Opis Unit-a CSMPBlok.pas

U Unit-u CSMPBlok.pas se nalazi kod za klasu TCSMPBlok. CSMPBlok predstavlja grafičku reprezentaciju funkcija koje se koriste tokom simulacije. Izgled klase (njenih metoda i polja) je sledeći:

```
TCSMPBlok = class(TCustomControl)
  private
    FAktivno: boolean;
    FOnAktiviran: TNotifyEvent;
    FSmer: TSmer;
    FOnPromenaSmera: TNotifyEvent;
    FSlikeLevo: TImageList;
    FSlikeDesno: TImageList;
    FSlikeDole: TImageList;
    FSlikeGore: TImageList;
    FAktivneSlike: TImageList;
    FImageIndex:integer;
    FLokalniBroj:integer;
    FNaziv:string;
    FOnDodajVezu: TNotifyEvent;
    FMaxUlaznihVeza: integer;
    FOnBlokMouseMove: TNotifyEvent;
    FOnKreirajJunction: TNotifyEvent;
    FOnBlokClick: TNotifyEvent;

    procedure SetImageIndex(const Value: integer);
    procedure SetAktivno(const Value: boolean);
    procedure SetNaziv(const Value: string);
    procedure SetOnAktiviran(const Value: TNotifyEvent);
    procedure SetSmer(const Value: TSmer);
    procedure SetOnPromenaSmera(const Value: TNotifyEvent);
    procedure SetSlikeDesno(const Value: TImageList);
    procedure SetSlikeDole(const Value: TImageList);
    procedure SetSlikeGore(const Value: TImageList);
    procedure SetSlikeLevo(const Value: TImageList);
    procedure SetAktivneSlike(const Value: TImageList);
    function GetNaziv: string;
    procedure SetLokalniBroj(const Value: integer);
    function GetImageIndex: integer;
    function GetlokalniBroj: integer;
    procedure SetOnDodajVezu(const Value: TNotifyEvent);
    function GetIzlaznaTacka: TPoint;
    function GetUlaznaTacka: TPoint;
```

```
    procedure SetMaxUlaznihVeza(const Value: integer);
    procedure SetOnBlokClick(const Value: TNotifyEvent);
    procedure SetOnBlokMouseMove(const Value: TNotifyEvent);
    procedure SetOnKreirajJunction(const Value: TNotifyEvent);
    { Private declarations }
  protected
    { Protected declarations }
    nX,nY:integer;
    MDown:boolean;

    procedure ObrisiUVezu(int:integer);
    procedure ObrisiIVezu(int:integer);
    procedure OdradiMove;
    procedure WMWindowPosChanged(var Message: TWMWindowPosChanged);
message WM_WINDOWPOSCHANGED;
  public
    { Public declarations }

    UlazneVeze:TList;
    IzlazneVeze:TList;
    ParI, ParII, ParIII: real;
    Nazivi: array[0..5] of string;
    Stanje:TStanje;

    BrojPar: byte; // Broj parametara
    IzlaznaVrednost: real;
    Izlaz: TCSMPBlok;
    Sortiran: boolean;
    Sifra: byte;
    BrojUlaza: byte;
    function  GdeUBloku(Veza:TVeza):integer;

    procedure Save(Koren:IXMLNode);
    procedure Load(Koren:IXMLNode);

    function NadjiKrajBlok(uVeza:TVeza):TCSMPBlok;
    constructor Create(AOwner: TComponent); override;
    destructor  Destroy;override;

    function DodajUVezu(sender: TVeza; Gde: integer): string;
    function DodajIVezu(sender: TVeza): string;

    procedure BrisiVezuU(sender:TObject);
    procedure BrisiVezuI(sender:TObject);

    procedure Paint;override;
```

```
    procedure MouseDown(Button: TMouseButton; Shift: TShiftState;
      X, Y: Integer); override;
    procedure MouseMove(Shift: TShiftState; X, Y: Integer);override;
    procedure MouseUp(Button: TMouseButton; Shift: TShiftState;
      X, Y: Integer); override;
    procedure RotirajCC;
    procedure RotirajC;
    function VratiUlaz(i: integer): TCSMPBlok;
    property Smer:TSmer read FSmer write SetSmer;
  published
    { Published declarations }
    property LokalniBroj:integer read GetlokalniBroj write SetLokalniBroj;
    property SlikeLevo:TImageList read FSlikeLevo write SetSlikeLevo;
    property SlikeDesno:TImageList read FSlikeDesno write SetSlikeDesno;
    property SlikeGore:TImageList read FSlikeGore write SetSlikeGore;
    property SlikeDole:TImageList read FSlikeDole write SetSlikeDole;
    property ImageIndex:integer read GetImageIndex write SetImageIndex;
    property Aktivno:boolean read FAktivno write SetAktivno;
    property Naziv:string read GetNaziv write SetNaziv;
    property AktivneSlike:TImageList read FAktivneSlike write SetAktivneSlike;


    property UlaznaTacka:TPoint read GetUlaznaTacka;
    property IzlaznaTacka:TPoint read GetIzlaznaTacka;

    property MaxUlaznihVeza:integer read FMaxUlaznihVeza write SetMaxUlaznihVeza;

    property OnAktiviran:TNotifyEvent read FOnAktiviran write SetOnAktiviran;
    property OnPromenaSmera:TNotifyEvent read FOnPromenaSmera write
SetOnPromenaSmera;
    property OnDodajVezu:TNotifyEvent read FOnDodajVezu write SetOnDodajVezu;

    property OnKreirajJunction:TNotifyEvent read FOnKreirajJunction write
SetOnKreirajJunction;
    property OnBlokClick:TNotifyEvent read FOnBlokClick write SetOnBlokClick;
    property OnBlokMouseMove:TNotifyEvent read FOnBlokMouseMove write
SetOnBlokMouseMove;

  end;
```

Kao što se vidi klasa TCSMPBlok nasleđuje klasu TCustomControl. Na ovaj način smo izbegli da ponovo implementiramo grafičke mogućnosti nasleđene klase. Opis same klase TcustomControl dat je u help-u Delphi-ja 7.
Usresredimo se na samu klasu TCSMPBlok. TCSMPBlok predstavlja grafičku interpretaciju neke matematičke f-je. Karakteriše ga LokalniBroj u levom uglu i grafička reprezentacija u vidu ikonice u desnom delu bloka.

Published sekcija

property Smer:TSmer;

Smer je property koji nam govori na koju stranu je dati CSMPBlok okrenut u datom trenutku. Moguće vrednosti su smLevo=1, smDole=2, smDesno=3, smGore=4. Vrednost ovog property-ja ne treba menjati ručno već preko metoda RotirajCC i RotirajC.

property LokalniBroj:integer;

Lokalni broj služi kao jedinstveni identifikator instance objekta TCSMPBlok. Na osnovu ovog broja mozemo lako da pristupimo svakom bloku bilo za potrebe učitavanja, upisivanja ili dodavanja nove veze. LokalniBroj se prikazuje u gornjem levom uglu TCSMPBloka. LokalniBroj se ažurira prilikom svakog brisanja ili dodavanja nekog od blokova (procedurom OdrediBrojeve() klase TCSMPRadnaPovrsina). Ovo se radi da bi se odrzala veza između indeksa CSMPBloka u listi blokova (unutar TCSMPRadnaPovrsina) i vrednosti promenljive LokalniBroj. Takođe, LokalniBroj se koristi i tokom simulacije u unit-u Obrada.

property SlikeLevo:TimageList;

property SlikeDesno:TImageList;

property SlikeGore:TImageList;

property SlikeDole:TImageList;

Prethodna četiri property-ja predstavljaju reference na instancu klase TimageList koja u sebi sadrži ikonice blokova u položaju Levo, Desno, Gore i Dole. Ovi objektni se ne instanciraju za svaki blok već predstavljaju deljeni resurs. Napomena: Inicijalizacija ovih referenci mora da se izvši prilikom kreiranja CSMPBloka inače može doći do neželjenih efekata prilikom iscrtavanja.

property ImageIndex:integer;

ImageIndex čuva vrednost indeksa ikone datog bloka koja se nalazi u SlikeLevo, SlikeDesno, SlikeGore, SlikeDole. Takodje, ImageIndex služi za utvrđivanje tipa bloka. Samu vrednost ImageIndex dobija prilikom prevlačenja ikonice iz TCSMPBar-a na TCSMPRadnaPovrsina. ImageIndex se cuva pod nazivom Sifra u Manifest.xml fajlu I prilikom cuvanja samog bloka.

property Aktivno:boolean;

Ovo svojstvo TCSMPBlok-a nam govori da li je dati CSMPBlok selektovan ili ne. Samo jedan blok može biti selektovan u jednom trenutku. Vrednost ovog svojstva se održava od strane TCSMPRadnaPovrsina i to na sledeći način:
Prilikom kreiranja CSMPBloka od strane CSMPRadnePovrsine vrši se povezivanje metode TCSMPRadnaPovrsina.Aktivacija(Sender:TObject) sa dogadjajem OnAktiviran:TnotifyEvent klase CSMPBlok. Na ovaj način kada god dođe do aktiviranja datog bloka poziva se metoda Aktivacija koja prolazi kroz sve blokove na CSMPRadnaPovrsina i deaktivira sve ostale blokove sem selektovanog.

property Naziv:string;

U nazivu se čuva naziv tipa bloka koji sluzi kao tekstualna informacija za korisnika.

property AktivneSlike:TImageList;

AktivneSlike predstavljaju referencu na jednu od četiri TimageList-a (Gore, Dole, Levo, Desno) i ažurira se prilikom svake promene smera. Ova referenca se ne instancira vec referencira neki od četiri postojeća objekta.

property UlaznaTacka:TPoint;

property IzlaznaTacka:TPoint;

UlaznaTacka i IzlaznaTacka predstavljaju poziciju na kojoj se ulazne i izlazna veza vezuju za dati blok. Date tacke zavise od smera CSMPBloka kao i njegove pozicije na ekranu. Tacke su date relativno u odnosu na TCSMPRadnaPovrsina na kojoj se nalazi dati CSMPBlok.

property MaxUlaznihVeza:integer;

MaxUlaznih veza je property koji čuva vrednost koja se odnosi na broj ulaznih veza koje CSMPBlok datog tipa može da ima. Ovaj property može uzeti vrednost iz intervala 0 do 3. Ovo ograničenje je nasleđeno iz ranije verzije programa ali je i sasvim korektno jer se u radu sa prethodnom aplikacijom došlo do zaključka da se više od 3 ulaza vrlo retko (ako ne i nikad) koristi. U slučaju da se jednog dana javi potreba da se izvrši proširenje kapaciteta programa to se lako može učiniti manjim izmenama u kodu i promenom vrednosti u Manifest.xml fajlu.

property OnAktiviran:TNotifyEvent;

Događaj OnAktiviran se poziva onog trenutka kada se vrednost property-ja Aktivno promeni u true. Ovo je vrlo značajan događaj koji obavestava TCSMPRadnaPovrsina i TCSMPInspektor da je neki CSMPBlok dobio fokus. Takodje, posle poziva ovog dogadjaja svi ostali blokovi prestaju da budu aktivni (za to se brine CSMPRadnaPovrsina).

property OnPromenaSmera:TNotifyEvent;

Dogadjaj OnPromena smera se aktivira onog trenutka kada se Smer CSMPBlok-a promeni. Trenutno nema upotrebnu vrednost, ali je pridodato radi postovanja principa enkapsulacije.

property OnDodajVezu:TNotifyEvent;

Ne koristi se, tu je radi enkapsulacije.

property OnKreirajJunction:TNotifyEvent;

Ne koristi se, tu je radi enkapsulacije.

property OnBlokClick:TNotifyEvent;

OnBlokClick je dogadjaj koji se poziva kada se klikne na CSMPBlok ali samo kada je blok u stanju stDodajemVezu ili stKreiramJunction. Ovo je bitno jer kada se veza dodaje dinamički prvo sve blokove postavimo u neko od ova dva stanja (u zavisnosti od toga šta

želimo da dodamo), te kada se klikne na blok znamo koji će nam blok biti krajnji tako sto ispitamo Sender samog događaja.

property OnBlokMouseMove:TNotifyEvent;

OnBlokMouseMove je događaj koji se poziva kada miš pređe preko bloka. U slučaju da dodajemo novu vezu ili junction prilikom prelaska miša preko bloka veza se sama pozicionira na mesto gde će biti prikačena.

Protected sekcija

procedure WMWindowPosChanged(var Message: TWMWindowPosChanged); message WM_WINDOWPOSCHANGED;

Ovo je procedura koja se poziva kada se od sistema primi poruka da se blok pomerio. Ona služi da se ponovo podese početne i krajnje tačke veza koje su prikačene za dati blok (kako ulaznih tako i izlaznih). Da bi ovo postigla poziva proceduru OdradiMove.

nX,nY:integer;

nX i nY predstavljaju pomeraj bloka u relativnim kordinatama u odnosu na blok. Naime, pomeranje bloka se vrši tako što se klikne na blok i postavite promenljivu MDown na true i pri svakom dogadjaju MouseMove pratimo za koliko se miš pomerio u odnosu na prethodni položaj i ažuriramo nX i nY.

MDown:boolean;

Prilikom svakog događaja MouseDown promenljiva MDown se postavlja na true a prilikom svakog MouseUp na false. Na ovaj način možemo da pratimo da li prelazimo preko bloka dok drzimo taster na mišu.

procedure OdradiMove;

Ovo je procedura koja podešava ulazne i izlazne tacke ulaznih i izlaznih veza.

procedure ObrisiUVezu(int:integer);

procedure ObrisiIVezu(int:integer);

ObrisiUVezu briše vezu sa zadatim indeksom iz liste ulaznih veza. ObrisiIVezu radi to isto samo sa izlaznim vezama.

