МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Разработка бота для игры в «Камень-ножницы-бумага»

Отчет

Лабораторная работа №2 по дисциплине

«Теория автоматов»

Выполнил студент группы ИВТ-23 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Мельцов В.Ю./

Киров 2020

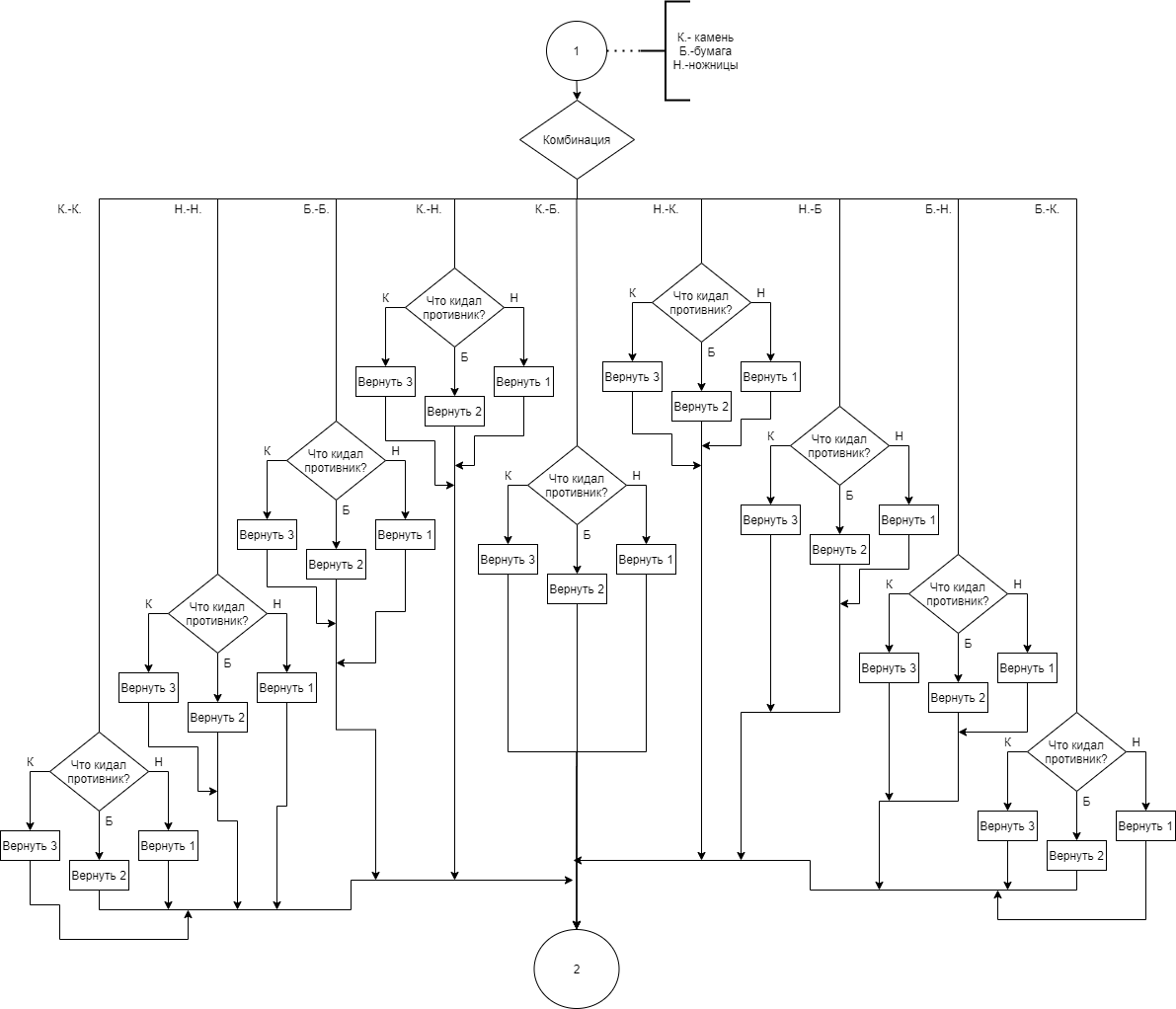
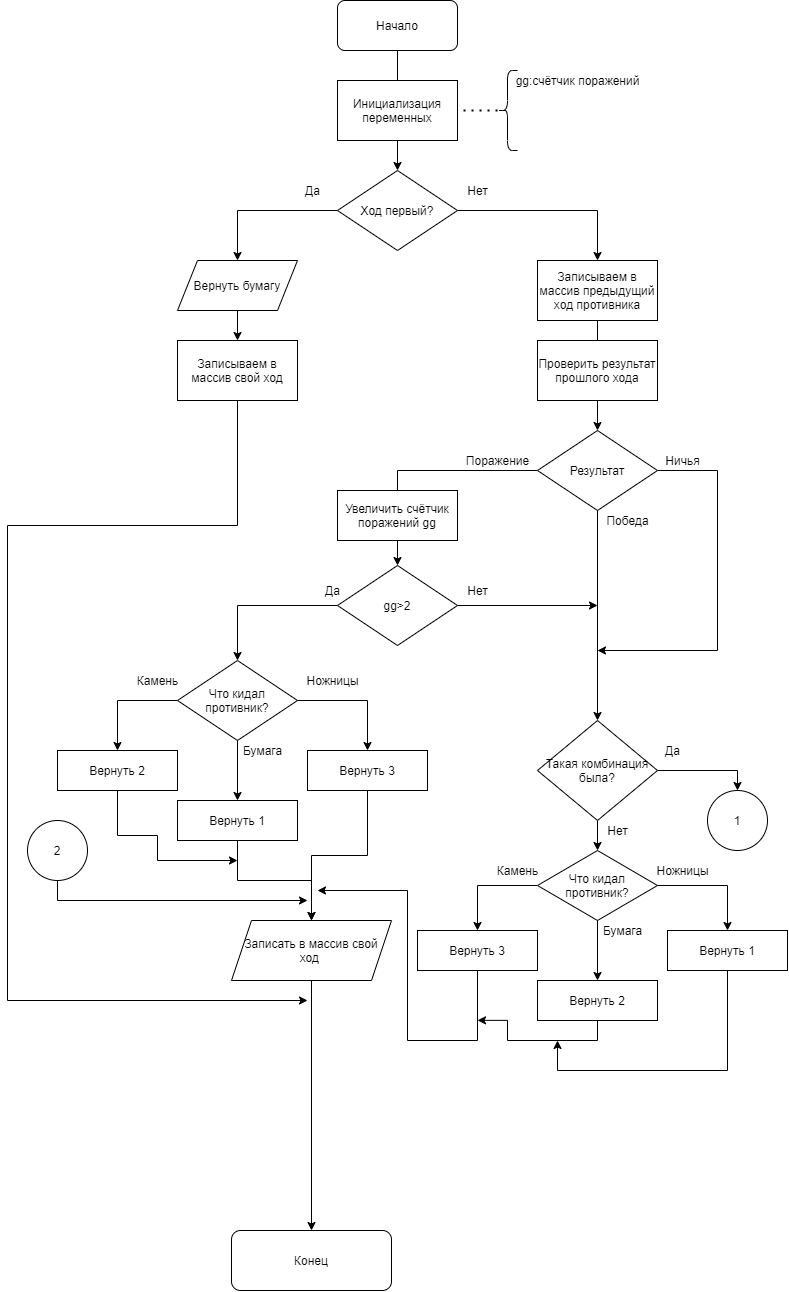
**Цель работы**

