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 «Морской бой»

Отчет

Лабораторная работа №3 по дисциплине

«Теория автоматов»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

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

Киров 2020

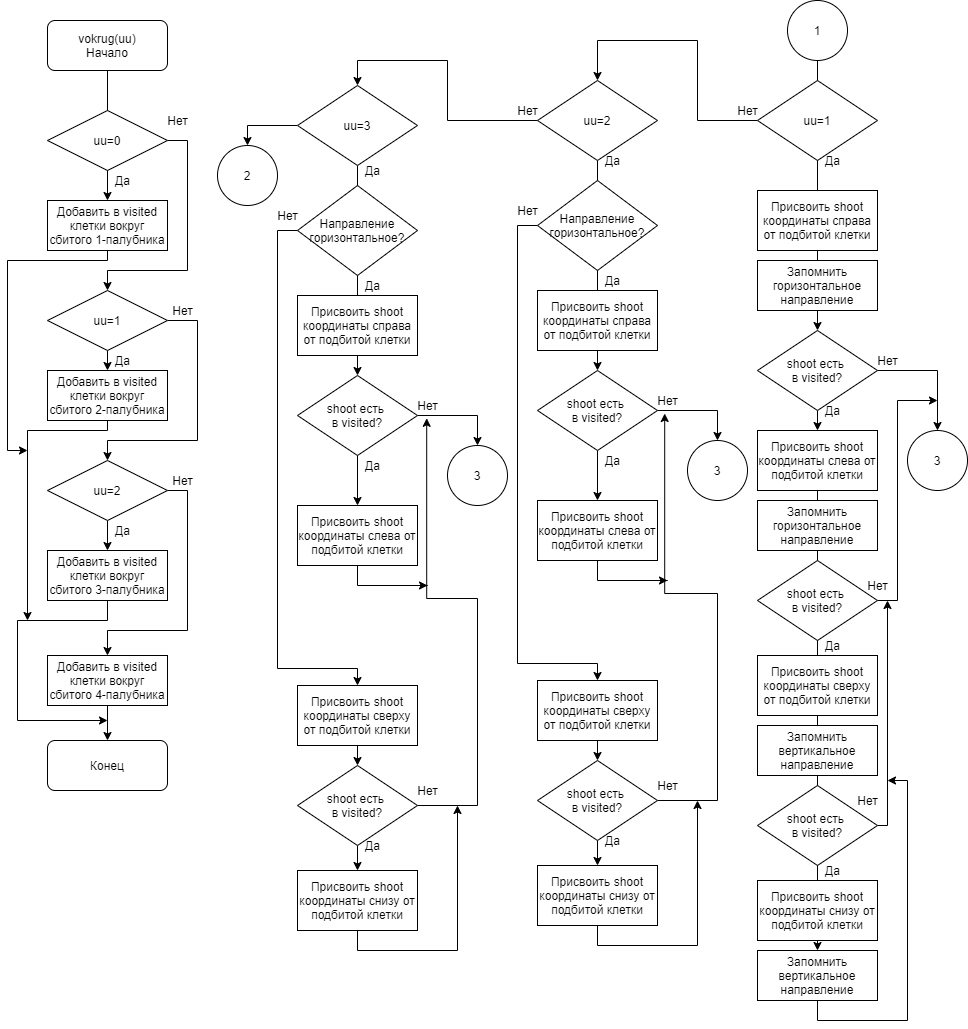
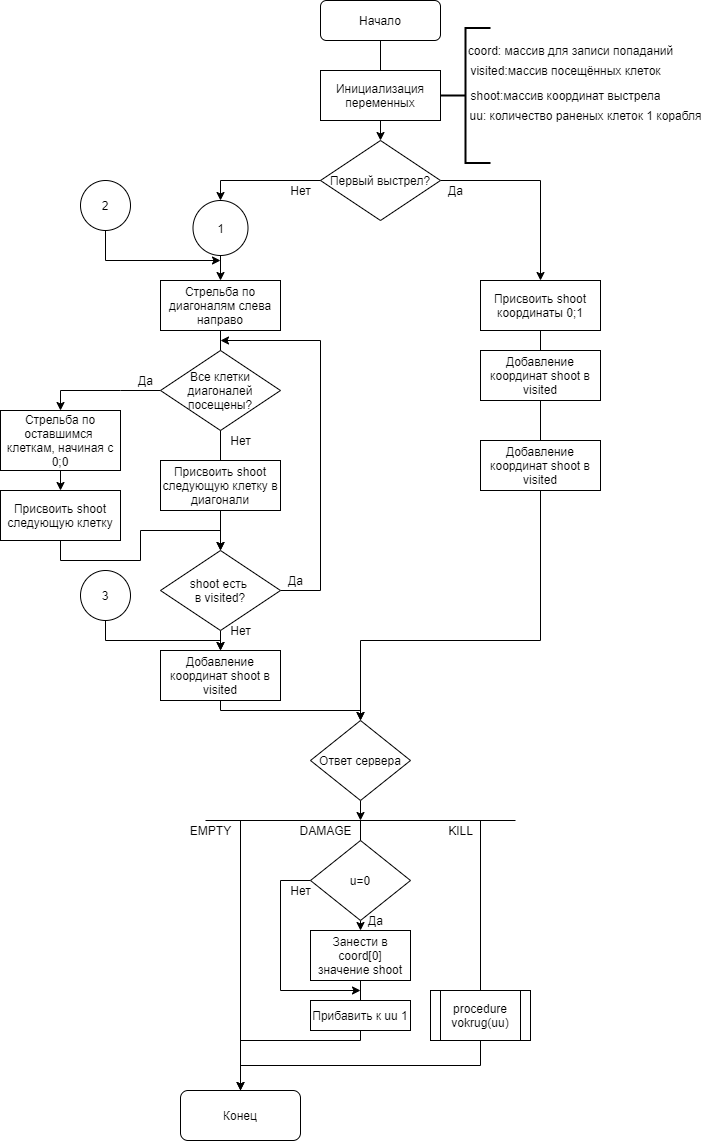
**Цель работы**

