# Ðàçðàáîòêà êëàññà ïðÿìîóãîëüíèêîâ

ÔÃÎÓ ÂÏÎ Êàì÷àòñêèé Ãîñóäàðñòâåííûé Òåõíè÷åñêèé Óíèâåðñèòåò

Ôàêóëüòåò èíôîðìàöèîííûõ òåõíîëîãèé

Êàôåäðà ñèñòåì óïðàâëåíèÿ

Îáúåêòíî-îðèåíòèðîâàííîå ïðîãðàììèðîâàíèå

Êóðñîâàÿ ðàáîòà

Òåìà: Ðàçðàáîòêà êëàññà ïðÿìîóãîëüíèêîâ

Ïåòðîïàâëîâñê-Êàì÷àòñêèé

**Ââåäåíèå**

Öåëü ðàáîòû – ñîçäàíèå êëàññà ïðÿìîóãîëüíèêîâ ñî ñòîðîíàìè ïàðàëåëüíûìè îñÿì êîîðäèíàò.

Çàäà÷åé êóðñîâîãî ïðîåêòèðîâàíèÿ ÿâëÿåòñÿ ðàçðàáîòêà ïðîãðàììíîé ñèñòåìû îò íà÷àëà (àíàëèç òðåáîâàíèé) äî êîíöà (òåñòèðîâàíèå è ñîïðîâîæäåíèå-äîêóìåíòàöèÿ).

Â õîäå âûïîëíåíèÿ êóðñîâîé ðàáîòû ïîëó÷èòü ïðîôåññèîíàëüíûå íàâûêè â ïîñòàíîâêå çàäà÷è, àíàëèçå òðåáîâàíèé, âûáîðå ïðåäñòàâëåíèÿ èñõîäíûõ äàííûõ è ðåçóëüòàòà, ðàçðàáîòêå ñïåöèôèêàöèé, ïðîåêòèðîâàíèÿ ïðîãðàììíîé ñèñòåìû, íàïèñàíèè ïðîãðàììû íà âûáðàííîì ÿçûêå ïðîãðàììèðîâàíèÿ ñ èñïîëüçîâàíèåì îáúåêòíî-îðèåíòèðîâàííîé òåõíîëîãèè è áèáëèîòåê êëàññîâ, òåñòèðîâàíèè è îòëàäêå ïðîãðàììû, îôîðìëåíèè äîêóìåíòàöèè.

**Îñíîâíûå òðåáîâàíèÿ ê ïðîãðàììå**

Êóðñîâàÿ ðàáîòà ïî êóðñó "Îáúåêòíî-îðèåíòèðîâàííîå ïðîãðàììèðîâàíèå» âûïîëíÿåòñÿ èíäèâèäóàëüíî êàæäûì ñòóäåíòîì â ñîîòâåòñòâèè ñ âûäàííûì ïðåïîäàâàòåëåì âàðèàíòîì. Îáÿçàòåëüíûì ÿâëÿåòñÿ èñïîëüçîâàíèå â êóðñîâîé ðàáîòå îáúåêòíî-îðèåíòèðîâàííîãî ïîäõîäà è ïîëüçîâàòåëüñêèõ êëàññîâ.

Â ïðîãðàììå äîëæåí áûòü ðåàëèçîâàí êëàññ ïðÿìîóãîëüíèêîâ.

Ïðîãðàììà äîëæíà ðàáîòàòü ïîä óïðàâëåíèåì îïåðàöèîííîé ñèñòåìîé «Windows» è áûòü ðàçðàáîòàíà íà ÿçûêå ïðîãðàììèðîâàíèÿ «Delphi».