Написание алгоритма и разработка бота для игры в «Камень-ножницы-бумага».

**Словесное описание алгоритма**

Данный алгоритм содержит 2 тактики боя против оппонента: на случай если он решит подстраховаться и нет. Суть обоих алгоритмов целиком зависит от двух динамических массивов, в которые записываются ходы противника и ходы моего бота. Первым ходом в первой игре с каждым соперником бот кидает бумагу, т.к. ещё нечего анализировать. Затем в массивы записывается каждая комбинация ходов, сделанная моим ботом и противником. Затем вся эта информация обрабатывается и в конечном счёте в массивах скапливается информация о всех возможных комбинациях, их всего 9. Главное, что потребуется для нормальной работы алгоритма – это следующее, что выкинул противник при данной комбинации. Примерно после 10 сетов уже имеется достаточно информации о боте противника. Теперь, когда в следующий раз, когда возникнет подобная комбинация, к примеру: камень-камень – мой бот уже знает, что следующим выкидывал соперник при такой же комбинации. Если в предыдущий раз при комбинации камень-камень соперник выкинул ножницы, то мой бот выкинет камень. Каждый раз, когда какая-либо комбинация повторяется, бот проверяет следующий ход именно по новой комбинации, а не по предыдущей такой же, чтобы избежать каких-нибудь «перестроек» в алгоритме противника. Соответственно, на этом и строится главная часть бота, а вот уже второй его алгоритм, можно назвать контралгоритмом. Он применяется в том случае, если у противника такой же бот или его бот «раскусил» алгоритм моего бота. Его суть состоит в том, что если мой бот проиграл три раза подряд по предыдущей тактике, то он просто меняет стратегию своей игры и теперь в ответ на всё выкидывает то, что должно было проиграть при предыдущей тактике. На этом весь алгоритм кончается.

**Схема алгоритма**



**Листинг бота**

Unit RPSBotImpl;

{$mode objfpc}{$H+}

interface

const

UNKNOWN = 0;

ROCK = 1;

SCISSORS = 2;

PAPER = 3;

procedure setParams(setCount, winsPerSet: integer);

function choice(previousRivalChoice: integer): integer;

procedure onCurrentGameEnd;

implementation

var m:array of integer;

mm: array of integer;

i,iii:integer;

procedure setParams(setCount, winsPerSet: integer);

begin

setlength(m,4); // массивы для записи своих ходов и ходов противника

setlength(mm,4);

i:=3;

iii:=4;

end;

function choice(previousRivalChoice: integer): integer;

var j,jj:integer;

a,b,c:integer;

aa,bb,cc,dd,ee,ff,gg:integer;

begin

a:=0;

b:=0;

c:=0;

gg:=0;

aa:=0;

bb:=0;

cc:=0;

dd:=0;

ee:=0;

ff:=0;

m[i]:=previousRivalChoice;

choice:=PAPER; //выкидываем бумагу в первый раз

if (m[i]=ROCK) and (mm[i-1]=ROCK) then

begin

for jj:=i-1 downto 4 do //проверяем была ли данная комбинация ранее. И так для всех комбинаций

if (m[jj]=ROCK) and (mm[jj-1]=ROCK) then

begin

gg:=gg+1;

if gg=1 then

begin

aa:=m[jj+1];

bb:=mm[jj];

end;

if gg=2 then

begin

cc:=m[jj+1];

dd:=mm[jj];

end;

if gg=3 then

begin

ee:=m[jj+1];

ff:=mm[jj];

break;

end;

end;

if gg=3 then //Контр тактика. Применяется, если мы три раза проиграли по прошлой тактике

if ((ff=ROCK) and (cc=PAPER) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=ROCK) and (aa=PAPER)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=PAPER) and (aa=SCISSORS)) then

if bb=ROCK then choice:=SCISSORS else //Выбираем значения, противоположные основной тактике

if bb=PAPER then choice:=ROCK else

if bb=SCISSORS then choice:=PAPER;

if (choice<>PAPER) or (choice<>SCISSORS) or (choice<>ROCK) then

begin

for j:=i-1 downto 4 do //Основной алгоритм. Выбрасываем то, что побеждает противника.