Public sekcija

UlazneVeze:TList;

Čuva listu ulaznih veza povezanih sa datim blokom.

IzlazneVeze:TList;

Čuva listu izlaznih veza (tipa junction) povezanih sa datim blokom.

Parl, ParII, ParIII: real;

Promenljive u kojima se čuvaju vrednosti parametara vezane za dati tip CSMPBloka.

Nazivi: array[0..5] of string;

Nazivi predstavljaju imena parametara koja bliže objašnjavaju svrhu samog bloka. Nazivi se takođe koriste prilikom čuvanja i učitavanja .csmp fajla da bi se imenovao IXMLNode;

Stanje:TStanje;

Stanje nam govori šta se trenutno dešava sa blokom. Stanje može uzeti vrednost iz sledećeg skupa stSelektujem=1 ,stDodajemVezu=2,stKreiramJunction=3. U zavisnoti od stanja u kome se blok nalazi različite akcije se preduzimaju prilikom pomeranja miša i spuštanja miša na blok.

BrojPar: byte;

BrojPar predstavlja promenljivu u kojoj se čuva vrednost koja nam govori koliko je moguće uneti parametara za dati blok. Na ovaj način se lako može zabraniti unošenje više parametara nego što je potrebno.

IzlaznaVrednost: real;

Izlazna vrednost je pomoćna promenljiva koja se koristi prilikom same simulacije i predstavlja vrednost funkcije koju dati blok reprezentuje.

Izlaz: TCSMPBlok;

Izlaz predstavlja referencu koja pokazuje na blok koji se nalazi na drugom kraju izlazne veze. Ova referenca se ažurira prilikom svakog dodavanja ili brisanja izlazne veze datom bloku.

Sortiran: boolean;

Promenljiva koja se koristi prilikom sortiranja pre nego što se počne sama simulacija i ona ukazuje da li je trenutni blok ubačen u listu sortiranih blokova ili nije. Ovo je potrebno uvesti iz razloga što je sam grafik u suštini graf koji može imati i povratne veze te bi smo onda mogli ući u beskonačnu petlju.

Sifra: byte;

Sifra predstavlja, kako jedinstveni identifikator datog bloka, tako i broj ikone u TImgList-u AktivneSlike. Vrednost dobija onog trenutka kada se TSkrolStavka prevuče na TCSMPRadnaPovrsina.

BrojUlaza: byte;

BrojUlaza predstavlja promenljivu koja nam govori koliko veza postoji na ulazu u dati blok. Ovaj broj se ažutrira prilikom svakog dodavanja ili brisanja veze.

function GdeUBloku(Veza:TVeza):integer;

GdeUBloku predstavlja pomoćnu funkciju koja se koristi prilikom čuvanja bloka u fajl. Naime, pošto su veze predstavljene pointerima teško je saznati na kom ulazu po redu se nalazi neka veza. GdeUBloku vraća baš to, za vezu koja je prenesena kao parametar vraća na kom se ulazu nalazi. U slučaju da se Veza ne nalazi ni na jednom od ulaza vraća se -1 kao šifra grečke.

procedure Save(Koren:IXMLNode);

Procedura Save čuva vrednosti koje jedinstveno identifikuju dati blok kao i stanje u kome se on nalazi u xml obliku i to kao dete (ChildNode) korenog XMLNode-a Koren. Vrednosti koje se čuvaju ovom procedurom su: Top, Left, Sifra, Naziv, LokalniBroj, BrojPar, MaxUlaznihVeza, Nazivi[i], ParI, ParII, ParIII, Smer.

procedure Load(Koren:IXMLNode);

Procedura Load ucitava vrednosti koje jedinstveno identifikuju dati blok kao i stanje u kome se on nalazi u xml obliku i to kao dete (ChildNode) korenog XMLNode-a Koren. Vrednosti koje se ičitavaju ovom procedurom su: Top, Left, Sifra, Naziv, LokalniBroj, BrojPar, MaxUlaznihVeza, Nazivi[i], ParI, ParII, ParIII, Smer.

function NadjiKrajBlok(uVeza:TVeza):TCSMPBlok;

NadjiKrajBlok je još jedna od pomoćnih funkcija koja služi za pronalaženje bloka koji se nalazi na drugom kraju veze (početnom kraju). Ovo ne bi bio problem kada bi sve veze bile bez junction-a. Iz ovog razloga potrebno je rekurzivno proći kroz sve junction veze dok se ne dođe do one veze koja nije junction i onda vratimo njen početni blok.

constructor Create(AOwner: TComponent); override;

Konstruktor služi za inicijalizaciju svih promnljivih i postavljanje početnog stanja bloka na koji se odnosi.

destructor Destroy;override;

Destruktor se brine da sva alocirana memorije bude oslobođena. U našem slučaju to su samo liste ulaznih i izlaznih veza. UlazneVeze.Free; IzlazneVeze.Free;

function DodajUVezu(sender: TVeza; Gde: integer): string;

Funkcija DodajUVezu dodaje vezu u listu ulaznih veza i to na poziciju koja je definisana promenljivom Gde.

function DodajIVezu(sender: TVeza): string;

Funkcija DodajIVezu dodaje vezu (sender) u listu izlaznih veza i to na kraj liste.

procedure BrisiVezuU(sender:TObject);

Procedura BrisiVezuU brise samo referencu u listi ulaznih veza ukoliko veza koja je preneta preko sender postoji u liti ulaznih veza.

procedure BrisiVezuI(sender:TObject);

BrisiVezuI radi istu stvar kao i prethodna procedura samo za izlazne veze.

procedure Paint;override;

Procedura Paint kao što joj ime kaže služi za iscrtavanje bloka. Ona se poziva svaki put kada je potrebno osvežiti prikaz na ekranu. Paint iscrtava LokalniBroj bloka u levom gornjem uglu kao i ikonicu iz AktivneSlike i to onu sa indeksom koji je jednak Sifra.

procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y: Integer); override;

Procedura MouseDown postavlja vrednost promenljive MDown na true. Kada se pritisne taster miša u zavisnosti od stanja u kome se nalazi trenutni blok može da se desi sledeće: ako je u stanju stSelektujem trenutni blok će se aktivirati (selektovati) i prebaciti ispred svih ostalih prozora (naravno svi ostali blokovi će se deselektovati jer će se pozvati događaj OnAktiviran), ako je u stanju stDodajemVezu pozvaće se događaji OnBlokClick i OnDodajVezu, i na kraju ako je u stanju stKreiramJunction onda će pozvati OnBlokClick i OnKreirajJunction.

procedure MouseMove(Shift: TShiftState; X, Y: Integer);override;

Procedura MouseMove okida događaj OnBlokMouseMove i u slučaju da je tokom pomeranja miša iznad bloka taster miša bio pritisnut tada će se i sam blok pomeriti. Pomeranje bloka je moguće samo po tačkama grida.

procedure MouseUp(Button: TMouseButton; Shift: TShiftState; X, Y: Integer); override;

Postavlja vrednost MDown na false;

procedure RotirajCC;

Vrši rotiranje smera trenutnog bloka u smeru suprotnom od skazaljke na satu za jedan.

procedure RotirajC;

Vrši rotiranje smera trenutnog bloka u smeru skazaljke na satu za jedan.

function VratiUlaz(i: integer): TCSMPBlok;

Funkcija VratiUlaz vraća blok koji se nalazi na početku veze koja ima redni broj i u listi ulaznih veza.

Private sekcija

Dole navedene promenljive i metode služe samo da enkapsuliraju property-je. Metode uglavnom imaju jednu liniju teksta koja služi za dodeljivanje vrednosti tako da je suvišno opisivati svaku od njih. Prilikom imenovanja promenljivih pridržavali smo se pravila koja je nametnuo Borland te svaka privatna promenljiva u kojoj se čuva vrednost property-ja ima isti naziv kao i property samo što na početku ima i veliko F.
Primer: peoperty Aktivno --- FAktivno.

```
FAktivno: boolean;
FOnAktiviran: TNotifyEvent;
FSmer: TSmer;
FOnPromenaSmera: TNotifyEvent;
FSlikeLevo: TImageList;
FSlikeDesno: TImageList;
FSlikeDole: TImageList;
FSlikeGore: TImageList;
FAktivneSlike: TImageList;
FImageIndex:integer;
FLokalniBroj:integer;
FNaziv:string;
FOnDodajVezu: TNotifyEvent;
FMaxUlaznihVeza: integer;
FOnBlokMouseMove: TNotifyEvent;
FOnKreirajJunction: TNotifyEvent;
FOnBlokClick: TNotifyEvent;
```

```
    procedure SetImageIndex(const Value: integer);
    procedure SetAktivno(const Value: boolean);
    procedure SetNaziv(const Value: string);
    procedure SetOnAktiviran(const Value: TNotifyEvent);
    procedure SetSmer(const Value: TSmer);
    procedure SetOnPromenaSmera(const Value: TNotifyEvent);
    procedure SetSlikeDesno(const Value: TImageList);
    procedure SetSlikeDole(const Value: TImageList);
    procedure SetSlikeGore(const Value: TImageList);
    procedure SetSlikeLevo(const Value: TImageList);
    procedure SetAktivneSlike(const Value: TImageList);
    function GetNaziv: string;
    procedure SetLokalniBroj(const Value: integer);
    function GetImageIndex: integer;
    function GetlokalniBroj: integer;
    procedure SetOnDodajVezu(const Value: TNotifyEvent);
    function GetIzlaznaTacka: TPoint;
    function GetUlaznaTacka: TPoint;
    procedure SetMaxUlaznihVeza(const Value: integer);
    procedure SetOnBlokClick(const Value: TNotifyEvent);
    procedure SetOnBlokMouseMove(const Value: TNotifyEvent);
    procedure SetOnKreirajJunction(const Value: TNotifyEvent);
```

Opis Unit-a CSMPRadnaPovrsina.pas

U Unit-u CSMPRadnaPovrsina.pas se nalazi kod za klasu TCSMPRadnaPovrsina. TCSMPRadnaPovršina je objekta na kome se odvija projektovanje grafika. Snabdeven je svim događajima kojima može da obavesti ostale objekte u projektu da se desilo npr. kreiranje nove veze, novog bloka ili slično. Takođe snabdeven je funkcijama koje mu omogućavaju da preslika postojeće stanje grafika preslika u xml fajl i naravno da kasnije po potrebi da te iste podatke učita iz fajla. Proces enkapsulacije je ispoštovan te je dati objekat moguće lako nadograditi, ponovo koristiti ili izmeniti.

Izgled klase (njenih metoda i polja) je sledeći:

```
TCSMPRadnaPovrsina = class(TCustomControl)
  private
    FSelektovanBlok: TCSMPBlok;
    FOnSelectBlok: TNotifyEvent;
    FSlikeDesno: TImageList;
    FSlikeDole: TImageList;
    FSlikeLevo: TImageList;
```

```
  FSlikeGore: TImageList;

  FDrzaci: TImageList;
  FStanje: TRPStanje;
  FOnStanjeChange: TNotifyEvent;
  FSelektovanaVeza: TVeza;
  FOnChange: TNotifyEvent;
  FSelektovanDrzac: TDrzac;
  FGridSlika:TBitmap;
  FOnDropBlok: TNotifyEvent;
  FOnPoveziVezu: TNotifyEvent;

  procedure VezaDesniKlik(Sender: TObject);
  procedure BlokMouseMove(Sender: Tobject);
  procedure BlokClick(Sender: TObject);
  procedure SetSelektovanBlok(const Value: TCSMPBlok);
  procedure SetOnSelectBlok(const Value: TNotifyEvent);
  procedure SetSlikeDesno(const Value: TImageList);
  procedure SetSlikeDole(const Value: TImageList);
  procedure SetSlikeGore(const Value: TImageList);
  procedure SetSlikeLevo(const Value: TImageList);

  procedure SetDrzaci(const Value: TImageList);
  procedure SetStanje(const Value: TRPStanje);
  procedure SetOnStanjeChange(const Value: TNotifyEvent);
  procedure DrzacKreiran(Sender: TObject);
  procedure SetSelektovanaVeza(const Value: TVeza);
  procedure SetOnChange(const Value: TNotifyEvent);
  procedure BrisiDrzac(Sender: TObject);
  procedure PomerajMisa(Sender: TObject; Shift: TShiftState; X, Y: Integer);
  procedure SetOnDropBlok(const Value: TNotifyEvent);
  procedure SetOnPoveziVezu(const Value: TNotifyEvent);
  { Private declarations }
protected
  { Protected declarations }
  procedure DeselektujVezu(Sender: TObject);
  procedure SelektujVezu(Sender: TObject);
  procedure KeyDown(var Key: Word; Shift: TShiftState);override;
  function NadjiKrajBlok(uVeza:TVeza):TCSMPBlok;
public
  { Public declarations }
  Veze: TList;
  TempIzlaz: TCSMPBlok;
  TempUlaz: TCSMPBlok;
  TempVeza: TVeza;
```

```
    TempDinV: TVeza;
    TempDinBS: TCSMPBlok;
    TempDinD: TDrzac;
    OpisProblema:string;
    procedure Paint;override;
    procedure SelektujDrzac(Sender: TObject);
    procedure DeselektujDrzac(Sender: TObject);
    procedure SelektujJunction(Sender: TObject);
    procedure DeSelektujJunction(Sender: TObject);


    procedure OdrediBrojeve;
    procedure ObrisiDrzac(d: TDrzac);
    procedure ObrisiVeze(uVeze: TList);
    procedure ObrisiVezu(Veza:TVeza);
    procedure ObrisiBlok(uBlok: TCSMPBlok);
    procedure DragDrop(Source: TObject; X, Y: Integer); override;
    procedure DragOver(Source: TObject; X, Y: Integer; State: TDragState;
      var Accept: Boolean); override;
    constructor Create(AOwner:TComponent);override;
    destructor Destroy;override;
    procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
override;
    procedure Aktivacija(Sender:TObject);
    procedure ObrisiSelektovano;

    procedure SaveToFile(FileName:string);
    procedure LoadFromFile(Filename:string);

    function  NadjiBlok(lb:integer):TCSMPBlok;

    procedure ObrisiSve;

    procedure SacuvajVezu(Koren:IXMLNode;Veza:TVeza);
    procedure SacuvajJunction(Koren:IXMLNode;Veza:TVeza);

    procedure UcitajVezu(Koren:IXMLNode;var Veza:TVeza);
    procedure UcitajJunction(Koren:IXMLNode;var Veza:TVeza;drz:TDrzac);
    procedure UcitajDrzac(Koren:IXMLNode;var Veza:TVeza);

    property SelektovanBlok: TCSMPBlok read FSelektovanBlok write
SetSelektovanBlok;
    property SelektovanaVeza: TVeza read FSelektovanaVeza write SetSelektovanaVeza;
  published
    { Published declarations }
    property SlikeLevo:TImageList read FSlikeLevo write SetSlikeLevo;
```

```
    property SlikeDesno:TImageList read FSlikeDesno write SetSlikeDesno;
    property SlikeGore:TImageList read FSlikeGore write SetSlikeGore;
    property SlikeDole:TImageList read FSlikeDole write SetSlikeDole;
    property Drzaci:TImageList read FDrzaci write SetDrzaci;
    property Align default alClient;
    property Color default clWhite;
    property Stanje: TRPStanje read FStanje write SetStanje;
    property OnSelectBlok: TNotifyEvent read FOnSelectBlok write SetOnSelectBlok;
    property OnStanjeChange: TNotifyEvent read FOnStanjeChange write
SetOnStanjeChange;
    property OnChange: TNotifyEvent read FOnChange write SetOnChange;
    property OnDropBlok: TNotifyEvent read FOnDropBlok write SetOnDropBlok;
    property OnPoveziVezu: TNotifyEvent read FOnPoveziVezu write SetOnPoveziVezu;

    property OnKeyDown;
    property OnKeyPress;
    property OnKeyUp;
  end;
```