Написание алгоритма и разработка бота для игры в «Морской бой».

**Словесное описание алгоритма**

Данный бот имеет два алгоритма: алгоритм стрельбы и расстановки кораблей. Сначала происходит расстановка кораблей. В боте она реализована данным образом: имеется 3 заранее заготовленных расстановки, которые в ходе сетов постоянно меняются по заранее заготовленному алгоритму. Суть алгоритма в том, что 1-палубники, 1-клеточные корабли, с каждым сетов смещаются на 1 по x и на 1 по y, при этом исключены все случаи неправильной расстановки. Вместе со сменой местоположения палубников также меняется расположение и остальных кораблей. 4-палубник, 3-палубники и 2-палубники в каждой из 3-х расстановок находятся с краю карты. После каждого сета боковые стороны меняются кораблями. Расстановка сбоку является наиболее выгодной, т.к. после уничтожения бокового корабля, противник закрывает меньшее количество клеток поля, за счёт клеток с одной из сторон, которых нет. Таким образом, в общей сложности получается около 30 различных расстановок. Вторым, не менее важным, алгоритмом является алгоритм обстрела карты противника. Данный бот исходит из предположения, что противник решит применить схожую тактику расстановки кораблей, боковую. Обстрел начинается с боковых точек, через 1 клетку. После обстрел превращается в диагональный. Как только бот попадёт по кораблю противника, его координаты записываются в отдельный массив и дальше продолжается добивание данного корабля. Сначала проверяются верхние, потом нижние, а затем правая и левая клетки. После повторного попадания в корабль запоминается ось, в которой расположен корабль, горизонтальная или диагональная. После этого добивание корабля происходит уже только по горизонтали или вертикали. Если корабль противника был убит, все клетки вокруг него, включая диагональные, закрываются для дальнейшего обстрела. При обстреле какой-либо клетки, она также закрывается. После диагонального обстрела, если у противника ещё остались корабли, идёт проверка всех оставшихся клеток.

