**Національний технічний університет України**

**«Київський політехнічний інститут»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №1**

*з дискретної математики*

*Виконав студент групи ІО-44*

***Барабаш Т.А.***

*Перевірив:*

***Новотарський М.А****.*

**2015**

**ЛАБОРАТОРНА РОБОТА №1**

*Множини: основні властивості та операції над ними, діаграми Венна*

**Мета:** вивчити основні аксіоми, закони і теореми теорії множин, навчитися застосовувати їх на практиці. Виконати наступні операції над множинами: доповнення множин; об’єднання, перетин, різниця, симетрична різниця.

**I. Завдання**

**4403 mod 10 = 3 – Варіант 3.** Літери українського алфавіту

А) Використовуючи блок-схему алгоритму, або створену самостійно блок-схему ефективнішого алгоритму, доповнити множину D до відповідної для неї універсальної множини Y. В результаті виконання доповнення сформувати множину R. Вивести елементи множин Y та R.

Б) Використовуючи блок-схему алгоритму або створену самостійно блок-схему ефективнішого алгоритму, виконати об’єднання множин A і B. В результаті виконання об’єднання сформувати множину С. Вивести елементи множини С.

В) Використовуючи блок-схему алгоритму, або створену самостійно блок-схему ефективнішого алгоритму, виконати перетин множин A і В. В результаті виконання перетину сформувати множину С. Вивести елементи множини С.

Г) Використовуючи блок-схему алгоритму, або створену самостійно блок-схему ефективнішого алгоритму, знайти різницю множин А та В. В результаті виконання перетину сформувати множину С. Вивести елементи множини С.

Д) Використовуючи блок-схему алгоритму, або створену самостійно блок-схему ефективнішого алгоритму, знайти симетричну різницю множин А та В. В результаті виконання перетину сформувати множину С. Вивести елементи множини С.

Е) Побудувати діаграми Венна для кожної з операцій з виділенням результуючої підмножини.

**ІI. Короткі теоретичні відомості**

**Множина** - сукупність визначених об'єктів, різних між собою, об’єднаних за певною ознакою чи властивістю.

**Об’єднання**. Об'єднанням множин A і B називається множина, що складається із всіх тих елементів, які належать хоча б однієї з множин A або B.

**Перетин.** Перетином множин A і B називається множина, що складається із всіх тих елементів, які належать і множині A й множині B .

**Доповнення.** Доповненням (або абсолютним доповненням) множини A називається множина, що складається із всіх елементів універсальної множини, які не належать A.

**Різниця.** Різницею множин A й B (або відносним доповненням) називається множина, що складається із всіх елементів множини A, які не належать B.

**Симетрична різниця.** Симетричною різницею множин A і B називається множина, що складається з об'єднання всіх елементів, що належать множині A і не містяться в B, і елементів, що належать множині B і не містяться в A.

**ІIІ. Алгоритми**

**Базові алгоритми**

|  |  |
| --- | --- |
| ***Процедура AddToArray*** – доповнення множини | ***Процедура Minus*** – різниця множин |
|  |  |

**Алгоритми операцій**

**Алгоритм доповнення множини** *(на прикладі множини А)*

**Алгоритм операції перетину**



**Алгоритм операції різниці множин** *(на прикладі різниці множин А та В)*



**Алгоритм симетричної різниці множин А та В**



**Алгоритм об’єднання множин**



**VІ. Код програми**

unit Unit12;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls,

ExtCtrls, Grids;

type Tarray = Array [1..100] of String;

{ TOperForm }

TOperForm = class(TForm)

Agrid: TStringGrid;

BGrid: TStringGrid;

ComplementA: TButton;

cClear: TButton;

ExitBtn: TButton;

ComplementB: TButton;

Img: TImage;

Label4: TLabel;

MinusBA: TButton;

Infopanel: TStaticText;

UkrGrid: TStringGrid;

Union: TButton;