Published sekcija

property SlikeLevo:TImageList;

property SlikeDesno:TImageList;

property SlikeGore:TImageList;

property SlikeDole:TImageList;

Prethodna četiri property-ja predstavljaju reference na instancu klase TImageList koja u sebi sadrži ikonice blokova u položaju Levo, Desno, Gore i Dole. Ovi objektni se ne instanciraju za svaki blok već predstavljaju deljeni resurs. Napomena: Inicijalizacija ovih referenci mora da se izvši prilikom kreiranja CSMPRadnePovrsine inače može doći do neželjenih efekata prilikom iscrtavanja.

property Drzaci:TImageList;

Ne koristi se više, u slučaju da se u budućnosti javi potreba za promenom izgleda držača može se iskoristiti ovaj TImgList.

property Stanje: TRPStanje;

Property Stanje može da uzme sledeće vrednosti: rpsSelektuj, rpsDodajDrzac, rpsDodajVezuEx, rpsKreirajJunctionEx, rpsBrisem. Postavljanjem vrednosti Stanje na neku od nabrojenih vrednosti menja se način na koji reaguju blokovi na radnoj površini (naravno isto važi i za veze, junction-e i držače).

property OnSelectBlok: TNotifyEvent;

OnSelectBlok je događaj koji se poziva kada se vrednost selektovanog bloka promeni. Ovaj događaj se koristi na glavnoj formi kada želimo da osvežimo prikaz svojstava bloka u CSMPInspektoru.

property OnStanjeChange: TNotifyEvent

Događaj koji se poziva u trenutku kada se stanje CSMPRadnePovrsine promeni. Na ovaj način se podešavaju stanja dugmadi na toolbaru.

property OnDropBlok: TNotifyEvent;

Događaj koji se poziva u trenutku kada se iz Skrolera na CSMPRadnuPovrsinu prevuce stavka (blok). U glavnoj formi se okida procedura UnosParametara koja dozvoljava korisniku da prilikom kreiranja novog bloka unese parametre za njega.

property OnPoveziVezu: TNotifyEvent;

Događaj koji se poziva u trenutku kada se izvrši povezivanje veze od jednog bloka do drugog ili veze i bloka. U glavnoj formi ovaj događaj okida proceduru IzaberiUlaz u kojoj se definiše na koji ulaz će se povezati novoostvarena veza.

Public sekcija

Veze: TList;

Veze predstavlja listu u kojoj se čuvaju sve veze koje su do sada napravljene na grafiku. Ovo je jedino mesto koje predstavlja stvarno skladište za objekte TVeza . Svako dodavanje i brisanje se vrši ovde. Na svim drugin mestima referenciraju se objekti koji se ovde čuvaju.

TempDinV: TVeza;

Promenljiva koja privremeno čuva referencu na vezu koja se dinamički kreira. Ova vrednost se ili trajno čuva u listi veza ili se briše ako se odustane od kreiranja veze.

TempDinBS: TCSMPBlok;

Pomoćna promenljiva u kojoj se čuva početni blok prilikom kreiranja nove veze.

TempDinD: TDrzac;

TempDinD predstavlja pomoćnu promenljivu u kojoj čuvamo držač na koju trenutno dodajemo novu junction vezu.

procedure OdrediBrojeve;

procedura OdrediBrojeve reindeksira sve LokalniBroj promenljive u blokovima koji se nalaze na radnoj površini da bi se njihovi indeksi poklapaju sa indeksima u listi blokova. Prava lista blokova ne postoji iz razloga što je CSMPBlok grafička komponenta nasleđena iz TCustomControl te svaki put kada napravimo novu instancu objekta TCSMPBlok ona se čuva u listi Components objekta TCSMPRadnaPovrsina. Iz istog razloga kreirane CSMPBlokove ne moramo eksplicitno brisati iz memorije prilikom izlaska iz programa.

procedure ObrisiVezu(Veza:TVeza);

Vrlo korisna procedura koja kao parametar uzima vezu koju želimo da obrisemo i pronalai sve blokove ili druge veze i držače koji je referenciraju, briše ove reference i na kraju je fizički briše iz liste veza.

procedure ObrisiBlok(uBlok: TCSMPBlok);

Takođe korisna finkcija koja pronalazi sve veze koje blok koji brisemo referencira i uništava ih pozivom procedure ObrisiVezu i na kraju briše sam blok iz liste blokova.

constructor Create(AOwner:TComponent);override;

destructor Destroy;override;

Procedure koje bezbedno kreiraju i uništavaju sve objekte potrebne za funkcionisanje TCSMPRadnaPovrsina.

procedure ObrisiSelektovano;

Ovo je procedura koja se brine o tome da u slušaju da postoji neki selektovani objekat (blok, veza, junction, drzac) izvrši njegovo bezbedno brisanje iz memorije.

procedure SaveToFile(FileName:string);

procedure LoadFromFile(Filename:string);

Procedure koje su zadužene da preslikaju stanje na radnoj površini u fajl i natrag u radnu površinu.

function NadjiBlok(lb:integer):TCSMPBlok;

Zgodna funkcija za lako dobijanje reference na CSMPBlok sa zadatim LicniBroj-em.

procedure ObrisiSve;

Procedura kojom se iz radne površine bezbedno uklanjaju svi kreirani objekt. Procedura se poziva prilikom kreiranja novog fajla ili učitavanja postojećeg.

property SelektovanBlok: TCSMPBlok;

property SelektovanaVeza: TVeza;

Property koji čuvaju vrednosti trenutno selektovanog bloka ili veze. U slučaju da blok nije selektovan vrednost SelektovanBlok je jednaka nil. Isto važi i za SlelektovanaVeza.

Opis Unit-a Skroler.pas

U unit-u Skroler nalaze se dve klase tesno povezane: TSkroler i TSkrolStavka. TSkroler je "radna površina" za listu TSkrolStavki koje su u suštini učitani blokovi raspoloživi korisniku programa. Klasa TSkrolStavka izgleda ovako:

```
  TSkrolStavka = class(TCustomControl)
  private
    FImageIndex: integer;
    FNaziv: string;
    FLabela: TLabel;
    FMaxUlaznihVeza: integer;
    FBrojParam: integer;

    procedure SetImageIndex(const Value: integer);
    procedure SetNaziv(const Value: string);
    procedure SetLabela(const Value: TLabel);
    procedure SetMaxUlaznihVeza(const Value: integer);
    procedure SetBrojParam(const Value: integer);
  protected
    Stanje: boolean;
  public
    Node: IXMLNode;
    Nazivi: array[0..5] of string;
    constructor Create(AOwner: TComponent; uNode: IXMLNode); overload;
    procedure Paint; override;
    procedure PodesiPolozaj;
    procedure CMMouseEnter(var Message: TMessage); message CM_MOUSEENTER;
    procedure CMMouseLeave(var Message: TMessage); message CM_MOUSELEAVE;
    procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
override;

  published
    property ImageIndex: integer read FImageIndex write SetImageIndex default 0;
    property Naziv: string read FNaziv write SetNaziv;
    property MaxUlaznihVeza: integer read FMaxUlaznihVeza write SetMaxUlaznihVeza;
    property BrojParam: integer read FBrojParam write SetBrojParam;
    property Height default 56;
    property Width default 56;
    property Labela: TLabel read FLabela write SetLabela;
  end;
```

Published sekcija

property ImageIndex: integer;

Kada se učitaju parametri Stavke (bloka) iz XML manifest fajla, učita se i njegova šifra - ImageIndex koja reprezentuje index slike koja će se kasnije učitati iz liste TImageList sličica klase TSkroler.

property Naziv: string;

property MaxUlaznihVeza: integer;

property BrojParam: integer;

property Height default 56;

property Width default 56;

U navedenih pet property-ja se čuvaju: naziv, maximalni broj ulaznih veza, broj parametara, širina i visina TSkrolStavke (bloka), koji se takođe učitavaju, pri kreiranju, iz XML manifest fajla.

property Labela: TLabel;

Pošto se u naziv učita naziv TSkrolStavke (bloka), labela se koristi kako bi se taj naziv zaista i ispisao ispod svake TSkrolStavke u odgovarajućim dimenzijama.

Public sekcija

Nazivi: array[0..5] of string;

Ovde se za svaku TSkrolStavku čuvaju nazivi parametara koje taj, budući, blok može da ima. Primer: za Mrtva zona blok, to su "Donja granica" i "Gornja granica".

constructor Create(AOwner: TComponent; uNode: IXMLNode); overload;

Za kreiranje svake TSkrolStavke prosleđuje se pored AOwner-a i uNode koji reprezentuje XML node iz koga se isčitavaju vrednosti o jednoj TSkrolStavci (bloku).

procedure Paint; override;

U paint procedure se učitava sličica iz TSkroler(parent).Slike sa određenim ImageIndex-om od TSkrolStavke, iscrtava se lep beli okvir i još sitnije dizajnerske stvarčice (razna senčenja i sl.).

procedure PodesiPolozaj;

Ovo je jedna od ključnih procedura u TSkrolStavka. Ova procedura poziva se iz Resize metode TSkroler-a kojemu aktuelna stavka pripada. Ova procedura ima za zadatak da u zavisnosti od FBrCol (pogledajte TSkroler za više detalja) regrupiše blokove u predviđeni broj kolona. Ako je FBrCol=2 to znači da sve TSkrolStavke koje pripadaju TSkroler-u treba organizovati u dve kolone, itd.

procedure CMMouseEnter(var Message: TMessage); message CM_MOUSEENTER;

procedure CMMouseLeave(var Message: TMessage); message CM_MOUSELEAVE;

U ove dve procedure hvata se prisustvo miša preko TSkrolStavke. Ako je miš na TSkrolStavci, briše se senka po default-u iscrtana ispod nje, čime se dobija efekat da je stavka utonula i obratno, ponovo se iscrtava senka ako je miš napustio objekat.

procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y: Integer); override;

Aktivira se Drag&Drop mehanizam, automatski podržan od strane Delphi-ja.

TSkroler klasa izgleda ovako:

```
 TSkroler = class(TScrollBox)
 private
   FSlike: TImageList;
   FbrCol: Byte;
   FOldBrCol: Byte;
   procedure SetSlike(const Value: TImageList);
 public
   constructor Create(AOwner: TComponent); override;
   destructor Destroy; override;
   procedure LoadFromXMLManifest(Stavke: IXMLNodeList);
   procedure Resize; override;
 published
   property Slike: TImageList read FSlike write SetSlike;
   property Color default clWhite;
   property BorderStyle default bsNone;
 end;
```

Published sekcija

property Slike: TImageList read FSlike write SetSlike;

TSkroler objektu se dodeljuje lista sličica sa blokovima. On koristi ovu listu pri učitavanju svake TSkrolStavke, na taj način što za svaki ImageIndex TSkrolStavke učitava sliku sa tim index-om iz ovog property-ja.

property Color default clWhite;

Ovaj property čuva informaciju o boji TSkroler-a.