if (m[j]=ROCK) and (mm[j-1]=ROCK) then

begin

if m[j+1]=ROCK then choice:=PAPER;

if m[j+1]=SCISSORS then choice:=ROCK;

if m[j+1]=PAPER then choice:=SCISSORS;

break;

end else

choice:=ROCK;

end;

end else

if (m[i]=SCISSORS) and (mm[i-1]=SCISSORS) then

begin

for jj:=i-1 downto 4 do

if (m[jj]=SCISSORS) and (mm[jj-1]=SCISSORS) then

begin

gg:=gg+1;

if gg=1 then

begin

aa:=m[jj+1];

bb:=mm[jj];

end;

if gg=2 then

begin

cc:=m[jj+1];

dd:=mm[jj];

end;

if gg=3 then

begin

ee:=m[jj+1];

ff:=mm[jj];

break;

end;

end;

if gg=3 then

if ((ff=ROCK) and (cc=PAPER) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=ROCK) and (aa=PAPER)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=PAPER) and (aa=SCISSORS)) then

if bb=ROCK then choice:=SCISSORS else

if bb=PAPER then choice:=ROCK else

if bb=SCISSORS then choice:=PAPER;

if (choice<>PAPER) or (choice<>SCISSORS) or (choice<>ROCK) then

begin

for j:=i-1 downto 4 do

if (m[j]=SCISSORS) and (mm[j-1]=SCISSORS) then

begin

if m[j+1]=ROCK then choice:=PAPER;

if m[j+1]=SCISSORS then choice:=ROCK;

if m[j+1]=PAPER then choice:=SCISSORS;

break;

end else

choice:=SCISSORS;

end;

end else

if (m[i]=PAPER) and (mm[i-1]=PAPER) then

begin

for jj:=i-1 downto 4 do

if (m[jj]=PAPER) and (mm[jj-1]=PAPER) then

begin

gg:=gg+1;

if gg=1 then

begin

aa:=m[jj+1];

bb:=mm[jj];

end;

if gg=2 then

begin

cc:=m[jj+1];

dd:=mm[jj];

end;

if gg=3 then

begin

ee:=m[jj+1];

ff:=mm[jj];

break;

end;

end;

if gg=3 then

if ((ff=ROCK) and (cc=PAPER) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=ROCK) and (aa=PAPER)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=PAPER) and (aa=SCISSORS)) then

if bb=ROCK then choice:=SCISSORS else

if bb=PAPER then choice:=ROCK else

if bb=SCISSORS then choice:=PAPER;

if (choice<>PAPER) or (choice<>SCISSORS) or (choice<>ROCK) then

begin

for j:=i-1 downto 4 do

if (m[j]=PAPER) and (mm[j-1]=PAPER) then

begin

if m[j+1]=ROCK then choice:=PAPER;

if m[j+1]=SCISSORS then choice:=ROCK;

if m[j+1]=PAPER then choice:=SCISSORS;

break;

end else

choice:=PAPER;

end;

end else

if (m[i]=ROCK) and (mm[i-1]=PAPER) then

begin

for jj:=i-1 downto 4 do

if (m[jj]=ROCK) and (mm[jj-1]=PAPER) then

begin

gg:=gg+1;

if gg=1 then

begin

aa:=m[jj+1];

bb:=mm[jj];

end;

if gg=2 then

begin

cc:=m[jj+1];

dd:=mm[jj];

end;

if gg=3 then

begin

ee:=m[jj+1];

ff:=mm[jj];

break;

end;

end;

if gg=3 then

if ((ff=ROCK) and (cc=PAPER) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=ROCK) and (aa=PAPER)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=PAPER) and (aa=SCISSORS)) then

if bb=ROCK then choice:=SCISSORS else

if bb=PAPER then choice:=ROCK else

if bb=SCISSORS then choice:=PAPER;

if (choice<>PAPER) or (choice<>SCISSORS) or (choice<>ROCK) then

begin

for j:=i-1 downto 4 do

if (m[j]=ROCK) and (mm[j-1]=PAPER) then

begin

if m[j+1]=ROCK then choice:=PAPER;