Intersection: TButton;

MinusAB: TButton;

SymMinus: TButton;

aClear: TButton;

aLoadFile: TButton;

bClear: TButton;

bLoadFile: TButton;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Panel1: TPanel;

CGrid: TStringGrid;

procedure aClearClick(Sender: TObject);

procedure Minus(arrA:TArray; countA:Integer; arrB:TArray; countB:Integer; var arrC:TArray; var countC:Integer);

procedure Peretun(arrA:TArray; countA:Integer; arrB:TArray; countB:Integer; var arrC:TArray; var countC:Integer);

procedure aLoadFileClick(Sender: TObject);

procedure bClearClick(Sender: TObject);

procedure bLoadFileClick(Sender: TObject);

procedure cClearClick(Sender: TObject);

procedure ComplementAClick(Sender: TObject);

procedure ComplementBClick(Sender: TObject);

procedure ExitBtnClick(Sender: TObject);

procedure IntersectionClick(Sender: TObject);

procedure MinusABClick(Sender: TObject);

procedure MinusBAClick(Sender: TObject);

procedure SymMinusClick(Sender: TObject);

procedure UnionClick(Sender: TObject);

private

procedure ClearGrid(M:String);

procedure LoadFromFile(M:String);

{ private declarations }

public

{ public declarations }

end;

Type T3Arr = Array [1..10,1..10] of String;

var

OperForm: TOperForm;

Var LocalPath:String;

Const MaxCol=10; MaxRow=7;

var

A3GridCol,A3GridRow,B3GridCol,B3GridRow:Integer;

A3Arr,B3Arr:T3Arr;

implementation

{$R \*.lfm}

{ TOperForm }

procedure addToArray(var arr: TArray; value:String; var count:Integer);

var i:integer;

q:boolean;

begin

q:=true;

for i:=1 to count do

if arr[i] = value then begin

q:=false;

break;

end;

if q then begin

inc(count);

arr[count]:=value;

end;

end;

procedure TOperForm.LoadFromFile(M:String);

var F : Text;

LogFileName,Str:String;

begin

ClearGrid(M);

LogFileName:=LocalPath+'DATA\'+M+'3.TXT';

AssignFile(F,LogFileName);

{$I-} Reset(F); {$I+}

if IOResult <> 0 then

begin

InfoPanel.Caption:='Невозможно прочитать из '+LogFileName;

Exit;

end;

If M='A' then

begin

A3GridCol:=0;

A3GridRow:=1;

While not EOF(F) do

begin

If A3GridCol<MaxCol then Inc(A3GridCol) else

begin

A3GridCol:=1;

Inc(A3GridRow);

If A3GridRow>MaxRow then

begin

InfoPanel.Caption:='Дані з файлу 3А.TXT не помістилися в таблицю';

CloseFile(F);

Exit;

end;

end;

Readln(F, Str); Agrid.Cells[A3GridCol,A3GridRow]:=Str;

end;

end;

If M='B' then

begin

B3GridCol:=0;

B3GridRow:=1;

While not EOF(F) do

begin

If B3GridCol<MaxCol then Inc(B3GridCol) else

begin

B3GridCol:=1;

Inc(B3GridRow);

If A3GridRow>MaxRow then

begin

InfoPanel.Caption:='Дані з файлу 3B.TXT не помістилися в таблицю';

CloseFile(F);

Exit;

end;

end;

Readln(F, Str); BGrid.Cells[B3GridCol,B3GridRow]:=Str;

end;

end;

CloseFile(F);

end;

procedure ToperForm.ClearGrid(M: String);

var i,j:Integer;

begin

If M='A' then

For i:=1 to MaxCol do

For j:=1 to MaxRow do Agrid.Cells[i,j]:='';

If M='B' then

For i:=1 to MaxCol do

For j:=1 to MaxRow do BGrid.Cells[i,j]:='';