Public sekcija

constructor Create(AOwner: TComponent); override;

Skoro ništa se ne odigrava u ovom konstruktoru van okvira Delphi default konstruktora TScrollBox komponente.

destructor Destroy; override;

Takođe u potpunosti nasleđen je i destructor.

procedure LoadFromXMLManifest(Stavke: IXMLNodeList);

Za prosleđeni XML manifest fajl, kreiraju se TSkrolStavke sa atributima iz ovog manifesta. Takođe, Labela svake TSkrolStavke dobija ovde vrednost Naziva te stavke, i koriguje se širina te Labele na širinu samog bloka/učitane sličice.

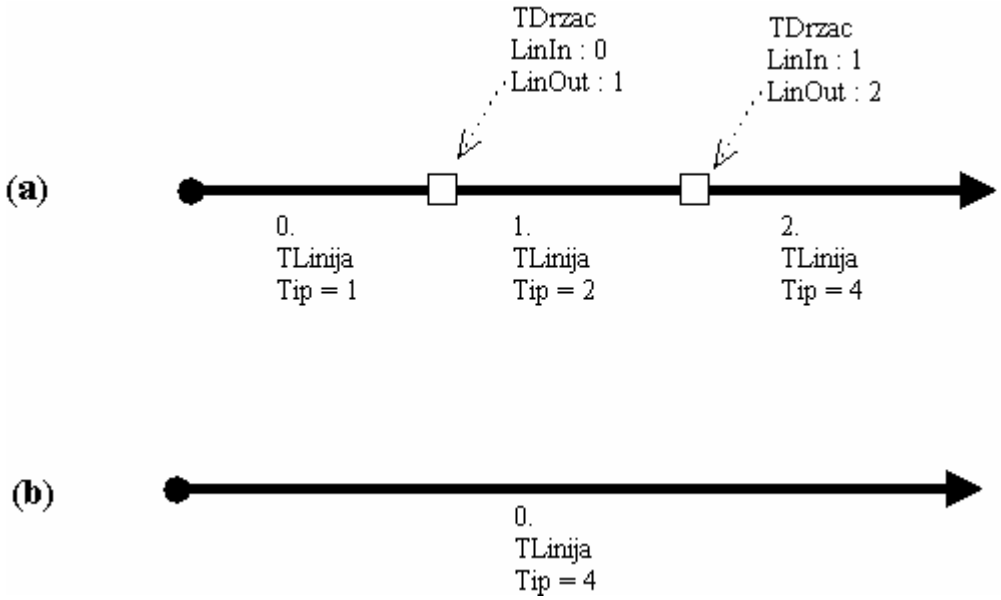
procedure Resize; override;

Ideja je da se sa promenom veličine TSkrolera menja i broj kolona (FBrCol) u kojima se iscrtavaju TSkrolStavke. Ako je width TSkrolera veća od jedne TSkrolStavke (60pix sa sve Labelom), onda se vrši deljenje po sledećoj formuli

```
FBrCol := ((self.Width+15) div 80);
```

Postoji i pomoćna promenljiva FOldBrCol koja čuva staru vrednost (za jednu unazad) FBrCol-a, te kada FBrCol dobije novu vrednost (različitu od FOldBrCol) poziva se za svaku stavku metoda PodesiPolozaj koja ima za zadatak da ponovo iscrta sve TSkrolStavke u novodobijenom broju kolona.

Kao mali uvod, pre svega, pogledajmo dve od mogućih struktura istance klase TVeza.



Opis Unit-a unLineEx.pas

U Unit-u unLineEx.pas se nalazi kod za klasu TLineEx. Izgled klase (njenih metoda i polja) je sledeći:

```
type TLineEx = class(TLine)
  private
    FTip: Byte;
    FKrugColor: TColor;
  protected
    SA: TShowArrows;
    procedure SetTip(Value: Byte); virtual;
    procedure SetKrugColor(Value: TColor); virtual;
    procedure Paint; override;
    procedure WMErase(var msg:TMessage);message WM_ERASEBKGND;
  published
    property Tip: Byte
    read FTip
      write SetTip
      default 1;
    property KrugColor: TColor
      read FKrugColor
      write SetKrugColor
```

```
    default clBlack;
 public
   constructor Create(AOwner: TComponent); override;
 end;
```

Kao što se vidi klasa TLineEx nasleđuje klasu TLine. Klasa TLine je osnovna klasa za iscrtavanje linije (opciono, sa strelicom). Na ovaj način dodajemo neke dodatne funkcije običnoj liniji.

Published sekcija

property Tip:Byte;

Ovaj property je i sama suština zbog koje smo nasledili klasu TLine. Ako je vrednost ovog property-ja: 1 time smo naznačili liniju za početnu te trebamo da iscrtamo kružić na početku), 2 linija je u sredini te ne treba da ima ni kružić ni strelicu na sebi, 3 za krajnju liniju (strelica na kraju) i 4 za liniju bez držača dakle i krug i strelica.

property KrugColor:TColor;

Omogućava vam da promenite boju kružića (default: clBlack) pri iscrtavanju.

procedure Paint; override;

Ova procedura prekriva nasleđenu iz klase TLine. Ako je Tip 1 ili 2 onda nije potrebno da iscrtavamo strelicu na kraju. Ako je Tip 1 ili 4 onda je potrebno iscrtati kružić (definisane boje sa SetColor) na početku ove linije. Ako je Tip 3 nije potrebno iscrtati ništa dodatno, samo liniju.

Opis Unit-a unLinija.pas

U Unit-u unLinija.pas se nalazi kod za klasu TLinija. Izgled klase (njenih metoda i polja) je sledeći:

```
type TLinija = class(TLineEx)
  private
    FBoja: TColor;
    FSP : TPoint;
    FEP : TPoint;
    FSelected: Boolean;
  protected
  published
  public
    constructor Create(AOwner: TComponent); override;
```

```
    function GetSelektovan: Boolean;
    function GetSP : TPoint;
    function GetEP : TPoint;
    procedure SetSelektovan(Value: Boolean);
    procedure SetSEP(sp, ep: TPoint);
    procedure SetBoja(C: TColor);
  end;
```

Kao što se vidi ova klasa nasleđuje gore opisanu klasu TLineEx, dodajući joj još neke važne karakteristike.

Public sekcija

procedure SetSEP(sp,ep:TPoint);

Ova procedura je akronim od "Set Starting Ending Point". Za prosleđene tačke sp i ep ona iscrtava liniju od sp do ep i postavlja enkapsulirane atribute FSP i FEP na vrednosti prosleđenih argumenata.

procedure SetBoja(C:TColor);

Ova procedura boji liniju ALI I KRUŽIĆ (nasleđeni property KrugColor) prosleđenom bojom i postavlja enkapsulirani atribut FBoja na ovu vrednost.

procedure SetSelektovan(Value:Boolean);

Ova procedura (de)selektuje liniju, na taj način što je iscrtava kao Dot ako je Value=True ili Solid ako je False i postavlja učaureni atribut FSelected na Value vrednost.

Opis Unit-a unDrzac.pas

U Unit-u unDrzac.pas se nalazi kod za klasu TDrzac. Izgled klase (njenih metoda i polja) je sledeći:

```
  TDrzac = class(TCustomControl)
  private
    FImageIndex: integer;
    FAktivno: boolean;
    FOnDrzacMove: TNotifyEvent;
    FOnDrzacClick: TNotifyEvent;
    FOnDrzacFree: TNotifyEvent;
    FJunction: TList;
    FOnDeleteJunction: TNotifyEvent;
    procedure SetImageIndex(const Value: integer);
    procedure SetAktivno(const Value: boolean);
```

```
    procedure SetOnDrzacMove(const Value: TNotifyEvent);
    procedure SetOnDrzacClick(const Value: TNotifyEvent);
    procedure SetOnDrzacFree(const Value: TNotifyEvent);
    procedure SetOnDeleteJunction(const Value: TNotifyEvent);
  protected
    MDown: boolean;
    nX, nY: integer;
  public
    LinIn: TLinija;
    LinOut: TLinija;
    constructor Create(AOwner: TComponent); override;
    function GetCentarXY: TPoint;
    procedure Paint; override;
    procedure MouseDown(Button: TMouseButton; Shift: TShiftState;
      X, Y: Integer); override;
    procedure MouseMove(Shift: TShiftState; X, Y: Integer); override;
    procedure MouseUp(Button: TMouseButton; Shift: TShiftState;
      X, Y: Integer); override;
    destructor Destroy; override;
    procedure AddJunction(j: TObject);
    procedure DeleteJunction(j: TObject);
    function GetJunctionList: TList;
  published
    property Aktivno: boolean read FAktivno write SetAktivno;
    property OnDrzacMove: TNotifyEvent read FOnDrzacMove write SetOnDrzacMove;
    property OnDrzacClick: TNotifyEvent read FOnDrzacClick write SetOnDrzacClick;
    property OnDrzacFree: TNotifyEvent read FOnDrzacFree write SetOnDrzacFree;
    property OnDeleteJunction: TNotifyEvent read FOnDeleteJunction write
SetOnDeleteJunction;
  end;
```

Držač je komponenta koja pri pomeranju sa sobom pomera i LineIn i LineOut. To su pokazivači na dve TLinije između kojih se sam držač i nalazi.

Public sekcija

property Aktivno:boolean;

Kada je vrednost propertyja True to znači da je držač selektovan, te da ga treba iscrtati kao selektovanog. Kada držač nije selektovan iscrtava se kao običan kvadratić. Kada je selektovan iscrtava se kao kvadratić sa linijom debljine (width) 2. Kada je držač ustvari Junction tada se on iscrtava kao pun crn kružić. Kada je držač Junction i pri tome je selektovan onda se on iscrtava samo kao kružić (belina u sredini).

property OnDrzacMove;

property OnDrzacClick;

property OnDrzacFree;

property OnDeleteJunction;

Ova četiri Event-a dižu se kada (respektivno): je držač pomeren, je kliknuto na držač, držač treba da se uništi (Destroy), je jedan od Junction-a obrisan. Naime, držač može biti običan držač koji služi za ispravljanje linije ali može biti i Junction ako je na njega nakačena jedna ili više veza. Svaka od tih veza se procedurom AddJunction ubacuje u listu veza na koju pokazuje TDrzac i koje treba pomeriti kada se i on sam pomeri, ali i po brisanju neke od tih veza se diže poslednje navedeni Event.

LinIn:TLinija;

Pokazivač na liniju, TLinija, koja ulazi u držač.

LinOut:TLinija;

Pokazivač na liniju, TLinija, koja izlazi iz držača.

function GetCentarXY:TPoint;

Vraća tačku koja je centar (left+(width div 2) i top+(height div 2)) držača kako bi se linije koje ulaze i izlaze iz njega postavile na ovu tačku pri iscrtavanju.

procedure Paint; override;

Kada je vrednost propertyja Aktivno True to znači da je držač selektovan, te da ga treba iscrtati kao selektovanog. Kada držač nije selektovan iscrtava se kao običan kvadratić. Kada je selektovan iscrtava se kao kvadratić sa linijom debljine (width) 2. Kada je držač ustvari Junction tada se on iscrtava kao pun crn kružić. Kada je držač Junction i pri tome je selektovan onda se on iscrtava samo kao kružić (belina u sredini).

procedure MouseDown;

procedure MouseUp;

procedure MouseMove;

Ove dve procedure su od značaja samo za generisanje događaja. Prve dve procedure su zadužene za to da podignu događaj OnDrzacClick, dok je poslednja zadužena za događaj OnDrzacMove.

procedure AddJunction(j:TObject);

Pointer na TVeza se prosleđuje ovoj metodi koja ima zadatak taj pointer da ubaci u listu pointer-a koja predstavlja skup veza koje držač "Handle-uje". Pri pomeranju držača prolazi se kroz listu ovih pokazivača koja se čuva u enkapsuliranom atributu FJunction:TList i sve se veze pomeraju na nove koordinate koje dobijaju sa GetCenterXY držača.

procedure DeleteJunction(j:TObject);

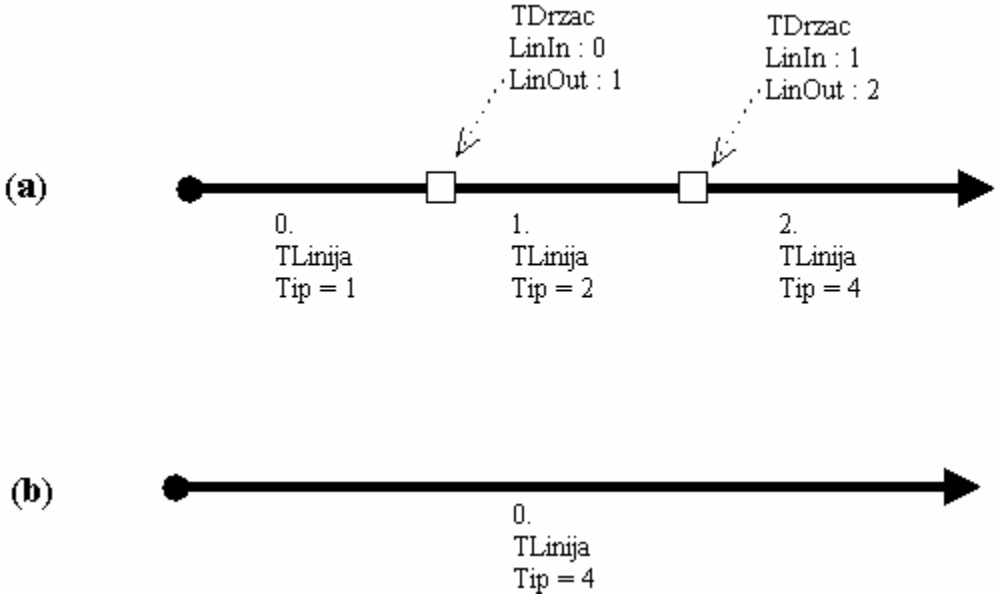
Ovaj procedura uklanja pokazivač i liste Veza na koje pokazuje držač.

function GetJunctionList:TList;

Ova funkcija je, može se reći, jedna od ključnih za komunikaciju na relaciji TVeza-TDržač. Kada se Držač pomeri, on generiše OnDrzacMove u TVeza objektu. Tada TVeza uzima (sa GetJunctionList) FJunction, prolazi kroz tu listu i pomera zaista objekte prikačene na taj držač. Dakle, ova funkcija vraća enkapsuliranu listu Veza na koje pokazuje sam držač.

