**Міністерство освіти і науки України**

**Полтавський національний технічний університет**

**імені Юрія Кондратюка**

**Факультет інформаційних та телекомунікаційних технологій і систем**

**Кафедра комп’ютерних та інформаційних технологій і систем**

**Курсова робота**

з дисципліни «Основи програмування та алгоритмічні мови»

КР \_\_\_\_\_\_

Розробив cтудент

групи 101-ТН

Керівник роботи

Полтава 2010

**Зміст**

Вступ

1. Постановка задачі

2. Розв’язання задачі

3. Алгоритм задачі

4. Реалізація програми 8

5. Демонстрація роботи програми

Висновок

Використана література

# Вступ

Щоб виконати дану роботу треба мати достатні знання мови програмування Турбо Паскаль, а також вміння складати алгоритми. Для коду даної програми я використав навики з виконання практичних робіт.

Гру «Життя» вперше створив Дж. Конвей. Це власне не гра, а проста модель еволюції групи живих організмів. Пасивне спостереження над сукупністю великої кількості поколінь цієї гри може дати якусь задоволення.

Ігровим полем(«житєвим простором») в цій грі являється двохвимірна поверхня, розділена на квадрати. В нас це поверхня тора, яка не має границь: наявність границь завжди зв’язано з проблемами. Кожна клітинка має 8 сосідів. Ячейка може бути заселена одним організмом, або бути порожньою.

# Постановка задачі

Розробити програму для вирішення поставленої прикладної задачі. При вирішенні задачі необхідно:

* Створити програму у середовищі програмування Turbo Pascal
* Розробити алгоритм функціонування програми
* Надати блок-схему алгоритму
* Продемонструвати роботу програми

**УМОВА ЗАДАЧІ**

Гра моделює життя поколінь гіпотетичної колонії живих кліток,які виживають, розмножуються або гинуть у відповідності з наступними правилами.

Клітка виживає, якщо має двох або трьох сусідів з восьми можливих, якщо один або жодного то клітка гине в ізоляції. Якщо має чотири та більше, то гине від перенаселення.У будь-якій порожній позиції у якої рівно три сусіди у наступному поколінні з'являється нова клітка.

## Розв’язання задачі

## Популяцію в першому поколінні задамо випадково. Це означає, що рішення про те, чи буде кожна клітка заселена, приймається з деякою вірогідністю. У програмі life при відображенні групи віртуальних організмів використана анімація. Ефект анімації оснований на послідовній зміні кадрів – графічних зображень. Для швидкого переходу від одного «кадра» до іншого використаємо дві графічні сторінки. Графічна сторінка – область відеопам’яті, яка зберігає зображення. При цьому, якщо режим роботи відеоадаптера підтримує роботу тільки з однією сторінкою, її вміст відображається на екрані. В деяких режимах роботи підтримуються декілька графічних сторінок. Вміст однієї з них («візуальної») відображається не на екрані, а на другій («активній»), невидимій користувачу, в цей час може будуватися нове зображення. Підготовлене на новій сторінці зображення виводиться на екран. Такий спосіб можливий, наприклад, про роботі в режимі VGA з роздільною здатність 640х480, бо в цьому режимі є дві графічні сторінки. В розділі опису констант задані параметри програми, які якщо доведеться змінювати, то рідко, тому некомфортно їх вводити при кожному запуску програми:

Hor – кількість кліток по горизонталі

Ver – кількість кліток по вертикалі

Cell\_width, cell\_height – ширина та висота клітки

Prob\_factor – параметр, що визначає вірогідність заселення кліток про формуванні початкової популяції.

Процедура init\_cells формує початкову популяцію, встановлюючи елементам масива new\_gen значення 0 або 1 з використаним параметром вірогідності.

Процедура next\_generation будує в масиві new\_gen будує наступну популяцію за правилами.

Процедура init\_screen ініціалізує графічний режим та заповнює масиви графічних координат центрів кліток. Процедура display виконує основну роботу по побудові зображення робочого поля гри на прихованій активній сторінці та виводить на екран. Це уможливлюють процедури SetActivePage та SetVisualPage, викликані з параметром 0 або 1, визначаючим номер графічної сторінки. Перша процедура при викликанні встановлює активну сторінку, а друга візуальну.

