МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(«ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №2

по дисциплине «Теория автоматов»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щесняк Д. С./

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Мельцов В.Ю./

Киров 2015

**1 Задание на лабораторную работу**

Реализовать в рамках битвы логических стратегий алгоритмы, позволяющие играть в игры «Камень, ножницы, бумага» и «Морской бой».

**2 КНБ. Блок – схема алгоритма**





3 Экранные формы

Экранные формы представлены на рисунке 1.

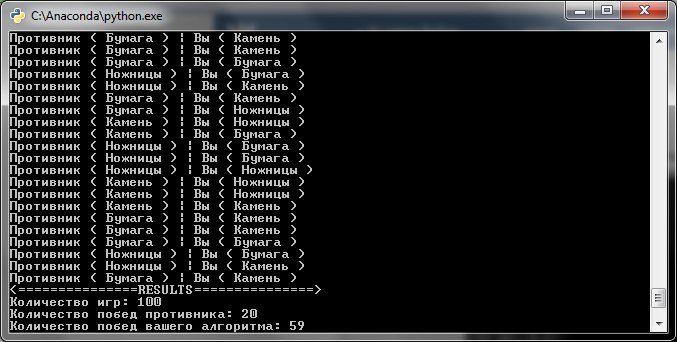


Рисунок 1 – результат работы алгоритма

4 Листинг программы КНБ.

|  |
| --- |
| unit RPSImpl;  interface  uses sysUtils, crt;  type  TBin = array [0..1] of integer;  TByteArray = array of integer;  TNeuron = Object  info : integer;  out : real;  w : array [0..3] of real;  constructor Create(n : integer);  function dot(input : array of integer) : real;  procedure learn(input : array of integer);  end;  TNeuralNet = Object  neurons : array[0..2] of TNeuron;  lastInput : array[0..3] of integer;  lastNum : integer;  constructor Create();  function run(input : array of integer) : integer;  procedure learn();  procedure show();  end;  procedure setParams(games, winsPerRound: integer);  function choice(prevRival: integer): integer;  procedure onCurrentGameSetEnd;  var NN : TNeuralNet;  loses : integer = 0;  gameCount, gameNum : integer;  hist : array of integer;  implementation  function bin(num : integer) : string;  begin  bin := IntToStr(num mod 2);  if num >= 2 then  bin := bin(num div 2) + bin;  end;  function intToBin(num: integer) : TBin;  var a : string;  i: integer;  begin  a := bin(num);  if length(a) = 1 then  a := '0' + a;  for i := 0 to length(a) - 1 do  intToBin[i] := StrToInt(a[i + 1]);  end;  function join(a, b: Tbin) : TByteArray;  var i : byte;  begin  setLength(join, length(a) + length(b));  for i := 0 to length(a) - 1 do  join[i] := a[i];  for i := 0 to length(b) - 1 do  join[i + length(a)] := b[i];  end;  constructor TNeuron.Create(n : integer);  var i : integer;  begin  inherited;  info := n;  out := 0;  for i := 0 to 3 do  w[i] := random(2000) / 1000 - 1;  end;  function TNeuron.dot(input : array of integer) : real;  var i: integer;  net : real;  begin  net := 0;  for i := 0 to 3 do  net := net + input[i] \* w[i];  out := 1 / (1 + exp(-net));  dot := out;  end;  procedure TNeuron.learn(input : array of integer);  var i : integer;  begin  for i := 0 to 3 do begin  w[i] := w[i] - out \* input[i];  end;  end;  constructor TNeuralNet.create();  var i : integer;  begin  randomize;  for i := 0 to 2 do  neurons[i].create(i + 1);  end;  function TNeuralNet.run(input : array of integer) : integer;  var k, i : integer;  begin  lastInput := input;  k := 0;  neurons[0].dot(input);  for i := 1 to 2 do begin  neurons[i].dot(input);  if neurons[i].out > neurons[k].out then  k := i;  end;  lastNum := neurons[k].info;  run := neurons[k].info;  end;  procedure TNeuralNet.learn();  begin  neurons[lastNum - 1].learn(lastInput);  end;  procedure TNeuralNet.show();  var i, j: integer;  begin  for i := 0 to 2 do begin  writeln(neurons[i].info);  for j := 0 to 1 do  writeln(neurons[i].w[j]);  writeln;  end;  end;  function cmp(a, b : integer) : integer;  begin  if a = b then cmp := 0  else if ( (a = 1) and (b = 2) ) or ( (a = 2) and (b = 3) ) or ( (a = 3) and (b = 1) ) then  cmp := 1  else cmp := -1;  end;  procedure setParams(games, winsPerRound: integer);  begin  gameCount := games;  gameNum := 0;  setLength(hist, games);  NN.create();  end;  function choice(prevRival: integer): integer;  var i : integer;  begin  if prevRival = 0 then  choice := NN.run(join(IntToBin(1), intToBin(1)))  else begin  if cmp(NN.lastNum, prevRival) <> 1 then begin  NN.learn();  end;  choice := NN.run(join(IntToBin(NN.lastNum), IntToBin(prevRival)));  end;  inc(gameNum);  end;  procedure onCurrentGameSetEnd;  begin  gameNum := 0;  NN.create();  end;  end. |

5 Блок – схема алгоритма игры в морской бой







6 Экранные формы

Экранные формы работы алгоритма представлены на рисунке 2

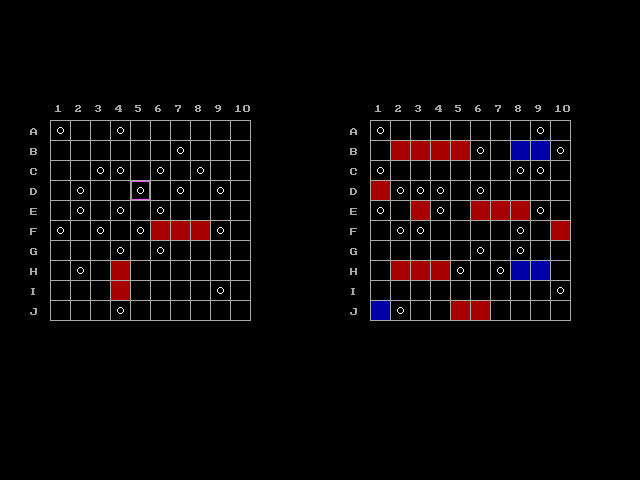


Рисунок 2 – демонстрация работы алгоритма