Opis Unit-a unVeza.pas

U Unit-u unVeza.pas se nalazi kod za klasu TVeza. Pogledajmo još jednom sledeću sliku:



Na njoj je prikazana struktura objekta TVeza. TVeza je složeni objekat koji je izgrađen (using) od više različitih, gore opisanih, objekata: TLinija, TDrzac. Objekat treba da se zakači za blok ili drzač (u slučaju kada je u pitanju Junction) i da njihovim pomeranjem/brisanjem izvrši odgovarajuće akcije.

```
type TStanja = (stNone, stKreirajDrzac, stPripremiZaJunction, stKreirajJunction);
type TVeza = class(TObject)
  private
    LLinija: TList;
    FBoja: TColor;
    FStanje: TStanja;
    sp: TPoint;
    ep: TPoint;
    C: TWinControl;
    FDrzaci: TImageList;
    FDrzacKreiran: TNotifyEvent;
    FSelectedDrzac: TDrzac;
    FLastDrzac: TDrzac;
    FOnSelektVeza: TNotifyEvent;
    FOnDeselectVeza: TNotifyEvent;
    FOnSelektDrzac: TNotifyEvent;
    FOnDeselectDrzac: TNotifyEvent;
    FSelektovano: boolean;
    FOnBrisanjeI: TNotifyEvent;
    FOnBrisanjeU: TNotifyEvent;
    FOnDeselectJunction: TNotifyEvent;
    FOnSelectJunction: TNotifyEvent;
    FIsJunction: boolean;
    FOnBrisiJunction: TNotifyEvent;
    FOnDrzacFree: TNotifyEvent;
    FOnMouseClickEx: TNotifyEvent;
    FPocBlok: TObject;
    FKrajBlok: TObject;
    procedure DrzacFree(Sender: TObject);
    function NadjiIndexL(l: PLinija): integer;
    function NadjiIndexD(d: PDrzac): integer;
    function VratiIndexDrzacaUKojiUvire(pl: PLinija): Integer;
    function DaLiKreiratiDrzac(pos: TPoint): Boolean;
    procedure RefreshLinija;
    procedure Paint;
    procedure SetDrzacKreiran(const Value: TNotifyEvent);
    procedure DrzacClick(Sender: TObject);
    procedure SetOnSelektVeza(const Value: TNotifyEvent);
    procedure SetOnDeselectVeza(const Value: TNotifyEvent);
    procedure SetOnSelektDrzac(const Value: TNotifyEvent);
    procedure SetOnDeselectDrzac(const Value: TNotifyEvent);
    procedure SetSelektovano(const Value: boolean);
    procedure SetOnBrisanjeI(const Value: TNotifyEvent);
    procedure SetOnBrisanjeU(const Value: TNotifyEvent);
    procedure SetOnDeselectJunction(const Value: TNotifyEvent);
    procedure SetOnSelectJunction(const Value: TNotifyEvent);
```

```
    procedure SetOnBrisiJunction(const Value: TNotifyEvent);
    procedure SetIsJunction(const Value: Boolean);
    procedure SetOnMouseClickEx(const Value: TNotifyEvent);
    procedure SetOnDrzacFree(const Value: TNotifyEvent);
    procedure SetStanje(const Value: TStanja);
    procedure SetKrajBlok(const Value: TObject);
    procedure SetPocBlok(const Value: TObject);
  public
    Parent:TVeza;
    LDrzaca: TList;
    procedure SelektovanJunction(Sender: TObject);
    procedure DeSelektovanJunction(Sender: TObject);
    procedure BrisiJunction(sender: TObject);
    procedure KreirajDrzac(p:TPoint);
    constructor Create(ICanvas: TWinControl; sp, ep: TPoint);
    procedure ObrisiDrzacM(Dr:TDrzac);
    procedure SetColor(C: TColor);
    procedure SetSEP(sp, ep: TPoint);
    procedure SetSP(sp: TPoint);
    procedure SetEP(ep: TPoint);
    function Selected: Boolean;
    procedure ObrisiSelektovanDrzac;
    procedure UnSelectAll;
    procedure UnSelectAllExcept(d: PDrzac); overload;
    procedure UnSelectAllExcept(junc: TVeza); overload;
    procedure SelectLinije;
    procedure DrzacSeMrdnuo(sender: TObject);
    destructor Destroy; override;
    function GetSelectedDrzac: TDrzac;
    procedure KreirajJunction(ep1: TPoint);
    function GetLastDrzac: TDrzac;
    procedure MouseUp(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X,
Y: Integer);
    procedure DeleteJunction(j:TVeza);

  published
    property Selektovano: boolean read FSelektovano write SetSelektovano;
    property IsJunction: boolean read FIsJunction write SetIsJunction;
    property DrzacKreiran: TNotifyEvent read FDrzacKreiran write SetDrzacKreiran;
    property OnSelektVeza: TNotifyEvent read FOnSelektVeza write SetOnSelektVeza;
    property OnDeselectVeza: TNotifyEvent read FOnDeselectVeza write
SetOnDeselectVeza;
    property OnSelektDrzac: TNotifyEvent read FOnSelektDrzac write SetOnSelektDrzac;
    property OnDeselectDrzac: TNotifyEvent read FOnDeselectDrzac write
SetOnDeselectDrzac;
    property OnBrisanjeU: TNotifyEvent read FOnBrisanjeU write SetOnBrisanjeU;
```

```
    property OnBrisanjeI: TNotifyEvent read FOnBrisanjeI write SetOnBrisanjeI;
    property OnSelectJunction: TNotifyEvent read FOnSelectJunction write
SetOnSelectJunction;
    property OnDeSelectJunction: TNotifyEvent read FOnDeSelectJunction write
SetOnDeSelectJunction;
    property OnBrisiJunction: TNotifyEvent read FOnBrisiJunction write
SetOnBrisiJunction;
    property OnMouseClickEx: TNotifyEvent read FOnMouseClickEx write
SetOnMouseClickEx;
    property OnDrzacFree: TNotifyEvent read FOnDrzacFree write SetOnDrzacFree;
    property Stanje: TStanja read FStanje write SetStanje;
    property PocBlok: TObject read FPocBlok write SetPocBlok;
    property KrajBlok: TObject read FKrajBlok write SetKrajBlok;
  end;
```

Published sekcija

property Selektovano: boolean;

Ovaj property služi za menjanje stanja veze. Naime Veza, pored ostalih stanja, može biti i u stanju Selektovano=True što znači da je potrbno sve TLinije koje sadrži iscrtati kao Aktivno=True (selektovane). Kada je ovaj properti false sve linije se iscrtavaju kao Solid.

property IsJunction: boolean;

Ako je veza nakačena na Junction sa jedne strane i Blok sa druge onda je ovaj property TVeze potrebno postaviti na True, iz razloga kako bi se ova veza razlikovala od "glavne" veze koja je nakačena na neki početni i krajnji TCSMPBlok.

property DrzacKreiran: TNotifyEvent;

Ovo je Event koji se podiže kada je TDrzac kreiran negde na TVeza (ovom objektu).

property OnSelektVeza: TNotifyEvent;

Kada korisnik klikne na bilo koju TLiniju u sklopu veze, podiže se ovaj Event i izvršava selektovanje ostalih TLinija koje čine TVeza.

property OnDeselectVeza: TNotifyEvent;

Kada korisnik deselektuje bilo koju TLiniju u sklopu veze (klikne na nju kada je ona selektovana ili klikne na radnu površinu), podiže se ovaj Event i izvršava deselektovanje ostalih TLinija koje čine TVeza.

property OnSelektDrzac: TNotifyEvent;

Kada korisnik klikne na držač koji je pre toga bio u stanju Selektovan=False (nije bio selektovan) podiže se ovaj Event.

property OnDeselectDrzac: TNotifyEvent;

Kada korisnik klikne na držač koji je pre toga bio u stanju Selektovan=True (bio selektovan) podiže se ovaj Event.

property OnBrisanjeU: TNotifyEvent;

Pri uništavanju ovog objekta podiže se ovaj Event kako bi Blok na koji je nakačena ova veza znao da je ona uništena, te kako bi je izbacio iz svoje liste referenci na TVeza objekte.

property OnBrisanjeI: TNotifyEvent;

Pri uništavanju ovog objekta podiže se ovaj Event kako bi Blok na koji je nakačena ova veza znao da je ona uništena, te kako bi je izbacio iz svoje liste referenci na TVeza objekte.

property OnSelectJunction: TNotifyEvent;

Kada korisnik klikne na bilo koju TLiniju u sklopu veze, a pri tome je svojstvo IsJunction=True, podiže se ovaj Event i izvršava selektovanje ostalih TLinija koje čine TVeza.

property OnDeSelectJunction: TNotifyEvent;

Kada korisnik deselektuje bilo koju TLiniju u sklopu veze (klikne na nju kada je ona selektovana ili klikne na radnu površinu), a pri tome je svojstvo IsJunction=False, podiže se ovaj Event i izvršava deselektovanje ostalih TLinija koje čine TVeza.

property OnBrisiJunction: TNotifyEvent;

Događaj koji se diže kada TDrzac da znak (OnDeleteJunction) da je sa njega obrisana veza koja ima svojstvo IsJunction=True.

property OnMouseClickEx: TNotifyEvent;

Događaj koji se podiže kada je kliknuto na vezu (bilo koju TLiniju) a pri tome je kliknuto desnim tasterom. Ovaj događaj "hvata" CSMPRadnaPovrsina brišući celu vezu ako je ona u stanju KreirajJunction, jer je to znak da je veza u procesu kreiranja, a da je korisnik kliknuo desnim tasterom što znaci da je odustao od kreiranja veze.

property OnDrzacFree: TNotifyEvent;

Kada se TDrzac koji pripada vezi obriše diže se ovaj Event.

property Stanje: TStanja;

Property "Stanje" predstavlja aktuelno stanje veze i može uzeti jednu od vrednosti TStanja=(stNone, stKreirajDrzac, stPripremiZaJunction, stKreirajJunction). Kada je veza u stanju stNone to znači da se klikom na nju neće ništa desiti osim što će se ako je selektovana deselektovati i obrnuto. Ako je veza u stanju stKreirajDrzac, klikom na nju (bilo koju TLiniju koja joj pripada) u toj tački će se kreirati TDrzac objekat i on će se smestiti u odgovarajući listu LDrzaca. Kada korisnik u programu klikne na taster "Kreiraj

junction", CSMPRadnaPovrsina prolazi kroz SVE TVeza objekte i njihova stanja postavlja na stPripremiZaJunction. Tek kada se klikne na određenu vezu, ona ako je (a jeste) u stanju stPripremiZaJunction počinje da kreira Junction. Klikom na odredišni blok, ona prelazi u stanje stKreirajJunction a veza se kreira.

property PocBlok: TObject;

Pokazivač na TCSMPBlok (u slučaju obične veze) ili TDrzac (u slučaju kada je Junction) iz koga TVeza izlazi.

property KrajBlok: TObject;

Pokazivač na TCSMPBlok u koji TVeza ulazi.

Public sekcija

Parent:TVeza;

U slučaju kada se kreira veza na neki Junction, tada takva veza ima "Parent" koji je ustvari veza na kojoj se nalazi sam Junction. Ovaj atribut neophodan je zbog rekurzivnog prolaska kroz sve veze, veze koje izlaze iz njih (Junction vezom), veze koje izlaze iz njih (Junction vezom) ...