Â êëàññå äîëæíû áûòü ïðåäóñìîòðåííû âîçìîæíîñòè:

1.         Ïåðåìåùåíèÿ ïðÿìîóãîëüíèêîâ;

2.         Èçìåíåíèÿ ðàçìåðîâ ïðÿìîóãîëüíèêîâ;

3.         Ïîñòðîåíèå íàèìåíüøåãî ïðÿìîóãîëüíèêà, ñîäåðæàùåãî äâà çàäàííûõ ïðÿìîóãîëüíàêà;

4.         Ïîñòðîåíèå ïðÿìîóãîëüíèêà ÿâëÿþùåãîñÿ îáùåé ÷àñòüþ (ïåðåñå÷åíèåì) äâóõ çàäàííûõ ïðÿìîóãîëüíèêîâ;

**Îïèñàíèå êëàññîâ ïðîãðàììû**

Ïðîãðàììà ñîäåðæèò â ñåáå äâà îñíîâíûõ êëàññà: stack è TForm1. stack ÿâëÿåòñÿ êëàññîì ñòåêà ïðÿìîóãîëüíèêîâ. Âñå ïðÿìîóãîëüíèêè õðàíÿòñÿ â í¸ì è âûçûâàþòñÿ ïî ïñåâäîíèìàì. TForm1 – êëàññ ãëàâíîãî îêíà ïðîãðàììû.

Êëàññ stack:

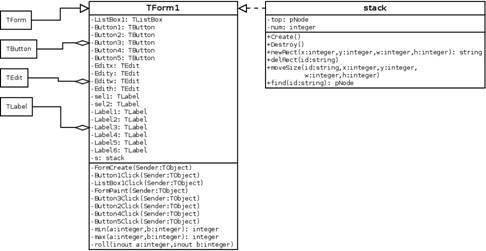
Õðàíèò è îáðàáàòûâàåò îáúåêòû ïðÿìîóãîëüíèêîâ.

|  |  |
| --- | --- |
| top : pNode; | Âåðøèíà ñòåêà. |
| num : integer; | Ñ÷¸ò÷èê äëÿ çàäàíèÿ óíèêàëüíûõ ïñåâäîíèìîâ. |
| constructor Create; | Êîíñòðóêòîð |
| destructor Destroy; | Äåñòðóêòîð |
| function newRect(x, y, w, h : integer) : string; | Ôóíêöèÿ ñîçäàíèÿ íîâîãî ïðÿìîóãîëüíèêà. |
| procedure delRect(id : string); | Ôóíêöèÿ óäàëåíèÿ ïðÿìîóãîëüíèêà |
| procedure moveSize(id : string; x, y, w, h : integer); | Ïðîöåäóðà äëÿ èçìåíåíèÿ ðàçìåðà è ïîëîæåíèÿ ïðÿìîóãîëüíèêà |
| function find(id : string) : pNode; | Ôóíêöèÿ ïîèñêà îáúåêòà ïðÿìîóãîëüíèêà ïî ïñåâäîíèìó. |

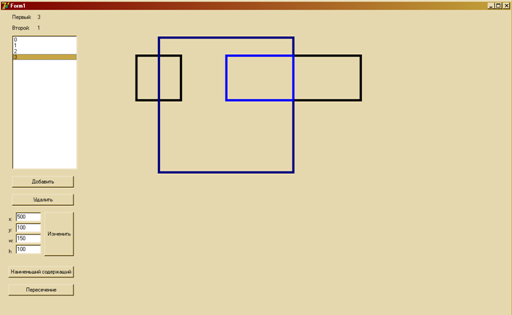
Êëàññ TForm1:

|  |  |
| --- | --- |
| ListBox1: TListBox | Ñïèñîê ïðÿìîóãîëüíèêîâ |
| Button1: TButton; | Êíîïêà ñîçäàíèÿ íîâîãî ïðÿìîóãîëüíèêà |
| sel2: TLabel; | Ïîëå ñ ïðåäûäóùèì âûáðàííûì îáúåêòîì |
| sel1: TLabel; | Ïîëå ñ òåêóùåì âûáðàííûì îáúåêòîì |
| Button2: TButton; | Êíîïêà äëÿ óäàëåíèÿ ïðÿìîóãîëüíèêà |
| Editx: TEdit; | Ïîëå äëÿ ââîäà êîîðäèíàòû x |
| Edity: TEdit; | Ïîëå äëÿ ââîäà êîîðäèíàòû y |
| Editw: TEdit; | Ïîëå äëÿ ââîäà øèðèíû |
| Edith: TEdit; | Ïîëå äëÿ ââîäà âûñîòû |
| Button3: TButton; | Êíîïêà äëÿ èçìåíåíèÿ ñîñòîÿíèÿ ïðÿìîóãîëüíèêà |
| Button4: TButton; | Êíîïêà äëÿ ïîñòðîåíèÿ íàèìåíüøåãî ïðÿìîóãîëüíèêà ñîäåðæàùåãî â ñåáå çàäàííûå. |
| Button5: TButton; | Êíîïêà äëÿ ïîñòðîåíèÿ ïðÿìîóãîëüíèêà íà ïåðåñå÷åíèè. |
| procedure FormCreate(Sender: TObject); | Ïðîöåäóðà ñîçäàíèÿ îêíà |
| procedure Button1Click(Sender: TObject); | Êíîïêà ñîçäàíèÿ ïðÿìîóãîëüíèêà |
| procedure ListBox1Click(Sender: TObject); | Ïðîöåäóðà âûäåëåíèÿ ïðÿìîóãîëüíèêà |
| procedure FormPaint(Sender: TObject); | Ïðîöåäóðà ðèñîâàíèÿ |
| procedure Button3Click(Sender: TObject); | Ïðîöåäóðà èçìåíåíèÿ ñîñòîÿíèÿ ïðÿìîóãîëüíèêà |
| procedure Button2Click(Sender: TObject); | Ïðîöåäóðà óäàëåíèÿ ïðÿìîóãîëüíèêà |
| procedure Button4Click(Sender: TObject); | Ïðîöåäóðà ïîñòðîåíèÿ íàèìåíüøåãî ïðÿìîóãîëüíèêà ñîäåðæàùåãî çàäàííûå |
| function min (a, b : integer) : integer; | Âñïîìîãàòåëüíàÿ ôóíêöèÿ îïðåäåëÿþùàÿ íàèìåíüøèé ýëåìåíò. |
| function max (a, b : integer) : integer; | Âñïîìîãàòåëüíàÿ ôóíêöèÿ îïðåäåëÿþùàÿ íàèáîëüøèé ýëåìåíò. |
| procedure roll(var a, b : integer); | Âñïîìîãàòåëüíàÿ ôóíêöèÿ ìåíÿþùàÿ çíà÷åíèÿ îïåðàíäîâ ìåñòàìè. |
| procedure Button5Click(Sender: TObject); | Ïðîöåäóðà ïîñòðîåíèÿ ïðÿìîóãîëüíèêà íà ïåðåñå÷åíèè äâóõ çàäàííûõ |
| s : stack; | Ñòåê ïðÿìîóãîëüíèêîâ. |

**Äèàãðàììà êëàññîâ ïðîãðàììû**



**Ðóêîâîäñòâî ïîëüçîâàòåëÿ**



Äëÿ äîáàâëåíèÿ ïðÿìîóãîëüíèêà íàäî íàæàòü êíîïêó "Äîáàâèòü". Â ðåçóëüòàòå ïîÿâèòñÿ ïðÿìîóãîëüíèê ñî ñòàíäàðòíûì ðàñïîëîæåíèåì. Ïîìåíÿòü åãî âû ìîæåòå âûáðàâ ïðÿìîóãîëüíèê â ñïèñêå è ââåäÿ êîîðäèíàòû â ñîîòâåòñòâóþùèå ïîëÿ, à ïîòîì íàæàâ "Èçìåíèòü". Ñèíèì öâåòîì âûäåëÿþòñÿ âûáðàííûå ïðÿìîóãîëüíèêè: ÿðêî ñèíèì – òåêóùèé, ò¸ìíî ñèíèì – ïðåäûäóùèé.

Ïðè íàæàòèè êíîïêè "Íàèìåíüøèé ñîäåðæàùèé" ñîçäàñòñÿ ìèíèìàëüíûé ïðÿìîóãîëüíèê ñîäåðæàùèé â ñåáå âûáðàííûé è ïðåäûäóùèé âûáðàííûé ïðÿìîóãîëüíèêè. Ïî òàêîìó æå ïðèíöèïó ðàáîòàåò êíîïêà "ïåðåñå÷åíèå", òîëüêî îíà ñîçäà¸ò ïðÿìîóãîëüíèê ñîäåðæàùèé ïåðåñå÷åíèå âûáðàííûõ.