* 1. **Алгоритм задачі**

**А**) Головний алгоритм



Б) Алгоритм процедури **init\_cells**

****

В) Алгоритм процедури **next\_generation**

****

Г) Алгоритм процедури **init\_screen**



Д) Алгоритм процедури **display**



Е) Алгоритм процедури **rule\_plane**



Реалізація програми

PROGRAM LIFE;

uses crt,dos,graph;

const

hor=100;

ver=70;

cell\_width=8;

cell\_height=6;

prob\_factor=0.5;

var

old\_gen,new\_gen:array[0..ver,0..hor] of 0..1;

prob:real;

ch:char;

x\_center:array[0..hor] of word;

y\_center:array[0..hor] of word;

gen\_count,radius,page:word;

ss:string[10];

procedure init\_cells;

var

j,k:word;

begin

gen\_count:=0;

for j:=0 to ver do

for k:=0 to hor do

begin

old\_gen[j,k]:=0;

if random<=prob then

new\_gen[j,k]:=1

else

new\_gen[j,k]:=0;

end;

end;

procedure next\_generation;

var

j,k,m,prev\_j,next\_j,prev\_k,next\_k:word;

begin

old\_gen:=new\_gen;

for j:=0 to ver do

begin

if j=0 then

prev\_j:=ver

else

prev\_j:=j-1;

if j=ver then

next\_j:=0

else

next\_j:=j+1;

for k:=0 to hor do

begin

if k=0 then

prev\_k:=ver

else

prev\_k:=k-1;

if k=hor then

next\_k:=0

else

next\_k:=k+1;

m:=old\_gen[prev\_j,prev\_k]

+old\_gen[prev\_j,k]

+old\_gen[prev\_j,next\_k]

+old\_gen[j,prev\_k]

+old\_gen[j,next\_k]

+old\_gen[next\_j,prev\_k]

+old\_gen[next\_j,k]

+old\_gen[next\_j,next\_k];

if (old\_gen[j,k]=1) and ((m<=1) or (m>=4)) then

new\_gen[j,k]:=0

else

if (old\_gen[j,k]=0) and (m=3) then

new\_gen[j,k]:=1

else

new\_gen[j,k]:=old\_gen[j,k];

end;

end;

end;

procedure init\_screen;

var

graphdriver,graphmode:integer;

j,k:word;

begin

graphdriver:=vga;

graphmode:=vgamed;

page:=0;

initgraph(graphdriver,graphmode,'');

if graphresult<>grOk then

halt;

for k:=0 to hor do

x\_center[k]:=k\*cell\_width+cell\_width div 2;

for j:=0 to ver do

y\_center[j]:=j\*cell\_height+cell\_height div 2;

radius:=4;

end;

procedure display;

var

j,k:word;

procedure rule\_plane;

var

j,k:word;

begin

setviewport(0,0,getmaxx,getmaxy,clipon);

setfillstyle(solidfill,blue);

bar(0,0,getmaxx,10);

setcolor(white);

outtext('Generation: ');

outtextxy(250,0,'Q: Quit');

outtextxy(450,0,'Any other key: renew');

str(gen\_count,ss); outtext(ss);

setbkcolor(darkgray);

end;

begin

if gen\_count<>0 then

next\_generation;

inc(gen\_count);

page:=1-page;

setactivepage(page);

cleardevice;

setcolor(yellow);

for j:=0 to ver do

for k:=0 to hor do

if new\_gen[j,k]=1 then

circle(x\_center[k],y\_center[j],radius);

rule\_plane;

setvisualpage(page);

end;

begin

init\_screen;

repeat

randomize;

prob:=0.1+prob\_factor\*random;

outtextxy(0,0,'Game of life');

writeln;

outtextxy(0,15,'Live cells inserted at random,');

str(prob:3:3,ss);

outtextxy(0,30,'with probability '+ss);

outtextxy(0,60,'Press any key to start: ');

ch:=readkey;

cleardevice;

init\_cells;

repeat

display;

if keypressed then

begin

ch:=readkey;

break;

end;

until false;

setviewport(0,0,getmaxx,getmaxy,clipon);

cleardevice;

setcolor(white);

if upcase(ch)='Q' then

break;

until false;

closegraph;

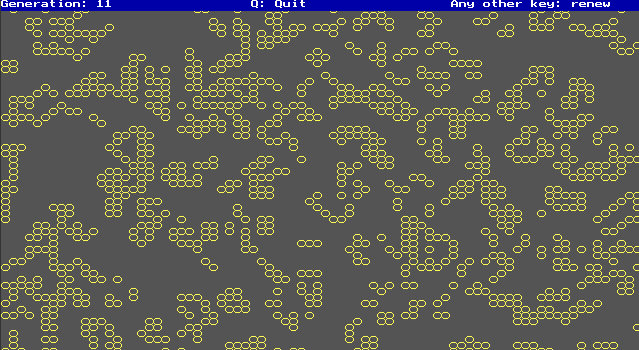
end.

* 1. **Демонстрація роботи програми**

Початкове вікно програми



Процес генерації 12 популяції



**ВИСНОВОК**

Розроблена програма дозволяє спостерігати життя, наприклад, колонії живих організмів. Я детальніше познайомився з принципами роботи в графічному режимі.

**ВИКОРИСТАНА ЛІТЕРАТУРА**

«Основи програмування»Т.В. Ковалюк

«Turbo Pascal» В.Фаронов