if m[j+1]=SCISSORS then choice:=ROCK;

if m[j+1]=PAPER then choice:=SCISSORS;

break;

end else

choice:=ROCK;

end;

end else

if (m[i]=SCISSORS) and (mm[i-1]=ROCK) then

begin

for jj:=i-1 downto 4 do

if (m[jj]=SCISSORS) and (mm[jj-1]=ROCK) then

begin

gg:=gg+1;

if gg=1 then

begin

aa:=m[jj+1];

bb:=mm[jj];

end;

if gg=2 then

begin

cc:=m[jj+1];

dd:=mm[jj];

end;

if gg=3 then

begin

ee:=m[jj+1];

ff:=mm[jj];

break;

end;

end;

if gg=3 then

if ((ff=ROCK) and (cc=PAPER) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=ROCK) and (aa=PAPER)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=PAPER) and (aa=SCISSORS)) then

if bb=ROCK then choice:=SCISSORS else

if bb=PAPER then choice:=ROCK else

if bb=SCISSORS then choice:=PAPER;

if (choice<>PAPER) or (choice<>SCISSORS) or (choice<>ROCK) then

begin

for j:=i-1 downto 4 do

if (m[j]=SCISSORS) and (mm[j-1]=ROCK) then

begin

if m[j+1]=ROCK then choice:=PAPER;

if m[j+1]=SCISSORS then choice:=ROCK;

if m[j+1]=PAPER then choice:=SCISSORS;

break;

end else

choice:=SCISSORS;

end;

end else

if (m[i]=PAPER) and (mm[i-1]=SCISSORS) then

begin

for jj:=i-1 downto 4 do

if (m[jj]=PAPER) and (mm[jj-1]=SCISSORS) then

begin

gg:=gg+1;

if gg=1 then

begin

aa:=m[jj+1];

bb:=mm[jj];

end;

if gg=2 then

begin

cc:=m[jj+1];

dd:=mm[jj];

end;

if gg=3 then

begin

ee:=m[jj+1];

ff:=mm[jj];

break;

end;

end;

if gg=3 then

if ((ff=ROCK) and (cc=PAPER) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=ROCK) and (aa=PAPER)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=PAPER) and (aa=SCISSORS)) then

if bb=ROCK then choice:=SCISSORS else

if bb=PAPER then choice:=ROCK else

if bb=SCISSORS then choice:=PAPER;

if (choice<>PAPER) or (choice<>SCISSORS) or (choice<>ROCK) then

begin

for j:=i-1 downto 4 do

if (m[j]=PAPER) and (mm[j-1]=SCISSORS) then

begin

if m[j+1]=ROCK then choice:=PAPER;

if m[j+1]=SCISSORS then choice:=ROCK;

if m[j+1]=PAPER then choice:=SCISSORS;

break;

end else

choice:=PAPER;

end;

end else

if (m[i]=ROCK) and (mm[i-1]=SCISSORS) then

begin

for jj:=i-1 downto 4 do

if (m[jj]=ROCK) and (mm[jj-1]=SCISSORS) then

begin

gg:=gg+1;

if gg=1 then

begin

aa:=m[jj+1];

bb:=mm[jj];

end;

if gg=2 then

begin

cc:=m[jj+1];

dd:=mm[jj];

end;

if gg=3 then

begin

ee:=m[jj+1];

ff:=mm[jj];

break;

end;

end;

if gg=3 then

if ((ff=ROCK) and (cc=PAPER) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=ROCK) and (aa=PAPER)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=PAPER) and (aa=SCISSORS)) then

if bb=ROCK then choice:=SCISSORS else

if bb=PAPER then choice:=ROCK else

if bb=SCISSORS then choice:=PAPER;