**Схема алгоритма**



**Листинг бота**

unit SeaFightBotImpl;

{$mode objfpc}{$H+}

interface

type

TByteArray = array of Byte;

TByteArray2D = array of TByteArray;

const

EMPTY = 0;

BLOCK = 1;

DAMAGE = 2;

KILL = 3;

function getMap: TByteArray2D;

function shoot: TByteArray;

procedure shootResult(resultCode: integer);

procedure rivalShoot(point: TByteArray);

procedure setParams(setsPerGame: integer);

procedure onCurrentSetEnd;

procedure onCurrentGameEnd;

implementation

var x,y,iiii:integer;

i,j,mas:integer;

m:array [0..9,0..9] of byte; //Массив для изменения расстановки 1 карты

dop: array [0..9,0..9] of byte;

enemy:array of integer;

c,d,e,f,g,h,proverka:integer;

iii:integer;

vopros,gg,tt:boolean;

destroy:integer;

realx,realy:integer;

visited:array [0..99] of boolean; //Массив координат всего поля, чтобы определить посещённые

a,b:integer;

kek:integer;

verh:boolean;

coord:array [0..3] of integer; //массив для запоминания координат подбитого корабля

uu:integer;

aa:boolean;

povorot:integer; //Запоминание стороны, в которую направлен подбитый корабль

bb:integer;

function getMap: TByteArray2D; //Заполнение карты происходит по 3-м расстановкам, которые в ходе каждого сета меняются и превращаются в 30

begin

aa:=false;

iiii:=1;

uu:=0;

for i:=0 to 3 do

coord[i]:=0;

verh:=true;

povorot:=0;

proverka:=0;

for i:=0 to 99 do

visited[i]:=false;

gg:=true;

if bb=29 then

bb:=0;

if bb=0 then

begin

for i:=0 to 9 do

for j:=0 to 9 do

begin

m[i,j]:=0;

dop[i,j]:=0;

end;

getMap := TByteArray2D.Create(

TByteArray.Create(0, 0, 1, 1, 1, 1, 0, 0, 0, 0),

TByteArray.Create(0, 0, 0, 0, 0, 0, 0, 0, 0, 0),

TByteArray.Create(0, 0, 0, 0, 0, 0, 0, 0, 0, 1),

TByteArray.Create(0, 0, 0, 0, 0, 0, 0, 0, 0, 1),

TByteArray.Create(0, 0, 0, 0, 1, 0, 0, 0, 0, 1),

TByteArray.Create(1, 0, 1, 0, 0, 0, 0, 0, 0, 0),

TByteArray.Create(1, 0, 0, 0, 0, 0, 0, 0, 0, 1),

TByteArray.Create(1, 0, 0, 0, 0, 1, 0, 1, 0, 1),

TByteArray.Create(0, 0, 0, 0, 0, 0, 0, 0, 0, 0),

TByteArray.Create(0, 0, 1, 1, 0, 1, 1, 0, 0, 0)

);

m[9,5]:=1;

m[9,6]:=1;

m[2,9]:=1;

m[3,9]:=1;

m[4,9]:=1;

m[4,4]:=1;

m[5,0]:=1;

m[6,0]:=1;

m[7,0]:=1;

m[5,2]:=1;

m[7,5]:=1;

m[7,7]:=1;

m[6,9]:=1;

m[7,9]:=1;

m[9,2]:=1;

m[9,3]:=1;

m[0,2]:=1;

m[0,3]:=1;

m[0,4]:=1;

m[0,5]:=1;

end else

if bb=9 then

begin

for i:=0 to 9 do

for j:=0 to 9 do

begin

m[i,j]:=0;

dop[i,j]:=0;

end;

getMap := TByteArray2D.Create(

TByteArray.Create(0, 0, 0, 0, 0, 1, 1, 1, 0, 0),

TByteArray.Create(0, 0, 0, 0, 0, 0, 0, 0, 0, 0),

TByteArray.Create(1, 0, 1, 0, 0, 0, 1, 0, 0, 1),

TByteArray.Create(1, 0, 0, 0, 0, 0, 0, 0, 0, 1),

TByteArray.Create(1, 0, 0, 0, 0, 0, 0, 0, 0, 1),

TByteArray.Create(0, 0, 1, 0, 0, 1, 0, 0, 0, 1),

TByteArray.Create(1, 0, 0, 0, 0, 0, 0, 0, 0, 0),

TByteArray.Create(1, 0, 0, 0, 0, 0, 0, 0, 0, 0),

TByteArray.Create(0, 0, 0, 0, 0, 0, 0, 0, 0, 0),

TByteArray.Create(0, 0, 1, 1, 0, 0, 1, 1, 0, 0)