If M='C' then

For i:=1 to MaxCol do

For j:=1 to MaxRow do CGrid.Cells[i,j]:='';

A3GridCol:=0;

A3GridRow:=1;

B3GridCol:=0;

B3GridRow:=1;

end;

procedure TOperForm.ExitBtnClick(Sender: TObject);

begin

Close;

end;

procedure ToperForm.aLoadFileClick(Sender: TObject);

begin

LoadFromFile('A');

InfoPanel.Caption := 'Завантажено з файлу множину А';

Img.Picture.LoadFromFile ('Img/Zav\_A.jpg');

end;

procedure TOperForm.bClearClick(Sender: TObject);

begin

ClearGrid('B');

InfoPanel.Caption := 'Очищено множину B';

Img.Picture.LoadFromFile('Img/Och\_B.jpg');

end;

procedure TOperForm.bLoadFileClick(Sender: TObject);

begin

LoadFromFile('B');

InfoPanel.Caption := 'Завантажено з файлу множину В';

Img.Picture.LoadFromFile ('Img/Zav\_B.jpg');

end;

procedure TOperForm.cClearClick(Sender: TObject);

begin

ClearGrid('C');

InfoPanel.Caption := 'Результат очищено';

Img.Picture.LoadFromFile('Img/Och\_C.jpg');

end;

procedure TOperForm.Minus(arrA:TArray; countA:Integer; arrB:TArray; countB:Integer; var arrC:TArray; var countC:Integer);

var

i,j:Integer;

q:boolean;

begin

countC:=0;

for i := 1 to countA do

begin

q := true;

for j := 1 to countB do

if( arrA[i] = arrB[j]) then

q :=false;

if(q) then

begin

INC(countC);

arrC[countC] := arrA[i];

end;

end;

end;

procedure TOperForm.ComplementAClick(Sender: TObject);

var

i,j:Integer;

AllArr, loadedArr, result:TArray;

countA, countB, countC:Integer;

begin

ClearGrid('C');

For i:=1 to maxcol do

For j:=1 to maxrow do

begin

AllArr[(j-1)\*10 + i]:=UKRGrid.Cells[i,j];

end;

countA := maxcol\*maxrow;

countB:=0;

For j:=1 to maxrow do

For i:=1 to maxcol do

begin

if (Agrid.Cells[i, j] <> '') then

addToArray(loadedArr, Agrid.Cells[i, j], countB);

end;

minus(AllArr, countA, loadedArr, countB, result, countC);

i := 0;

while ((i < countC) AND (i < maxcol\*maxrow)) do begin

Cgrid.Cells[(i MOD 10) + 1, (i DIV 10) + 1] := result[i + 1];

INC(i);

end;

InfoPanel.Caption := 'Діаграмма Ейлера-Венна для операції доповненя множини А';

Img.Picture.LoadFromFile('Img/Dop\_A.jpg');

end;

procedure TOperForm.ComplementBClick(Sender: TObject);

var

i,j:Integer;

AllArr, loadedArr, result:TArray;

countA, countB, countC:Integer;

begin

ClearGrid('C');

For i:=1 to MaxCol do

For j:=1 to MaxRow do

begin

AllArr[(j-1)\*10 + i]:=UKRGrid.Cells[i,j];

end;

countA := maxcol\*maxrow;

countB:=0;

For j:=1 to MaxROw do

For i:=1 to MaxCol do

begin

if (Bgrid.Cells[i, j] <> '') then

addToArray(loadedArr, Bgrid.Cells[i, j], countB);

end;

minus(AllArr, countA, loadedArr, countB, result, countC);

i := 0;

while ((i < countC) AND (i < maxcol\*maxrow)) do begin

Cgrid.Cells[(i MOD 10) + 1, (i DIV 10) + 1] := result[i + 1];

INC(i);

end;

InfoPanel.Caption := 'Діаграмма Ейлера-Венна для операції доповненя множини В';