**Çàêëþ÷åíèå**

Öåëü ðàáîòû, ðàçðàáîòêà êëàññà ïðÿìîóãîëüíèêîâ ñî ñòîðîíàìè ïàðàëåëüíûìè îñÿì êîîðäèíàò, óñïåøíî âûïîëíåíà. Ïðîãðàììà âûïîëíåíà â ñðåäå ðàçðàáîòêè Delphi è ïðåäíàçíà÷àåòñÿ äëÿ ðàáîòû ïîä óïðàâëåíèåì îïåðàöèîííîé ñèñòåìû Windows. Ïðîãðàììà ïîçâîëÿåò âûïîëíÿòü ïåðåìåùåíèå, äîáàâëåíèå, óäàëåíèå è èçìåíåíèå ðàçìåðà ïðÿìîóãîëüíèêîâ, à òàê æå ïîçâîëÿåò ïîñòðîèòü ìèíèìàëüíûé ïðÿìîóãîëüíèê âìåùàþùåãî â ñåáÿ äâà çàäàííûõ, è ïîñòðîåíèå ïðÿìîóãîëüíèêà ÿâëÿþùåãîñÿ îáùåé ÷àñòüþ äâóõ âûáðàííûõ.

**Ñïèñîê ëèòåðàòóðû**

1.    Êëèìîâà Ë.Ì. Delphi 7. Îñíîâû ïðîãðàììèðîâàíèÿ. Ðåøåíèå òèïîâûõ çàäà÷. Ñàìîó÷èòåëü. Èçäàíèå òðåòüå.- Ì.: ÊÓÄÈÖ- ÎÁÐÀÇ, 2006.- 480 ñ.

2.    Õîìîíåíêî À.Ä. è äð. Delphi 7/ Ïîä îáù. Ðåä. À.Ä. Õîìîíåíêî.- ÑÏá: ÁÕÂ- Ïåòåðáóðã, 2005.- 1216 ñ.

3.    Áîáðîâñêèé Ñ.È. Delphi 7. Ó÷åáíûé êóðñ- ÑÏá: Ïèòåð, 2005.- 736 ñ.

4.    http://khpi-iip.mipk.kharkiv.edu/library/case/leon/gl3/gl3.html

5.    http://khpi-iip.mipk.kharkiv.edu/library/case/leon/gl5/gl5.html

6.    Ïàâëîâñêàÿ Ò.À. Ïðîãðàììèðîâàíèå íà ÿçûêå âûñîêîãî óðîâíÿ – ÑÏá: Ïèòåð, 2006.

7.    Ïîãàí À.Ì. Ðóêîâîäñòâî ïðîãðàììèñòà – «Ýñêèìî», 2006.

8.    Ëåñíåâñêèé À.Ñ. Îáúåêòíî-îðèåíòèðîâàííîå ïðîãðàììèðîâàíèå

9.    Èâàíîâà Ã.Ñ. Îáúåêòíî-îðèåíòèðîâàííîå ïðîãðàììèðîâàíèå – Ì, 2003.

**Ïðèëîæåíèå À**

êëàññ ïðÿìîóãîëüíèê ïðîãðàììà èíòåðôåéñ

Ëèñòèíã ïðîãðàììû

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Unit2;

type

TForm1 = class(TForm)

ListBox1: TListBox;

Button1: TButton;

Label1: TLabel;

Label2: TLabel;

sel2: TLabel;

sel1: TLabel;

Button2: TButton;

Editx: TEdit;

Edity: TEdit;

Editw: TEdit;

Edith: TEdit;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Button3: TButton;

Button4: TButton;