);

m[6,0]:=1;

m[7,0]:=1;

m[0,6]:=1;

m[0,7]:=1;

m[0,5]:=1;

m[2,9]:=1;

m[3,9]:=1;

m[4,9]:=1;

m[5,9]:=1;

m[2,2]:=1;

m[5,5]:=1;

m[2,0]:=1;

m[3,0]:=1;

m[4,0]:=1;

m[5,2]:=1;

m[2,6]:=1;

m[9,3]:=1;

m[9,2]:=1;

m[9,6]:=1;

m[9,7]:=1;

end else

if bb=19 then

begin

for i:=0 to 9 do

for j:=0 to 9 do

begin

m[i,j]:=0;

dop[i,j]:=0;

end;

getMap := TByteArray2D.Create(

TByteArray.Create(0, 0, 0, 0, 1, 1, 1, 1, 0, 0),

TByteArray.Create(0, 0, 0, 0, 0, 0, 0, 0, 0, 0),

TByteArray.Create(0, 0, 0, 0, 0, 0, 1, 0, 0, 1),

TByteArray.Create(1, 0, 0, 0, 0, 0, 0, 0, 0, 1),

TByteArray.Create(1, 0, 0, 0, 0, 0, 0, 0, 0, 1),

TByteArray.Create(0, 0, 0, 1, 0, 0, 0, 0, 0, 0),

TByteArray.Create(1, 0, 0, 0, 0, 0, 0, 1, 0, 1),

TByteArray.Create(1, 0, 0, 1, 0, 0, 0, 0, 0, 1),

TByteArray.Create(0, 0, 0, 0, 0, 0, 0, 0, 0, 0),

TByteArray.Create(0, 0, 1, 1, 1, 0, 0, 0, 0, 0)

);

m[0,4]:=1;

m[0,5]:=1;

m[0,6]:=1;

m[0,7]:=1;

m[2,6]:=1;

m[2,9]:=1;

m[3,9]:=1;

m[4,9]:=1;

m[3,0]:=1;

m[4,0]:=1;

m[5,3]:=1;

m[7,7]:=1;

m[6,0]:=1;

m[7,0]:=1;

m[7,3]:=1;

m[6,9]:=1;

m[7,9]:=1;

m[9,2]:=1;

m[9,3]:=1;

m[9,4]:=1;

end else

begin //последюущие расстановки. Палубники смещаются на 1 по x и y, а остальные корабли немного по-другому.

for i:=0 to 9 do begin //боковые грани обмениваются кораблями

dop[0,i]:=m[0,i];

dop[9,i]:=m[9,i];

dop[i,0]:=m[i,0];

dop[i,9]:=m[i,9];

end;

for i:=0 to 9 do

begin

m[0,i]:=0;

m[9,i]:=0;

m[i,0]:=0;

m[i,9]:=0;

end;

if tt=false then

for i:=0 to 9 do

begin

m[0,i]:=dop[9,i];

m[9,i]:=dop[0,i];

m[i,0]:=dop[i,9];

m[i,9]:=dop[i,0];

tt:=true;

end else

for i:=0 to 9 do

begin

m[0,i]:=dop[i,9];

m[9,i]:=dop[i,0];

m[i,0]:=dop[9,i];

m[i,9]:=dop[0,i];

tt:=false;

end;

for i:=7 downto 2 do //парусники

for j:=7 downto 2 do

begin

if m[i,j]=1 then

begin

m[i,j]:=0;

if (j=7) and (i=7) then

m[2,2]:=1 else

if j=7 then

m[i+1,2]:=1 else

if i=7 then

m[2,j+1]:=1 else

m[i+1,j+1]:=1;

end;

end; //конечное значение getMap;

getMap := TByteArray2D.Create(

TByteArray.Create(m[0,0], m[0,1],m[0,2], m[0,3], m[0,4], m[0,5], m[0,6], m[0,7], m[0,8], m[0,9]),