Img.Picture.LoadFromFile('Img/Dop\_B.jpg');

end;

procedure TOperForm.aClearClick(Sender: TObject);

begin

ClearGrid('A');

InfoPanel.Caption:= 'Очищено множину А';

Img.Picture.LoadFromFile('Img/Och\_A.jpg');

end;

procedure TOperForm.Peretun(arrA:TArray; countA:Integer; arrB:TArray; countB:Integer; var arrC:TArray; var countC:Integer);

var

i,j:Integer;

begin

countC:=0;

for i := 1 to countA do

for j := 1 to countB do

if( arrA[i] = arrB[j]) then

begin

INC(countC);

arrC[countC] := arrA[i];

end;

end;

procedure TOperForm.IntersectionClick(Sender: TObject);

var

i,j:Integer;

arrA, arrB, arrC:TArray;

countA, countB, countC:Integer;

begin

countA :=0;

countB :=0;

ClearGrid('C');

For j:=1 to maxrow do

For i:=1 to maxcol do

begin

if (Agrid.Cells[i, j] <> '') then

addToArray(arrA, agrid.Cells[i, j], countA);

if (bgrid.Cells[i, j] <> '') then

addToArray(arrB, bgrid.Cells[i, j], countB);

end;

Peretun(arrA, countA, arrB, countB, arrC, countC);

i := 0;

while ((i < countC) AND (i < maxcol\*maxrow)) do begin

CGrid.Cells[(i MOD 10) + 1, (i DIV 10) + 1] := arrC[i + 1];

INC(i);

end;

InfoPanel.Caption := 'Діаграма Ейлера-Венна для операції перетину множин А та В';

Img.Picture.LoadFromFile('Img/int.jpg');

end;

procedure TOperForm.MinusABClick(Sender: TObject);

var

i,j:Integer;

arrA, arrB, arrC:TArray;

countA, countB, countC:Integer;

begin

countA :=0;

countB :=0;

ClearGrid('C');

For j:=1 to maxrow do

For i:=1 to maxcol do

begin

if (AGrid.Cells[i, j] <> '') then

addToArray(arrA, AGrid.Cells[i, j], countA);

if (BGrid.Cells[i, j] <> '') then

addToArray(arrB, BGrid.Cells[i, j], countB);

end;

minus(arrA, countA, arrB, countB, arrC, countC);

i := 0;

while ((i < countC) AND (i < maxcol\*maxrow)) do begin

CGrid.Cells[(i MOD 10) + 1, (i DIV 10) + 1] := arrC[i + 1];

INC(i);

end;

InfoPanel.Caption := 'Діаграма Ейлера-Венна для операції різниці множин А та В';

Img.Picture.LoadFromFile('Img/MinusAB.jpg');

end;

procedure TOperForm.MinusBAClick(Sender: TObject);

var

i,j:Integer;

arrA, arrB, arrC:TArray;

countA, countB, countC:Integer;

begin

countA :=0;

countB :=0;

ClearGrid('C');

For j:=1 to maxrow do

For i:=1 to maxcol do

begin

if (AGrid.Cells[i, j] <> '') then

addToArray(arrA, AGrid.Cells[i, j], countA);

if (BGrid.Cells[i, j] <> '') then

addToArray(arrB, BGrid.Cells[i, j], countB);

end;

minus(arrB, countB, arrA, countA, arrC, countC);

i := 0;

while ((i < countC) AND (i < maxcol\*maxrow)) do begin

CGrid.Cells[(i MOD 10) + 1, (i DIV 10) + 1] := arrC[i + 1];

INC(i);

end;

InfoPanel.Caption := 'Діаграма Ейлера-Венна для операції різниці множин В та А';

Img.Picture.LoadFromFile('Img/MinusBA.jpg');

end;

procedure TOperForm.SymMinusClick(Sender: TObject);

var

i,j:Integer;

arrA, arrB, arrC1, arrC2:TArray;

countA, countB, countC1, countC2:Integer;

begin

countA :=0;

countB :=0;