LDrzaca: TList;

Ovo je lista svih TDrzaca koji se nalaze na jednoj vezi. Držač, kao što smo već naveli može biti običan držač pri čemu ima funkciju da koriguje vezu (prelama je) onako kako vi želite, a može biti i Junction kada iz njega izlazi jedna ili više veza.

procedure SelektovanJunction(Sender: TObject);

Ako je IsJunction=True podiže se Event OnSelectJunction koji hvata i obrađuje CSMPRadnaPovrsina. Takođe, vrši se deselekcija svih držača koji joj pripadaju.

procedure DeSelektovanJunction(Sender: TObject);

Ako je IsJunction=True podiže se Event OnDeselectJunction koji hvata i obrađuje CSMPRadnaPovrsina. Takođe, vrši se deselekcija svih držača koji joj pripadaju.

procedure BrisiJunction(sender: TObject);

Ova procedura diže Event OnBrisiJunction, koji hvata i obražuje CSMPRadnaPovrsina. Ova procedura se poziva kada je kliknut taster "Obrisi" a postoji TVeza koja je selektovana i ima property IsJunction=True.

procedure KreirajDrzac(p:TPoint);

Za prosleđenu tačku p, ova procedura poziva MouseUp događaj (koji će kasnije biti objašnjen) prosleđujući mu kao parametre X i Y, ovu tačku.

constructor Create(ICanvas: TWinControl; sp, ep: TPoint);

Default konstuktor koji kreira TVeza objekat, inicijalizuje ga postavljajući neke default vrednosti kao što su: neselektovana crna TLinija, inicijalizuje liste LDrzaca i LLinija, postavlja SP (StartingPoint) i EP (EndingPoint) na prosleđene vrednosti. SP je enkapsulirana promenljiva koja predstavlja početnu tačku veze, tačku odakle treba početi iscrtavanje, dok EP predstavlja kranju tačku.

procedure ObrisiDrzacM(Dr:TDrzac);

Za prosleđeni pokazivač na TDrzac, veza ga pronalazi u svojoj listi držača i briše ga - destroy.

procedure SetColor(C: TColor);

Postavljanje boje veze. Po default-u je clBlack, ali se boja veze može promeniti ovom metodom u bilo koji TColor.

procedure SetSEP(sp, ep: TPoint);

procedure SetSP(sp: TPoint);

procedure SetEP(ep: TPoint);

Ove tri metode zadužene su za menjanje enkapsuliranih atributa SP i EP. Kao što smo gore naveli, SP predstavlja tačku iz koje treba početi iscrtavanje (izlaz iz TCSMPBloka) dok EP predstavlja krajnju tačku do koje treba iscrtati TVeza (tj. pripadajuće joj TLinija objekte).

function Selected: Boolean;

Vraća vrednost boolean da li je veza selektovana ili ne.

procedure ObrisiSelektovanDrzac;

Ovaj metod poziva CSMPRadnaPovrsina, prolazeći kroz SVE veze. Ukoliko u vezi postoji selektovani držač on se briše. TVeza prolazi kroz listu LDrzaca, ispituje za svaki da li je property Aktivno postavljeno na True i ako jeste uništava taj TDrzac.

procedure UnSelectAll;

Prolazi kroz liste LDrzaca i LLinija i deselektuje (SetSelektovano=False i Aktivno=False) sve objekte.

procedure UnSelectAllExcept(d: PDrzac); overload;

Prolazi kroz liste LDrzaca i LLinija i deselektuje (SetSelektovano=False i Aktivno=False) sve objekte izuzev prosleđenog držača d.

procedure UnSelectAllExcept(junc: TVeza); overload;

Prolazi kroz liste LDrzaca i LLinija i deselektuje (SetSelektovano=False i Aktivno=False) sve objekte izuzev prosleđene veze junc.

procedure SelectLinije;

Selektuje sve TLinija objekte iz liste LLinija, a deselektuje sve držače iz liste LDrzaca.

procedure DrzacSeMrdnuo(sender: TObject);

Kada se neki držač pomeri, uzima se metodom GetJunctionList lista svih veza na koje pokazuje taj TDrzac (cast-ovani sender), prolazi se kroz nju, i sve SP tačke tih veza se metodom SetSP postavljaju na novu vrednost. Koju vrednost? Pa vrednost cast-ovanog sender-a u TDrzac pa GetCenterXY.

destructor Destroy; override;

Pri uništavanju veze potrebno je uništiti i sve držače, linije i Junction-e koji joj pripadaju. To se upravo radi u destruktoru ovog objekta.

function GetSelectedDrzac: TDrzac;

Prolazi kroz listu LDrzaca i pronalazi i vraća kao rezultat TDrzac koji je Aktivan.

procedure KreirajJunction(ep1: TPoint);

Kada se klikne na krajnji CSMPBlok, iz CSMPRadnePovrsine poziva se ovaj metod i prosleđuje mu se koordinata ulaza. Kreira se veza potom sa IsJunction=True koja kao EP ima prosleđeni argument.

function GetLastDrzac: TDrzac;

Za potrebe CSMPRadnePovrsine obezbeđena je i ova funkcija koja se oslanja na ucaureni FLastDrzac koji pretstavlja poslednje kliknuti TDrzac u vezi (bilo on selektovan ili ne).

procedure MouseUp(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

U ovoj proceduri se odigrava suština cele veze. Kada se klikne na vezu u zavisnosti od stanja u kojoj se ona nalazi ona se ili selektuje ili deselektuje, ili se na Xi Y kreira držač. Kako se kreira držač? Pre svega, ispituje se da li na tim koordinatama već postoji neki držač. Ako postoji onda se odustaje od kreiranja. Ako ne postoji, kreira se novi. TLinija na koju je kliknuto se vizuelno "cepa" na dva dela, a ispod haube, ona na koju je kliknuto se povlači tako da ona sada faktički ulazi u TDrzac (njen EP se postavlja na GetCentarXY od držača), a kreira se nova čiji se SP postavlja na sredinu držača, dok se EP postavlja na kraj do tadašnje TLinije koja se pomerila. I novo kreirani držač i novo kreirana TLinija se ubacuju u liste LDrzaca i LLinija.

procedure DeleteJunction(j:TVeza);

Prolazi se kroz listu držača LDrzaca i pronalazi se onaj držač koji u svojoj listi Junction veza ima prosledjeni parametar. Kada se nadje taj držač, iz njega se briše referenca na tu vezu.

Opis Unit-a CSMPDugme.pas

CSMPDugme je taster koji je zamena za standardno dugme. Ova komponenta je sastavni deo komponenti CSMPInspektor i CSMPGrupa. Svrha ove komponente je da simulira rad običnog dugmeta i da se svojim izgledom potpuno uklopi u okolinu u koju je stavljeno – u ovom slučaju u dizajn celog programa.

Poseban nabrojiv tip podataka koji se koristi u realizaciji CSMPDugmeta jeste TStanjeMisa:

TStanjeMisa = (smUnutar, smVan, smDole)

TStanjeMisa je nabrojivi tip koji služi da CSMPDugme u svakom trenutku zna kakvo je stanje miša u odnosu na njega:

smUnutar - pokazivač miša je unutar okvira dugmeta (iznad komponente)
smVan - pokazivač miša je van okvira dugmeta
smDole - dugme misa je pritisnuto (CSMPDugme je kliknuto)

Ova promenljiva je neophodna jer je kod iscrtavanja CSMPDugmeta bitno gde de nalazi miš kako bi se ono iscrtalo na adekvatan način.

Sama komponenta definisana je na sledeći način:

```
TCSMPDugme = class(TCustomControl)
private
  FAktivniPanel: TPanel;
  FBoja: TColor;
  FBojaOkvira: TColor;
  FStanjeMisa: TStanjeMisa;
  FTekst: string;
  FOnClick: TNotifyEvent;
  Label1: TLabel;

  procedure SetAktivniPanel(const Value: TPanel);
  procedure SetBoja(const Value: TColor);
  procedure SetBojaOkvira(const Value: TColor);
  procedure SetTekst(const Value: string);
  procedure SetOnClick(const Value: TNotifyEvent);
  procedure LabelMouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
  procedure LabelMouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);

protected
  procedure CMMouseEnter( var Msg: TMessage ); message cm_MouseEnter;
  procedure CMMouseLeave( var Msg: TMessage ); message cm_MouseLeave;
```

```
    procedure MouseDown(Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer); override;
    procedure MouseUp(Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer); override;

  public
    constructor Create(AOwner: TComponent); override;
    procedure Paint; override;

  published
    property AktivniPanel: TPanel read FAktivniPanel write SetAktivniPanel;
    property Boja: TColor read FBoja write SetBoja;
    property BojaOkvira: TColor read FBojaOkvira write SetBojaOkvira;
    property Tekst: string read FTekst write SetTekst;
    property OnClick: TNotifyEvent read FOnClick write SetOnClick;
  end;
```

Private sekcija

Label1: TLabel

Label1 predstavlja tekst koji CSMPDugme nosi. On se nalazi preko cele komponente tako da sve operacije koje se događaju vezane za miša idu preko njega.

FStanjeMisa: TStanjeMisa

Ova promenljiva čuva podatak o trenutnom stanju miša u odnosu na komponentu. Ovo je bitno sa stanovišta procedure Paint.

procedure LabelMouseDown(Sender:TObject;Button:TMouseButton;Shift:TshiftStateX,Y:Integer)
procedure LabelMouseUp(Sender:TObject;Button:TMouseButton;Shift:TShiftState; X, Y: Integer)

Ove procedure se izvršavaju kada korisnik klikne na *Label1* i one događaje samo prosleđuju procedurama *MouseDown* i *MouseUp*.

Protected sekcija

procedure CMMouseEnter(var Msg: TMessage); message cm_MouseEnter
procedure CMMouseLeave(var Msg: TMessage); message cm_MouseLeave

Ove procedure se pozivaju od strane sistema u trenutku kada pokazivač miša „napusti" ili „uđe" u pravougaonik koji predstavlja okvir komponente. Ove događaje je neophodno pratiti jer je boja teksta različita kada je miš iznad dugmeta i kada nije – ažurira se promenljiva FStanjeMisa.

procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y: Integer); override
procedure MouseUp(Button: TMouseButton; Shift: TShiftState; X, Y: Integer); override

Ove procedure promenljivoj FStanjeMisa dodeljuju odgovarajuće vrednosti obaveštavajući proceduru Paint o nastaloj promeni.

Public sekcija

constructor Create(AOwner: TComponent); override

Ovde se postavljaju početne vrednosti za promenljive i izvršavaju početna podešavanja.

procedure Paint; override

Procedura zadužena za crtanje komponente. Iscrtava se okvir bojom *BojaOkvira*, taj okvir se popunjava bojom *Boja* i tekst se ispisuje u zavisnosti od trenutnog položaja miša.

Odeljak PUBLISHED

property AktivniPanel: TPanel read FAktivniPanel write SetAktivniPanel

AktivniPanel je svojstvo koje sadrži instancu na panel čije će stanje da se menja klikom na CSMPDugme.

property Boja: TColor read FBoja write SetBoja

Boja je glavna boja CSMPDugmeta.

property BojaOkvira: TColor read FBojaOkvira write SetBojaOkvira

BojaOkvira je boja kojom će da se iscrta okvir za CSMPDugme.

property Tekst: string read FTekst write SetTekst

Tekst je string koji se ispisuje na CSMPDugmetu.

property OnClick: TNotifyEvent read FOnClick write SetOnClick

OnClick je događaj koji će da se desi (ukoliko je postavljen) kada se CSMPDugme klikne i prvenstveno je namenjen za promenu ponašanja aktivnog panela, ali nije vezan za njega.

Opis Unit-a CSMPGrupe.pas

U prethodnoj verzji programa svi blokovi koje je korisnik želeo da uvrsti u svoj model nalazili su se iznad površine za model. Autori nove verzije, u razgovoru sa korisnicima programa, došli su do zaključka da bi razbijanje ove liste na grupe koje sadrže srodne blokove bilo od koristi pri navigaciji kod stvaranja modela. Iz tog razloga nastale su CSMPGrupe. Sama CSMPGrupa je običan panel koji se sastoji iz dva dela: iz tela, imlementiranog pomoću promenljive *MyBody*, i iz naslova, datog kroz promenljivu *MyCaption*.

```
TCSMPGrupa = class(TPanel)
private
  FOnCaptionClick: TNotifyEvent;
  Klik: boolean;
  procedure Kliknuto(Sender: TObject);
  procedure SetOnCaptionClick(const Value: TNotifyEvent);

protected
  procedure WMErase(var Message: TMessage); message WM_ERASEBKGND;

public
  MyBody: TPanel;
  MyCaption: TCSMPDugme;

  constructor Create(AOwner: TComponent); override;
```

```
 published
   property OnCaptionClick: TNotifyEvent read FOnCaptionClick write
SetOnCaptionClick;

 end;
```

Promenljiva ***Klik*** je logička promenljiva koja je TRUE ukoliko je korisnik kliknuo mišem na naslov. Tada se poziva procedura ***Kliknuto*** koja poziva događaj ***OnCaptionClick*** i vlasnik CSMPGrupe zna šta se desilo. Samo telo je samo običan panel u koga se stavlja sav sadržaj koji se nalazi u grupi. Naslov je ***CSMPDugme*** čijim klikom će se grupa skupljati i širiti.

Ono što objedinjava i manipuliše svim grupama jeste komponenta nazvana CSMPGroupBar.

```
 TCSMPGroupBar = class(TPanel)
 private
   FOpenedHeight: Integer;
   FAktivnaGrupa: TCSMPGrupa;
   FSlike: TImageList;

   procedure SetAktivnaGrupa(const Value: TCSMPGrupa);
   procedure WMErase(var Message: TMessage); message WM_ERASEBKGND;
   procedure SetSlike(const Value: TImageList);

 protected
   procedure DodajGrupu(XMLGrupa: IXMLNodeList);
   procedure DodajElemente(XMLGrupa: IXMLNodeList; Grupa:TCSMPGrupa);

 public
   Grupe: TList;

   constructor Create(AOwner: TComponent); override;
   procedure Paint; override;
   procedure Resize; override;
   procedure CaptionClick(Sender: TObject);
   procedure AddGrupu(Grupa: TCSMPGrupa);
   procedure LoadFromXML(FileName: TFileName);

 published
   property AktivnaGrupa: TCSMPGrupa read FAktivnaGrupa write SetAktivnaGrupa;
   property Slike: TImageList read FSlike write SetSlike;
 end;
```

Private sekcija

FOpenedHeight: Integer

Kada korisnik klikne na naslov bilo koje grupe, ona se otvara, dok se sve ostale zatvaraju. Da bi se u svakom trenutku podesila veličina aktivne grupe, promenljiva FOpenedHeight čuva ovu veličinu, a ažurira se svaki put kada se veličina CSMPGroupBar-a ili njegovog roditelja promeni, ili kada se doda nova grupa.