TByteArray.Create(m[1,0], m[1,1],m[1,2], m[1,3], m[1,4], m[1,5], m[1,6], m[1,7], m[1,8], m[1,9]),

TByteArray.Create(m[2,0], m[2,1],m[2,2], m[2,3], m[2,4], m[2,5], m[2,6], m[2,7], m[2,8], m[2,9]),

TByteArray.Create(m[3,0], m[3,1],m[3,2], m[3,3], m[3,4], m[3,5], m[3,6], m[3,7], m[3,8], m[3,9]),

TByteArray.Create(m[4,0], m[4,1],m[4,2], m[4,3], m[4,4], m[4,5], m[4,6], m[4,7], m[4,8], m[4,9]),

TByteArray.Create(m[5,0], m[5,1],m[5,2], m[5,3], m[5,4], m[5,5], m[5,6], m[5,7], m[5,8], m[5,9]),

TByteArray.Create(m[6,0], m[6,1],m[6,2], m[6,3], m[6,4], m[6,5], m[6,6], m[6,7], m[6,8], m[6,9]),

TByteArray.Create(m[7,0], m[7,1],m[7,2], m[7,3], m[7,4], m[7,5], m[7,6], m[7,7], m[7,8], m[7,9]),

TByteArray.Create(m[8,0], m[8,1],m[8,2], m[8,3], m[8,4], m[8,5], m[8,6], m[8,7], m[8,8], m[8,9]),

TByteArray.Create(m[9,0], m[9,1],m[9,2], m[9,3], m[9,4], m[9,5], m[9,6], m[9,7], m[9,8], m[9,9])

);

end;

vopros:=true;

bb:=bb+1;

end;

procedure vokrug(u:integer); //Вычеркивание клеток, находящизся рядом с убитым кораблём

begin

for i:=0 to u do

begin

if (coord[i]=1) or (coord[i]=2) or (coord[i]=3) or (coord[i]=4) or (coord[i]=5) or (coord[i]=6) or (coord[i]=7) or (coord[i]=8) then

begin

visited[coord[i]+1]:=true;

visited[coord[i]-1]:=true;

visited[coord[i]+9]:=true;

visited[coord[i]+10]:=true;

visited[coord[i]+11]:=true;

end else

if (coord[i]=91) or (coord[i]=92) or (coord[i]=93) or (coord[i]=94) or (coord[i]=95) or (coord[i]=96) or (coord[i]=97) or (coord[i]=98) then

begin

visited[coord[i]+1]:=true;

visited[coord[i]-1]:=true;

visited[coord[i]-9]:=true;

visited[coord[i]-10]:=true;

visited[coord[i]-11]:=true;

end else

if (coord[i]=10) or (coord[i]=20) or (coord[i]=30) or (coord[i]=40) or (coord[i]=50) or (coord[i]=60) or (coord[i]=70) or (coord[i]=80) then

begin

visited[coord[i]+10]:=true;

visited[coord[i]-10]:=true;

visited[coord[i]+1]:=true;

visited[coord[i]-9]:=true;

visited[coord[i]+11]:=true;

end else

if (coord[i]=19) or (coord[i]=29) or (coord[i]=39) or (coord[i]=49) or (coord[i]=59) or (coord[i]=69) or (coord[i]=79) or (coord[i]=89) then

begin

visited[coord[i]-1]:=true;

visited[coord[i]-10]:=true;

visited[coord[i]+10]:=true;

visited[coord[i]-11]:=true;

visited[coord[i]+9]:=true;

end else

if coord[i]=0 then

begin

visited[coord[i]+1]:=true;

visited[coord[i]+10]:=true;

visited[coord[i]+11]:=true;

end else

if coord[i]=9 then

begin

visited[coord[i]-1]:=true;

visited[coord[i]+10]:=true;