Button5: TButton;

procedure FormCreate(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure ListBox1Click(Sender: TObject);

procedure FormPaint(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

function min (a, b : integer) : integer;

function max (a, b : integer) : integer;

procedure roll(var a, b : integer);

procedure Button5Click(Sender: TObject);

private

{ Private declarations }

s : stack;

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

s := stack.Create;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

ListBox1.Items.Add(s.newRect(300, 100, 100, 100));

Refresh;

end;

procedure TForm1.ListBox1Click(Sender: TObject);

var

selItem : string;

f : pNode;

begin

selItem := ListBox1.Items[ListBox1.ItemIndex];

if selItem <> sel1.Caption then

begin

sel2.Caption := sel1.Caption;

sel1.Caption := selItem;

f := s.find(selItem);

if f <> nil then

begin

EditX.Text := IntToStr(f.x);

EditY.Text := IntToStr(f.y);

EditW.Text := IntToStr(f.w);

EditH.Text := IntToStr(f.h);

end;

end;

Refresh;

end;

procedure TForm1.FormPaint(Sender: TObject);

var

i : integer;

p : pNode;

begin

for i := 0 to ListBox1.Items.Count - 1 do

begin

p := s.find(ListBox1.Items[i]);

if p <> nil then

begin

if p.id = sel1.Caption then Canvas.Pen.Color := clBlue

else if p.id = sel2.Caption then Canvas.Pen.Color := clNavy

else Canvas.Pen.Color := clBlack;

Canvas.Pen.Width := 5;

Canvas.Brush.Style := bsClear;

Canvas.Rectangle(p.x, p.y, p.x + p.w, p.y + p.h);

end;

end;

end;

procedure TForm1.Button3Click(Sender: TObject);

var

f : pNode;

begin

f := s.find(sel1.Caption);

if f <> nil then

begin

f.x := StrToInt(Editx.Text);

f.y := StrToInt(Edity.Text);

f.w := StrToInt(Editw.Text);

f.h := StrToInt(Edith.Text);

Refresh;

end;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

if sel1.Caption <> '' then s.delRect(sel1.Caption);

ListBox1.Items.Delete(ListBox1.ItemIndex);

Refresh;

end;

procedure TForm1.Button4Click(Sender: TObject);

var

f1, f2 : pNode;

x, y, w, h : integer;

begin

f1 := s.find(sel1.Caption);

f2 := s.find(sel2.Caption);

x := min(min(f1.x, f1.x + f1.w), min(f2.x, f2.x + f2.w));

w := max(max(f1.x, f1.x + f1.w), max(f2.x, f2.x + f2.w));

w := w - x;

y := min(min(f1.y, f1.y + f1.h), min(f2.y, f2.y + f2.h));

h := max(max(f1.y, f1.y + f1.h), max(f2.y, f2.y + f2.h));

h := h - y;

ListBox1.Items.Add(s.newRect(x, y, w, h));

Refresh;

end;

function TForm1.min(a, b : integer) : integer;

begin

if a < b then min := a else min := b;

end;

function TForm1.max(a, b : integer) : integer;

begin

if a > b then max := a else max := b;

end;

procedure TForm1.roll(var a, b : integer);

var

c : integer;

begin

c := a;

a := b;

b := c;

end;

procedure TForm1.Button5Click(Sender: TObject);

var

x1, y1, w1, h1 : integer;

x2, y2, w2, h2 : integer;

x, y, w, h : integer;

f : pNode;

begin

f := s.find(sel1.Caption);

x1 := f.x;

y1 := f.y;

w1 := x1 + f.w;

if w1 < x1 then roll(w1, x1);

h1 := y1 + f.h;

if h1 < y1 then roll(h1, y1);

f := s.find(sel2.Caption);

x2 := f.x;

y2 := f.y;

w2 := x2 + f.w;

if w2 < x2 then roll(w2, x2);

h2 := y2 + f.h;

if h2 < y2 then roll(h2, y2);

if (((x1 > x2) and (x1 < w2)) or ((w1 > x2) and (w1 < w2)) or

((x2 > x1) and (x2 < w1)) or ((w2 > x1) and (w2 < w1))) and

(((y1 > y2) and (y1 < h2)) or ((h1 > y2) and (h1 < h2)) or

((y2 > y1) and (y2 < h1)) or ((h2 > y1) and (h2 < h1))) then

begin

x := max(x1, x2);

y := max(y1, y2);

w := min(w1, w2);

h := min(h1, h2);

w := w - x;

h := h - y;