Protected sekcija

procedure DodajGrupu(XMLGrupa: IXMLNodeList)

DodajGrupu dodaje novu grupu učitanu iz datoteke Manifest.xml, povezuje njene događaje i poziva proceduru DodajElemente.

procedure DodajElemente(XMLGrupa: IXMLNodeList; Grupa:TCSMPGrupa)

Ova procedura puni *Skroler* (objašnjeno u okviru unit-a *Skroler*) elementima (blokovima) iz xml datoteke i postavlja njihove vrednosti preko sopstvene procedure LoadFromXMLManifest.

Public sekcija

Grupe: TList

Ovo je lista grupa koje su ubačene u CSMPGroupBar.

procedure Resize; override

Obrađuje događaj promene veličine komponente. Ovo je bitno zbog promenljive FOpenedHeight.

procedure CaptionClick(Sender: TObject)

Menja aktivnu grupu kao odgovor na klik naslova neke grupe i tu grupu postavlja za aktivnu.

procedure AddGrupu(Grupa: TCSMPGrupa)

Dodaje tek napravljenu grupu u listu grupa.

procedure LoadFromXML(FileName: TFileName)

Procedura koja se prva izvršava. Učitava xml datoteku u kojoj se nalaze definisane sve grupe, i svi njihovi elementi – blokovi.

Published sekcija

property AktivnaGrupa: TCSMPGrupa read FAktivnaGrupa write SetAktivnaGrupa

AktivnaGrupa je grupa koja je trenutno otvorena. Menja se svaki put kada korisnik otvori neku drugu grupu.

property Slike: TImageList read FSlike write SetSlike

Ovo je lista slika koje se koriste kod iscrtavajna elemenata u grupi.

Opis Unit-a CSMPInspektor.pas

Pored ostalih, CSMPInspektor je nova komponenta u programu CSMP. Ona služi da korisniku prikazuje sve što je bitno vezano za trenutni selektovani blok. Omogućuje korisniku da lako promeni parametre, da u svakom trenutku vidi koje su njegove ulazne a koje izlasne veze, kao i da, klikom na bilo koji blok koji se nalazi u listi blokova na vrhu, selektuje blok po želji.

CSMPInspektor sadrži dve tabele koje pored osnovne funkcije imaju i još neke nestandardne. One su objekti klase TCSMPPolja.

```
TCSMPPolja = class(TStringGrid)
public
  constructor Create(AOwner: TComponent); override;
  procedure Resize; override;
  procedure Ocisti;
  procedure SelektovanaCelija(Sender: TObject; Col, Row: Longint; var CanSelect: Boolean);
end;
```

Procedura *Resize* podešava veličine obe kolone u tabeli, procedura *Ocisti* briše sav sadržaj iz tabele, dok *SelektovanaCelija* obezbeđuje da korisnik ne može da selektuje ćeliju kojoj nije dozvoljeno da bude promenjena.

Glavi događaji opisani su u okviru klase TCSMPInspektor.

```
 TCSMPInspektor = class(TPanel)
 private
   SelektujemSpolja:boolean;
   FCaption: TCaption;
   FAktivnaPovrsina: TCSMPRadnaPovrsina;

   procedure SetCaption(const Value: TCaption);
   procedure SetAktivnaPovrsina(const Value: TCSMPRadnaPovrsina);

 protected
   DBlokovi: TCSMPDugme;
   DElem: TCSMPDugme;
   DUlazi: TCSMPDugme;
   DIzlazi: TCSMPDugme;

   PBPanel: TPanel;
   PEPanel: TPanel;
   PUPanel: TPanel;
   PIPanel: TPanel;

   Splitter1: TCSMPSplit;
   Splitter2: TCSMPSplit;
   ListView1: TListView;

   AktivanBlok: TCSMPBlok;

   procedure LVSelektuj(Sender: TObject; Item: TListItem; Selected: Boolean);

 public
   SGUlazi: TCSMPPolja;
   SGIzlazi: TCSMPPolja;

   constructor Create(AOwner: TComponent); override;
   procedure Resize; override;

   procedure OsveziListu(Sender:TObject);
   procedure SelektujUListi(Sender:TObject);
   procedure Osvezi(Sender: TObject);
   procedure Popuni(Blok: TCSMPBlok);
   procedure Klik(Sender: TObject);
   procedure ProveraUnosa(Sender: TObject; ACol, ARow: Longint; const Value:
string);

 published
   property Caption: TCaption read FCaption write SetCaption;
```

property AktivnaPovrsina:TCSMPRadnaPovrsina read FAktivnaPovrsina write SetAktivnaPovrsina;
end;

Protected sekcija

DBlokovi: TCSMPDugme
DElem: TCSMPDugme
DUlazi: TCSMPDugme
DIzlazi: TCSMPDugme

Prethodna četiri CSMPDugmeta su, respektivno: dugme koje se nalazi iznad liste blokova; dugme koje je glavno za prikaz/skrivanje parametara i veza za selektovani blok; dugme koje je iznad tabele ulaza i parametara; dugme koje je iznad tabele izlaza. Sva četiri dugmeta imaju svrhu da skrivaju ili prikazuju ono što se nalazi neposredno ispod njih.

PBPanel: TPanel
PEPanel: TPanel
PUPanel: TPanel
PIPanel: TPanel

Ova četiri panela služe za lepši prikaz samog inspektora i za manipulaciju sa skrivanjem elemenata.

Splitter1: TCSMPSplit
Splitter2: TCSMPSplit

CSMPSpliteri koji omogućavaju da korisnik menja izgled inspektora po svojoj volji.

ListView1: TListView

Ovo je lista svih blokova koji se trenutno nalaze u modelu.

AktivanBlok: TCSMPBlok;

Blok koji je selektovan u radnoj površini i čiji se podaci trenutno prikazuju u inspektoru.

procedure LVSelektuj(Sender: TObject; Item: TListItem; Selected: Boolean)

Ova procedura selektuje blok na radnoj površini koji je kliknut u listi blokova.

Public sekcija

SGUlazi: TCSMPPolja
SGIzlazi: TCSMPPolja

Ovo su tabele u kojima se prikazuju ulazi i parametri trenutno aktivnog bloka, odnosno njegovi izlazi.

constructor Create(AOwner: TComponent); override

Najveća procedura u okvoru ove klase je *Create* u kojoj se postavljaju sve vrednosti i povezuju svi događaji za sve komponente u okviru inspektora i time stvara inspektor ovakav kakav je.

procedure Resize; override

Ova procedura podešava veličinu kolone za listu blokova svaki put kada se veličina samog inspektora menja.

procedure OsveziListu(Sender:TObject)

OsveziListu je procedura koja omogućava da se pri dodavanju novog bloka, kao i pri brisanju postojećeg iz radne površine osveži lista blokova i da u svakom trenutku prikazuje stvarno stanje na radnoj površini.

procedure SelektujUListi(Sender:TObject)

Ova procedura radi suprotno od procedure LVSelektuj, tj. obezbeđuje da inspektor uvek prikazuje podatke za selektovani blok.

procedure Osvezi(Sender: TObject);

Osvežava podatke o selektovanom bloku, tj. prikazuje podatke za aktivni blok.

procedure Popuni(Blok: TCSMPBlok);

Popunjava inspektor validnim podacima. Prolazi kroz sve elemente aktivnog bloka i njegova podatke upisuje u odgovarajuće tabele.

procedure Klik(Sender: TObject)

Ova procedura je povezana sa događajem *OnCaptionClick* svakog dugmeta u inspektoru i kada korisnik klikne mišem na dugme skriva panel koji je u CSMPDugmetu postavljen za aktivni.

procedure ProveraUnosa(Sender: TObject; ACol, ARow: Longint; const Value: string)

Da korisnik ne bi uneo pogrešne vrednosti za parametre u tabeli koja ih prikazuej, ova procedura proverava šta je uneto i ako je unet neodgovarajući znak prijavljuje korisniku poruku o nastaloj grešci.

Published sekcija

property Caption: TCaption read FCaption write SetCaption

Ovo svojstvo čuva naziv trenutno aktivnog bloka da bi on mogao da se prikazuje na dugmetu iznad dela za elemente svaki put kada korisnik klikne mišem na neki blok na radnoj površini.

property AktivnaPovrsina: TCSMPRadnaPovrsina

Da bi se znalo odakle će inspektor da uzima podatke koje treba da prikaže, pri kreiranju CSMPInspektora od strane programera popunjava se ovo svojstvo. Ako se *AktivnaPovrsina* ne postavi, CSMPInspektor neće da prikazuje nikakve podatke, jer nezna odakle da ih uzima.

Opis Unit-a CSMPSplit.pas

U okviru mnogih komponenti koje se koriste u programu korišćena je klasa TCSMPSplit. Ova komponenta je, u stvari, Splitter i služi za odvajanje i menjanje veličina raznih delova i komponenti u programu. Za razliku od komponente TSplitter koju nasleđuje, a koja je standardna komponenta Delfija, ova komponenta daje korisniku mogućnost da klikom miša na poseban pravougaonik, nazvan HotSpot, sakrije panel koji joj je dodeljen preko svojstva AktivniPanel i na taj način poveća prostor za prikazivanje ostalog na ekranu, a prvenstvena funkcija je da poveća površinu na kojoj se stvara model (ova površina je detaljno objašnjena u okviru objašnjenja za unit CSMPRadnaPovrsina).

U okviru ovog unit-a nalazi se poseban tip TSide:

TSide = (sdLeft, sdTop, sdRight, sdBottom)

TSide je nabrojivi tip koji se koristi pri iscrtavanju HotSpot-a. Koristi se u proceduri DrawSides koja se nalazi unutar procedure Paint, a služi za iscrtavanje strana HotSpot-a odvojeno jedna od druge, jer on može na taj način da se iscrta lepše nego korišćenjem već postojećih funkcija za iscrtavanje pravougaonika.

Sama klasa TCSMPSpliter data je na sledeći način:

```
TCSMPSplit = class (TSplitter)
private
  FVelicina: integer;
  FPrvo: boolean;
  FDole: boolean;
  FClick: boolean;
  FAktivniPanel: TCustomControl;
  FBojaTastera: TColor;
  FHotSpotRect: TRect;
  Obnovljeno: boolean;

  procedure SetAktivniPanel(const Value: TCustomControl);
  procedure SetBojaTastera(const Value: TColor);

protected
  procedure CMMouseLeave( var Msg: TMessage ); message cm_MouseLeave;

public
  HighLight: boolean;

  constructor Create(aOwner: TComponent); override;
  procedure Paint; override;
  procedure Resize; override;
  procedure MouseDown(Button: TMouseButton; Shift: TShiftState;
    X, Y: Integer); override;
  procedure MouseMove(Shift: TShiftState; X, Y: Integer); override;
  procedure MouseUp(Button: TMouseButton; Shift: TShiftState;
    X, Y: Integer); override;

published
  property AktivniPanel: TCustomControl read FAktivniPanel write SetAktivniPanel;
  property BojaTastera: TColor read FBojaTastera write SetBojaTastera;

end;
```

U nastavku slede detaljni opisi svih relevantnih funkcija i promenljivih dati po redosledu kako su navedeni u samoj klasi.

Private sekcija

FVelicina: integer

FVelicina je broj koji predstavlja veličinu HotSpot-a, tj. njegovu visinu (ako CSMPspliter „stoji" uspravno) ili širinu (ako CSMPspliter „stoji" položen) i sluzi da se u proceduri Paint odredi gde će se na komponenti iscrtavati HotSpot i koja će biti njegova velićina. Ovo je potrebno da se zna jer se veličina menja srazmerno sa promenom veličine cele komponente.

FPrvo: boolean

FPrvo je pomoćna promenljiva koja služi da se kod prvog pojavljivanja ove komponente pozove procedura Resize i podesi veličina HotSpot-a u zavisnosti od veličine samog CSMPSplitera i za sam rad i ponasanje komponente nije relevantna.

FDole: boolean

FDole označava da li je pritisnut taster miša. Potrebna je kod ispitivanja pomeranja miša, jer pomeranje miša sa pritisnutim tasterom znači da se komponenta, tj. spliter, pomera i time menja veličinu aktivnog panela (i komponente koja se nalazi u pravcu suprotnom od pravca strelica na HotSpot-u).

FClick: boolean

FClick označava da li je kliknut taster miša. Za razliku od promenljive FDole, FClick označava da je taster miša kliknut i da miš nije u međuvremenu pomeran. Tako se zna da li korisnik želi da klikne na HotSpot i time sakrije aktivni panel (ako je podešen), ili zeli da menja veličinu aktivnog panela.

FHotSpotRect: TRect

FHotSpotRect je pravougaonik koji predstavlja okvir za iscrtavanje HotSpot-a. Promenljiva je uvedena radi lakšeg i bržeg manipulisanja u proceduri Paint.

Obnovljeno: boolean

Pošto se pri svakom iscrtavanju komponente, a samim tim i HotSpot-a, proverava položaj miša i na osnovu njega menja ili ne boja HotSpot-a, što dovodi do ponovnog iscrtavanja, dolazi do "treperanja" cele komponente zbog ovih dodatnih nepotrebnih iscrtavanja. Promenljiva Obnovljeno služi da se HotSpot i ceo spliter dodatno iscrtava

samo onda kada je to potrebno. Detaljno objašnjenje kako je ovo implementirano dato je u objašnjenju procedure MouseMove.

