log(`(${M1Plus}) + (${E2start}) \* (${D})`);  
 console.log(`(${M1Plus}) + (${M1})`);  
 console.log(`(${M2Plus}) + (${E3start}) \* (${D})`);  
 console.log(`(${M2Plus}) + (${M2})`);  
 console.log(`(${M3Plus}) + (${E4start}) \* (${D})`);  
 console.log(`(${M3Plus}) + (${M3})`);  
 console.log(`(${M4Plus}) + (${E5start}) \* (${D})`);  
 console.log(`(${M4Plus}) + (${M4})`);