ListBox1.Items.Add(s.newRect(x, y, w, h));

Refresh;

end;

end;

end.

unit Unit2;

interface

uses sysutils, classes;

type

pNode = ^Node;

Node = record

id : string;

x, y, w, h : integer;

next : pNode;

end;

stack = class

top : pNode;

num : integer;

public

constructor Create;

destructor Destroy;

function newRect(x, y, w, h : integer) : string;

procedure delRect(id : string);

procedure moveSize(id : string; x, y, w, h : integer);

function find(id : string) : pNode;

end;

implementation

constructor stack.Create;

begin

top := nil;

num := 0;

end;

destructor stack.Destroy;

var

del : pNode;

begin

while top <> nil do

begin

del := top;

top := top.next;

Dispose(del);

 end;

end;

function stack.newRect(x, y, w, h : integer) : string;

var

n : pNode;

begin

n := New(pNode);

n.id := IntToStr(num);

Inc(num);

n.x := x;

n.y := y;

n.w := w;

n.h := h;

n.next := top;

top := n;

newRect := n.id;

end;

procedure stack.delRect(id : string);

var

f, d : pNode;

begin

f := top;

if f <> nil then

if f.id = id then

begin

top := top.next;

Dispose(f);

end

else

begin

while f.next <> nil do

begin

if f.next.id = id then

begin

d := f.next;

f.next := d.next;

Dispose(d);

break;

end

end;

end;

end;

procedure stack.moveSize(id : string; x, y, w, h : integer);

var

f : pNode;

begin

f := find(id);

if f <> nil then

begin

f.x := x;

f.y := y;

f.w := w;

f.h := h;

end;

end;

function stack.find(id : string) : pNode;

var

f : pNode;

begin

f := top;

while f <> nil do

begin

if f.id = id then break;

f := f.next;

end;

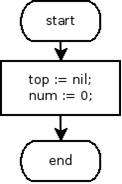
find := f;

end;

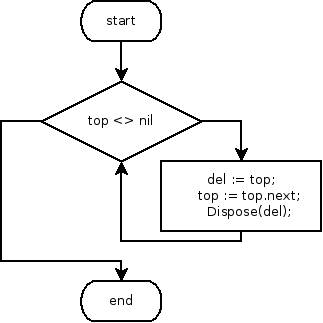
end.

Áëîê-ñõåìû ðàçðàáîòàííûõ ìåòîäîâ

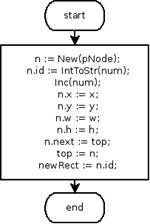
constructor stack.Create;



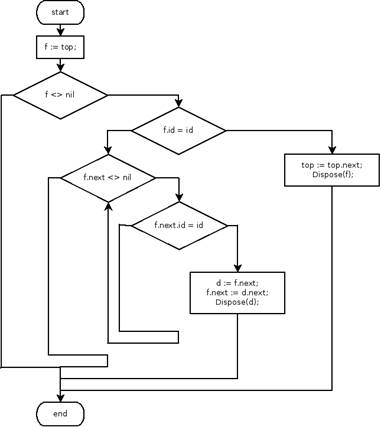
destructor stack.Destroy;



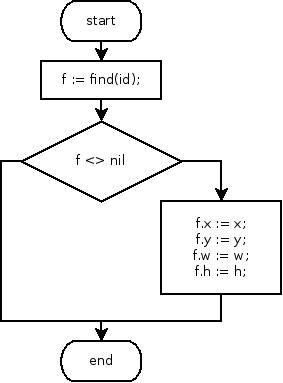
function stack.newRect(x, y, w, h : integer) : string;



procedure stack.delRect(id : string);



procedure stack.moveSize(id : string; x, y, w, h : integer);



function stack.find(id : string) : pNode;