if (choice<>PAPER) or (choice<>SCISSORS) or (choice<>ROCK) then

begin

for j:=i-1 downto 4 do

if (m[j]=ROCK) and (mm[j-1]=SCISSORS) then

begin

if m[j+1]=ROCK then choice:=PAPER;

if m[j+1]=SCISSORS then choice:=ROCK;

if m[j+1]=PAPER then choice:=SCISSORS;

break;

end else

choice:=SCISSORS;

end;

end else

if (m[i]=SCISSORS) and (mm[i-1]=PAPER) then

begin

for jj:=i-1 downto 4 do

if (m[jj]=SCISSORS) and (mm[jj-1]=PAPER) then

begin

gg:=gg+1;

if gg=1 then

begin

aa:=m[jj+1];

bb:=mm[jj];

end;

if gg=2 then

begin

cc:=m[jj+1];

dd:=mm[jj];

end;

if gg=3 then

begin

ee:=m[jj+1];

ff:=mm[jj];

break;

end;

end;

if gg=3 then

if ((ff=ROCK) and (cc=PAPER) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=ROCK) and (aa=PAPER)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=PAPER) and (aa=SCISSORS)) then

if bb=ROCK then choice:=SCISSORS else

if bb=PAPER then choice:=ROCK else

if bb=SCISSORS then choice:=PAPER;

if (choice<>PAPER) or (choice<>SCISSORS) or (choice<>ROCK) then

begin

for j:=i-1 downto 4 do

if (m[j]=SCISSORS) and (mm[j-1]=PAPER) then

begin

if m[j+1]=ROCK then choice:=PAPER;

if m[j+1]=SCISSORS then choice:=ROCK;

if m[j+1]=PAPER then choice:=SCISSORS;

break;

end else

choice:=PAPER;

end;

end else

if (m[i]=PAPER) and (mm[i-1]=ROCK) then

begin

for jj:=i-1 downto 4 do

if (m[jj]=PAPER) and (mm[jj-1]=ROCK) then

begin

gg:=gg+1;

if gg=1 then

begin

aa:=m[jj+1];

bb:=mm[jj];

end;

if gg=2 then

begin

cc:=m[jj+1];

dd:=mm[jj];

end;

if gg=3 then

begin

ee:=m[jj+1];

ff:=mm[jj];

break;

end;

end;

if gg=3 then

if ((ff=ROCK) and (cc=PAPER) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=ROCK) and (cc=PAPER) and (dd=PAPER) and (aa=SCISSORS)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=ROCK) and (aa=PAPER)) or

((ff=PAPER) and (cc=SCISSORS) and (dd=SCISSORS) and (aa=ROCK)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=ROCK) and (aa=PAPER)) or

((ff=SCISSORS) and (cc=ROCK) and (dd=PAPER) and (aa=SCISSORS)) then

if bb=ROCK then choice:=SCISSORS else

if bb=PAPER then choice:=ROCK else

if bb=SCISSORS then choice:=PAPER;

if (choice<>PAPER) or (choice<>SCISSORS) or (choice<>ROCK) then

begin

for j:=i-1 downto 4 do

if (m[j]=PAPER) and (mm[j-1]=ROCK) then

begin

if m[j+1]=ROCK then choice:=PAPER;

if m[j+1]=SCISSORS then choice:=ROCK;

if m[j+1]=PAPER then choice:=SCISSORS;

break;

end else

choice:=ROCK;

end;

end;

mm[i]:=choice;

i:=i+1;

iii:=iii+1;

setlength(m,iii);

setlength(mm,iii);

end;

procedure onCurrentGameEnd;

begin

setlength(m,4); //Обнуляем массивы

setlength(mm,4);

i:=3;

iii:=4;

end;

end.

**Вывод**

В ходе выполнения второй лабораторной работы по теории автоматов была проделана работа по созданию бота, предназначенного для участия в турнире «Камень-ножницы-бумага». Главной особенностью данной лабораторной работой стало то, что появилась возможность проверить свои знания в действии и посоревноваться против других участников турнира со своим алгоритмом.