visited[coord[i]+9]:=true;

end else

if coord[i]=90 then

begin

visited[coord[i]+1]:=true;

visited[coord[i]-10]:=true;

visited[coord[i]-9]:=true;

end else

if coord[i]=99 then

begin

visited[coord[i]-1]:=true;

visited[coord[i]-10]:=true;

visited[coord[i]-11]:=true;

end else

begin

visited[coord[i]+1]:=true;

visited[coord[i]-1]:=true;

visited[coord[i]+10]:=true;

visited[coord[i]-10]:=true;

visited[coord[i]+9]:=true;

visited[coord[i]+11]:=true;

visited[coord[i]-9]:=true;

visited[coord[i]-11]:=true;

end;

end;

end;

function shoot: TByteArray; //Используем тактику стрельбы по диагоналям

var i:integer;

begin

if (aa=true) and (uu=1) then //Добиваем корабль при попадании

begin

kek:=coord[0]+1;

if (kek div 10 = coord[0] div 10) and (kek<=99) then

if (visited[kek]=false) then

begin

coord[1]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

povorot:=2;

exit;

end;// else

kek:=coord[0]-1;

if (kek div 10 = coord[0] div 10) and (kek>=0)

if (visited[kek]=false) then

begin

coord[1]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

povorot:=1;

exit;

end;// else

kek:=coord[0]+10;

if (kek<=99) then

if (visited[kek]=false) then

begin

coord[1]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

povorot:=4;

exit;

end;// else

kek:=coord[0]-10;

if (kek>=0) then

if (visited[kek]=false) then

begin

coord[1]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

povorot:=3;

exit;

end;

end;// else

if (aa=true) and (uu=2) then

begin

if (povorot=1) or (povorot=2) then

begin

kek:=coord[1]+1;

if (kek div 10 = coord[0] div 10) and (kek<=99) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[1]-1;

if (kek div 10 = coord[0] div 10) and (kek>=0)

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[0]+1;

if (kek div 10 = coord[0] div 10) and (kek<=99) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[0]-1;

if (kek div 10 = coord[0] div 10) and (kek>=0) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

end;// else

if (povorot=3) or (povorot=4) then

begin

kek:=coord[1]+10;

if (kek<=99) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[1]-10;

if (kek>=0) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[0]+10;

if (kek<=99) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[0]-10;

if (kek>=0) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

end;

end;// else

if (aa=true) and (uu=3) then //Для 4-хклеточного корабля

begin

if (povorot=1) or (povorot=2) then

begin

kek:=coord[1]+1;

if (kek div 10 = coord[0] div 10) and (kek<=99) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[1]-1;

if (kek div 10 = coord[0] div 10) and (kek>=0) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[0]+1;

if (kek div 10 = coord[0] div 10) and (kek<=99) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[0]-1;

if (kek div 10 = coord[0] div 10) and (kek>=0) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[2]+1;

if (kek div 10 = coord[0] div 10) and (kek<=99) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[2]-1;

if (kek div 10 = coord[0] div 10) and (kek>=0) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

end;// else

if (povorot=3) or (povorot=4) then

begin

kek:=coord[1]+10;

if (kek<=99) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[1]-10;

if (kek>=0) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[0]+10;

if (kek<=99) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[0]-10;

if (kek>=0) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[2]+10;

if (kek<=99) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;// else

kek:=coord[2]-10;

if (kek>=0) then

if (visited[kek]=false) then

begin

coord[2]:=kek;

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

end;

end;// else//переход к основному

kek:=10; //Обстрел карты по диагоналям

for i:=1 to 2 do begin

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end else

kek:=kek-9;

end;

kek:=98;

for i:=1 to 2 do begin

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end else

kek:=kek-9;

end;

kek:=30;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=3;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=96;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=69;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=50;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=5;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=94;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=49;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=70;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=7;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=92;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=29;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=90;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

kek:=9;