ClearGrid('C');

For j:=1 to maxrow do

For i:=1 to maxcol do

begin

if (AGrid.Cells[i, j] <> '') then

addToArray(arrA, AGrid.Cells[i, j], countA);

if (BGrid.Cells[i, j] <> '') then

addToArray(arrB, BGrid.Cells[i, j], countB);

end;

minus(arrA, countA, arrB, countB, arrC1, countC1);

minus(arrB, countB, arrA, countA, arrC2, countC2);

for i:=1 to countC2 do

begin

INC(countC1);

arrC1[countC1] := arrC2[i];

end;

i := 0;

while ((i < countC1) AND (i < maxcol\*maxrow)) do begin

CGrid.Cells[(i MOD 10) + 1, (i DIV 10) + 1] := arrC1[i + 1];

INC(i);

end;

InfoPanel.Caption := 'Діаграма Ейлера-Венна для операції симетричної різниці А та В';

Img.Picture.LoadFromFile('Img/SymMinus.jpg');

end;

procedure TOperForm.UnionClick(Sender: TObject);

var

i,j,count : Integer;

arr : TArray;

begin

count := 0;

ClearGrid('C');

For j:=1 to maxrow do

For i:=1 to maxcol do

if (AGrid.Cells[i, j] <> '') then

addToArray(arr, AGrid.Cells[i, j], count);

For j:=1 to maxrow do

For i:=1 to maxcol do

if (BGrid.Cells[i, j] <> '') then

addToArray(arr, BGrid.Cells[i, j], count);

i := 0;

while ((i < count) AND (i < maxcol\*maxrow)) do begin

CGrid.Cells[(i MOD 10) + 1, (i DIV 10) + 1] := arr[i + 1];

INC(i);

end;

InfoPanel.Caption := 'Діаграма Ейлера-Венна для операції об`єднанання множин А та В';

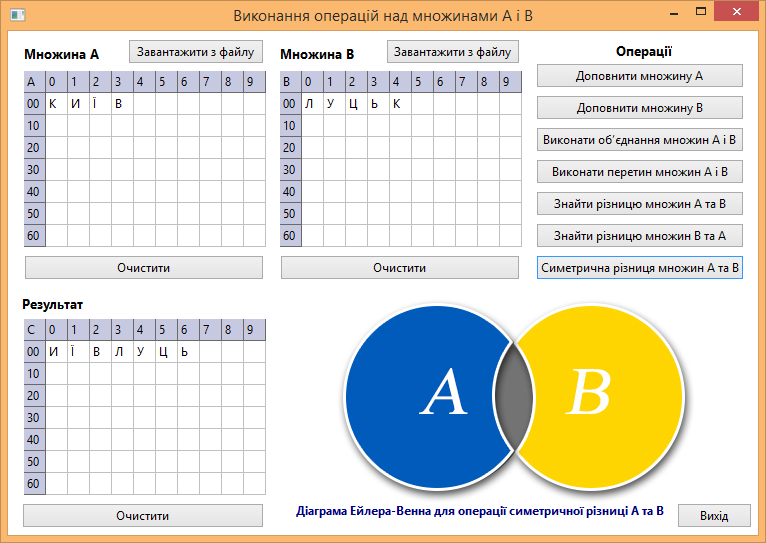
Img.Picture.LoadFromFile('Img/Uni.jpg');

end;

end.

**V. Результат**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



**VI. Висновок та аналіз результатів**

У ході лабораторної роботи було виконано операції над множинами, а саме: доповнення, об’єднання, перетин, різниця та симетрична різниця. Для цих операцій було створено просту програму з візуальною частиною.

Для спрощення завдання та зменшення кількості підрахунків, множини було представлено у вигляді одномірних масивів, оскільки саме у такому вигляді найкраще реалізовувати алгоритми операцій над множинами.