Protected sekcija

procedure CMMouseLeave(var Msg: TMessage); message cm_MouseLeave

Procedura CMMouseLeave je procedura koja se poyiva od strane sistema u trenutku kada pokazivač miša „napusti pravougainok koji predstavlja okvir komponente. Pošto to znači i da miš sigurno ne pokazuje na HotSpot, promenljiva HighLight postaje FALSE i HotSpot se iscrtava bojom koja je postavljena preko svojstva BojaTastera.

Public sekcija

HighLight: boolean

HighLight promenljiva logičkog tipa se koristi kod iscrtavanja HotSpot-a u proceduri Paint i ima vrednost TRUE ukoliko se pokazivač miša nalazi iznad HotSpot-a, dok u suprotnom ima vrednost FALSE. Time se dobija utisak blagog „iskakanja" HotSpot-a svaki put kada korisnik postavi pokazivač miša iznad njega.

constructor Create(aOwner: TComponent); override

Create predstavlja konstruktor same klase i on se izvrsava pri samom kreiranju klase postavljajući vrednosti promenljivih na početne.

procedure Paint; override

Ova procedura se izvršava svaki put kada na ekranu dođe do neke promene. Ona daje kod koji služi da se sama komponenta iscrta na način na koji to njen stvaralac želi. O ovom slučaju njen najbitniji deo je iscrtavanje HotSpot-a. HotSpot se iscrtava na malo komplikovaniji način. Tj. svaki njegov deo se posebno iscrtava. Ovo je urađeno jer je iscrtavanje na taj način brže, a efekat je odličan. Prvo se iscrta okvir. Za iscrtavanje okvira se koristi procedura DrawSides. Ovoj se proceduri prosleđuju boje i stranice koje se žele da iscrtaju, kao i okvir i mesto gde one treba da se iscrtaju. Ona ih, zatim, jednu po jednu iscrtava postižući efekat ispupčenosti HotSpot-a. Posle iscrtavalja stranica se prelazi na crtanje strelica (trouglova) koje, u zavisnosti od toga na koju je stranu postavljen AktivniPanel, pokazuju na njega. Na kraju se iscrtavaju tačkice između prethodno iscrtanih strelica.

procedure Resize; override

Procedura Resize se poziva svaki put kada komponenta promeni svoju veličinu. U ovom slučaju to je bitno jer se u njoj podešavu promenljive FVelicina i FhotSpotRect koje su zadužene za regulisanje veličine i položaja HotSpot-a.

procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y: Integer); override

Svaki put kada korisnik klikne neko dugme miša, pozove se ova procedura. Ovaj događaj je bitan jer pritisak tastera miša oznacava da korisnik želi da nešto uradi sa komponentom. Ako je miš iznad HotSpot-a promenljiva FClick dobija vrednost TRUE i FDole takođe označavajući trenutni položaj tastera miša.

procedure MouseMove(Shift: TShiftState; X, Y: Integer); override

Kada se miš kreće u okviru komponente, procedura MouseMove se izvršava. Ona djae informaciju šta u tom slučaju treba raditi. Ako je miš iznad HotSpot-a, promenljiva HighLight to beleži, a kursor miša se menja. Ukoliko je prethodno pritisnut taster miša, potrebno je pomerati celu komponentu. U oba slučaja zna se sigurno da korisnik ne želi da samo klikne na HotSpot tako da se promenljiva FClick ažutira vrednošću FALSE.

procedure MouseUp(Button: TMouseButton; Shift: TShiftState; X, Y: Integer); override

Kada je, na kraju, korisnik opustio taster miša potrebno je da se proveri stanje promenljive FClick i, ukoliko je ona TRUE i postoji aktivni panel, mora se on sakriti ili otkriti u zavisnosti od prethodnog stanja. Na samom kraju obavezno je postavljanje vrednosti promenljivih FClick i FDole na FALSE.

Published sekcija

property AktivniPanel: TCustomControl read FAktivniPanel write SetAktivniPanel

AktivniPanel je opcion atribut i ako je postavljen od strane korisnika onda on čuva referencu na panel koji se da se sakriva i otkriva kada korisnik klikne mišem na HotSpot. Ako on nije postavljen onda HotSpot nema svoju primarnu ulogu, ali se i dalje iscrtava u zavisnosti od položaja miša.

property BojaTastera: TColor read FBojaTastera write SetBojaTastera

FBojaTastera je boja kojom se iscrtava HotSpot ukoliko se na njemu ne nalazi pokazivač miša. U suprotnom, on se iscrtava bojom datom u proceduri Paint.

S obzirom na to da ovaj unit nema globalne promenljive poslednjim objašnjenjem objašnjen je unit CSMPSplit.pas u celosti.

Opis Unit-a OpcijeSim.pas

Kada se, u okviru glavnog unit-a programa CSMP, klikne na taster kojim se započinje simulacija, i, u okviru unit-a Obrada, izvrši logička kontrola, otvara se prozor u kome je potrebno uneti potrebne podatke za obavljanje same simulacije. U okviru ovog unit-a data je implementacija tog prozora sa svojim promenljivama i funkcijama.

Klasa TOpcije je ništa drugo do običan Windows® prozor kome koji je raznim podešavanjima promenjen izgled (ovo je isto urađeno za sve prozore koji se pojavljuju u okviru programa, tako da će se način kako je ovo urađeno opisati samo ovde).

Sama klasa data je kroz sledeći kod:

```
TOpcije = class(TForm)
  Image1: TImage;
  Image2: TImage;
  Panel1: TPanel;
  Panel2: TPanel;
  SpeedButton1: TSpeedButton;
  SpeedButton2: TSpeedButton;
  Edit1: TEdit;
  Edit2: TEdit;
  Edit3: TEdit;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Group: TGroupBox;

  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton2Click(Sender: TObject);

 private
  procedure WMNCHITTEST(var msg:TMessage);message WM_NCHITTEST;
  procedure WMERASE(var msg:TMessage);message WM_ERASEBKGND;

 public
  Ok: boolean;
  DuzInt: real;
  IntInt: real;
```

```
  IntStamp: real;
end;
```

Komponente na samom prozoru:

Image1: TImage

Image1 je slika koja se nalazi iza osnovnih komponenti i vizuelno predstavlja okvir prozora.

Image2: TImage

Image2 je slika koja simulira tzv. Caption svakog prozora, tj. deo na vrhu samog prozora koji sadrži naslov i sistemske ikonice. Ove ikonice ovde nisu potrebne tako da njihovo pojavljivanje nije implementirano.

Panel1: TPanel

Panel1 je „podloga" na kojoj stoje ostale komponente, a koja se nalazi unutar „okvira" prozora.

Panel2: TPanel

Panel2 je podloga i okvir u kome se nalaze dva tastera za prihvatanje ili odbijanje unetih podataka.

SpeedButton1: TSpeedButton

SpeedButton1 je taster za prihvatanje unetih vrednosti za opcije za simulaciju. Klikom na njega poziva se procedura SpedButton1Click.

SpeedButton2: TSpeedButton

SpeedButton2 je taster čijim klikom se odbija dalje izvršenje simulacije i vraća na model.

Edit1: TEdit

Edit1 je prostor za unošenje vrednosti za dužinu simulacije.

Edit2: TEdit

Edit1 je prostor za unošenje vrednosti za interval integracije.

Edit3: TEdit

Edit1 je prostor za unošenje vrednosti za interval štampanja rezultata.

Label1: TLabel
Label2: TLabel
Label3: TLabel

Ove komponente prikazuju običan tekst na ekranu u zavisnosti ispred kog polja za unos podataka se nalaze.

Label4: TLabel

Ovo je tekst koji predstavlja naslov prozora i nalazi se na samom vrhu prozora preko slike Image2. Podešavanjem svojstva *Transparent* na TRUE dobio se efekat providnosti teksta.

Group: TGroupBox

Group je komponenta koja pravi okvir oko polja za unos podataka i uz to prikazuje tekst *Trajanje simulacije* u svom naslovu.

procedure SpeedButton1Click(Sender: TObject;

Kada korisnik klikne na dugme za prihvatanje rezultata poziva se ova procedura koja je zadužena da izvrši logičku kontrolu unetih podataka. Ukoliko je korisnik uneo nedozvoljene podatke prikazuje se poruka o grešci i od korisnika se traži da pogrešne podatke zameni. Ako su svi podaci pravilno uneti promenljiva Ok dobija vrednost TRUE, sve javne promenljive dobijaju odgovarajuće vrednosti i prozor se zatvara.

procedure SpeedButton2Click(Sender: TObject)

Ako korisnik želi da odustane od simulacije kliknuće na taster SpeedButon2 i pozvaće se ova procedura u kojoj promenljiva Ok dobija vrednost FALSE nakon čega se prozor zatvara.

Private sekcija

procedure WMNCHITTEST(var msg:TMessage);message WM_NCHITTEST;

Pošto je svojstvo *BorderStyle* samog prozora postavljeno na *bsNone* naslov prozora je zamenjen slikom Image2. Da bi ipak prozor mogao da se pomera „hvatanjem” za naslov kao o svaki drugi običan prozor, obavezno je da se u proceduri ***WMNCHITTEST*** naznači koji je to region prozora koji će da služi da, kada se klikne na

njega, prozor može da se pomera. Inače, ova procedura se poziva svaki put kada korisnik klikne mišem bilo gde u prozoru.

Public sekcija

Ok: boolean

Ako je sve proteklo u redu kod unosa podataka za simulaciju i simulacija treba da se izvrši, ova promenljiva će biti TRUE. U suprotnom, njena vrednost će se postviti na FALSE.

DuzInt: real
IntInt: real
IntStamp: real

Ove promenljive čuvaju vrednosti koje je korisnik uneo za dužinu simulacije, interval integracije i dužinu intervala štampanja, respektivno.

Opis Unit-a uUnosParam.pas

TUnosPar je još jedan od prozora koji se javljaju u programu. On se javlja kada korisnik spusti na radnu površinu nov blok koji u sebi ima parametre. U ovom prozoru se ti parametri unose i pamte u bloku.

```
TUnosPar = class(TForm)
  Image1: TImage;
  Image2: TImage;
  CaptionLabel: TLabel;
  Panel1: TPanel;
  Panel2: TPanel;
  GroupBox1: TGroupBox;
  GroupBox2: TGroupBox;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Edit1: TEdit;
  Edit2: TEdit;
  Edit3: TEdit;
  SpeedButton1: TSpeedButton;
  SpeedButton2: TSpeedButton;
```

```
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton2Click(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure Panel7Click(Sender: TObject);

private
  procedure WMNCHITTEST(var msg:TMessage);message WM_NCHITTEST;
  procedure WMERASE(var msg:TMessage);message WM_ERASEBKGND;

public
  Ok: boolean;

end;
```

Skoro sve komponente na ovom prozoru su istovetne sa komponentama opisanim u unit-u OpcijeSim.pas i zbog toga se njihov opis ovde neće ponavljati.

Elementi koji su ovde posebni jesu samo mali panel u gornjem levom uglu okvira na kome piše **Napomena** i čijim klikom se prikazuje poruka o redosledu unosa parametara data u proceduri Panel2Click, kao i procedura FormShow koja se izvršava svaki put kada se prozor pojavi i u kojoj se vrednosti za paametre postavljaju na početnu vrednost. Ako korisnik klikne na SpeedButton2 na kome piše „**odustani”**, onda će se smatrati da korisnik ne želi da postavi spušteni blok i on će se izbrisati.

Opis Unit-a ulzborUlaza.pas

U ovom unit-u se nalazi opis za prozor koji korisniku nudi da za trenutno napravljenu vezu odredi na koj ulaz u blok će se ona zakačiti. S obzirom na to da je ovaj prozor skoro isti kaoprozor za unos parametara, njegov opis čitalac ovog teksta može da pogleda u opisu prethodno navedenog prozora.

```
TIzborUlaza = class(TForm)
  Image1: TImage;
  Image2: TImage;
  Label1: TLabel;
  Panel1: TPanel;
  GroupBox1: TGroupBox;
  RadioButton1: TRadioButton;
  RadioButton2: TRadioButton;
  RadioButton3: TRadioButton;
  GroupBox2: TGroupBox;
  SpeedButton1: TSpeedButton;
  SpeedButton2: TSpeedButton;
  Panel2: TPanel;
```

```
  procedure SpeedButton1Click(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure SpeedButton2Click(Sender: TObject);
  procedure Panel3Click(Sender: TObject);

 private
  procedure WMNCHITTEST(var msg:TMessage);message WM_NCHITTEST;
  procedure WMERASE(var msg:TMessage);message WM_ERASEBKGND;

 public
  Ok: boolean;

 end;
```

Opis Unit-a uUnosOpisa.pas

Unit uUnosOpisa sadrži prozor koji je nešto novo u odnosu na prethodne verzije programa. Sada korisnik može da, pored modela kojeg stvara, napiše i opis samog problema, belešku autora modela ili bilo šta drugo što smatra da može da ide uz napravljeni model. Izborom opcije **Edit|Opis problema** pojavljuje se ovaj prozor gde korisnik unosi svoje beleške. Promenljiva *Tekst* čuva sve što je uneto.

```
TUnosOpisa = class(TForm)
  Image1: TImage;
  Image2: TImage;
  Memo1: TMemo;
  Panel1: TPanel;
  Label1: TLabel;
  SpeedButton1: TSpeedButton;
  SpeedButton2: TSpeedButton;
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton2Click(Sender: TObject);

 private
  procedure WMNCHITTEST(var msg:TMessage);message WM_NCHITTEST;
  procedure WMERASE(var msg:TMessage);message WM_ERASEBKGND;

 public
  Tekst: String;

 end;
```