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end;

verh:=true;

a:=3;

b:=1;

kek:=30;

for i:=1 to 4 do begin

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end else

begin

if verh=true then

begin

kek:=30-9\*a;

a:=a-1;

verh:=false;

end else

begin

kek:=30-9\*b;

b:=b+1;

verh:=true;

end;

end;

end;

verh:=true;

a:=3;

b:=1;

kek:=96;

for i:=1 to 4 do begin

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end else

begin

if verh=true then

begin

kek:=96-9\*a;

a:=a-1;

verh:=false;

end else

begin

kek:=96-9\*b;

b:=b+1;

verh:=true;

end;

end;

end;

verh:=true;

a:=5;

b:=1;

kek:=50;

for i:=1 to 6 do begin

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end else

begin

if verh=true then

begin

kek:=50-9\*a;

a:=a-1;

verh:=false;

end else

begin

kek:=50-9\*b;

b:=b+1;

verh:=true;

end;

end;

end;

verh:=true;

a:=5;

b:=1;

kek:=94;

for i:=1 to 6 do begin

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end else

begin

if verh=true then

begin

kek:=94-9\*a;

a:=a-1;

verh:=false;

end else

begin

kek:=94-9\*b;

b:=b+1;

verh:=true;

end;

end;

end;

verh:=true;

a:=7;

b:=1;

kek:=70;

for i:=1 to 8 do begin

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end else

begin

if verh=true then

begin

kek:=70-9\*a;

a:=a-1;

verh:=false;

end else

begin

kek:=70-9\*b;

b:=b+1;

verh:=true;

end;

end;

end;

verh:=true;

a:=7;

b:=1;

kek:=92;

for i:=1 to 8 do begin

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end else

begin

if verh=true then

begin

kek:=92-9\*a;

a:=a-1;

verh:=false;

end else

begin

kek:=92-9\*b;

b:=b+1;

verh:=true;

end;

end;

end;

verh:=true;

a:=9;

b:=1;

kek:=90;

for i:=1 to 10 do begin

if visited[kek]=false then

begin

shoot:=TByteArray.Create(kek mod 10, kek div 10);

visited[kek]:=true;

exit;

end else

begin

if verh=true then

begin

kek:=90-9\*a;

a:=a-1;

verh:=false;

end else

begin

kek:=90-9\*b;

b:=b+1;

verh:=true;

end;

end;

end;

for i:=0 to 99 do begin //Обстреливание оставшейся карты после диагоналей

if visited[i]=false then

begin

shoot:=TByteArray.Create(i mod 10, i div 10);

visited[i]:=true;

exit;

end;

end;

end;

procedure shootResult(resultCode: integer);

begin

if resultCode=DAMAGE then

begin

if aa=false then

coord[0]:=kek;

uu:=uu+1;

aa:=true; //присваивать coord[uu]:=kek на каждом попадании

end;

if resultCode=KILL then

begin

if uu=0 then

begin

coord[0]:=kek;

vokrug(uu);

aa:=false;

uu:=0;

povorot:=0;

for i:=0 to 3 do

coord[i]:=0;

end;

if uu=1 then

begin

coord[1]:=kek;

vokrug(uu);

aa:=false;

uu:=0;

povorot:=0;

for i:=0 to 3 do

coord[i]:=0;

end;

if uu=2 then

begin

coord[2]:=kek;

vokrug(uu);

aa:=false;

uu:=0;

povorot:=0;

for i:=0 to 3 do

coord[i]:=0;

end;

if uu=3 then

begin

coord[3]:=kek;

vokrug(uu);

aa:=false;

uu:=0;

povorot:=0;

for i:=0 to 3 do

coord[i]:=0;

end;

end;

end;

procedure rivalShoot(point: TByteArray);

begin

end;

procedure setParams(setsPerGame: integer);

begin

for i:=0 to 99 do

visited[i]:=false;

bb:=0;

end;

procedure onCurrentSetEnd;

begin

for i:=0 to 99 do

visited[i]:=false;

end;

procedure onCurrentGameEnd;

begin

tt:=false;

for i:=0 to 99 do

visited[i]:=false;

bb:=0;

for i:=0 to 9 do

for j:=0 to 9 do

begin

m[i,j]:=0;

dop[i,j]:=0;

end;

end;

end.

**Вывод**

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