

## Minimalistisen ohjelmatulkin ohjelmointi käyttäen puolalaista notaatiota.

Puolalainen notaatio(PN) eli prefix-notaatio on rekursiivinen matemaattisten kaavojen esitysmuoto, joka mahdollistaa yksinkertaisen ohjelmatulkin teon. PN:ssä operaattorit ja funktiot esitetään aina ennen argumenttejaan. Sulkuja ei tarvita, eikä operaattoreilla ole etuajo-oikeuksia.

Muuttujamäärittelyssä muiden notaatioiden ongelmat ovat marginaalisia puolalaisessa notaatiossa. Samannimisiä muuttujia voi olla eri tyypeissä vapaasti. Mahdolliset ongelmat on helppo huomioida etukäteen. PN mahdollistaa eri tyyppisiin objekteihin sekaisin perustuvan syntaksin. Esimerkiksi pisteen määrittelyssä voidaan yhdistää pisteitä ja arvoja. Myös viittaus mutkikkaimpien objektien osiin on mahdollista.

Nelilaskin voidaan ohjelmoida Pascal-kielellä näin lyhyesti:

```
const isoluku = 1e300 ;
function SamaTeksti(Str1,Str2 : string) : boolean ;
begin
    if Str1=Str2 then SamaTeksti := true else SamaTeksti := false ;
end ;

function pop(var Str1 : string) : string ;
var ik,nk : integer ;
begin
    ik := Pos(' ',Str1) ;
    nk := length(Str1) ;
    if ik>0 then
    begin
        pop := copy(Str1,1,ik-1) ;
        if ik=nk then Str1 := '' else Str1 := copy(Str1,ik+1,nk-ik) ;
    end
    else
    begin
        pop := Str1 ;
        Str1 := '' ;
    end ;
end ;

function arvo(var lause : string) : real ;
var Str1 : string ;
    arv2,arv3 : real ;
begin
    arvo := 0 ;
    Str1 := pop(lause) ;
    if (SamaTeksti(Str1,'+') or SamaTeksti(Str1,'-')) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if SamaTeksti(Str1,'+') then arvo := arv2+arv3 else arvo := arv2-arv3;
    end
    else if (SamaTeksti(Str1,'*') or SamaTeksti(Str1,'/')) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if SamaTeksti(Str1,'*') then arvo := arv2*arv3 else
        if arv3<>0 then arvo := arv2/arv3 else
        if arv2<>0 then arvo := arv2/abs(arv2)*isoluku else arvo := 1 ;
    end
    else arvo := StrToInt(Str1) ;
end ;
```

'SamaTeksti'-functiossa verrataan vain tekstien samankaltaisuutta.

'pop'-functiossa otetaan tekstimuuttujan alusta teksti joka erottuu tyhjän lyönnin avulla. (blankko erottelu).

Kokonaislukuilla pelaaminen riittää, koska desimaaliluvut voidaan korvata lausekkeilla:

+ 2 / 23 1000 joka on sama kuin 2,023 tai voidaan määritellä funktio de3 2 23 ajaen saman asian.

Nelilaskin voidaan laajentaa funktiolaskimeksi lisäämällä esimerkiksi ohjelmakoodia:

```
else if SamaTeksti(Str1,':') then
begin
  arv2 := arvo(lause) ;
  arv3 := arvo(lause) ;
  arvo := ArcTan2(arv2,arv3)*45/ArcTan(1) ;
end
else if SamaTeksti(Str1,'atan') then
arvo := ArcTan(arvo(lause))*45/ArcTan(1)
else if SamaTeksti(Str1,'sin') then
arvo := Sin(arvo(lause)/45*ArcTan(1))
```

Muuttujien arvojen sijoitus vaatii vektorien varausta, tulkin käyttöä ja sijoituskomennon:

```
temuut,tearvo,telaus : array[0..tasoja,1..sataa] of string ;
muuttu,mulaus,piste,pilaus : array[0..tasoja,1..sataa] of string ;
muarvo,pisx,pisy,pisx1,pisy1 : array[0..tasoja,0..sataa] of real ;

procedure tulkinta(var lause : string) ;
var Str1,Str2,Str3,Str4,Str5 : string ;
  ik,ii,nk,nn : integer ;
  ok : boolean ;
  arv1,arv2,arv3,arv4,arv5,arv6,arv7 : real ;
  si1,si2,col,co2,r1,r2,det : real ;
begin
  Str1 := pop(lause) ;
  if SamaTeksti(Str1,'laita') then
  begin
    Str2 := pop(lause) ;
    Str3 := lause ;
    arv2 := arvo(lause) ;
    if round(muarvo[taso,0])>0 then
    begin
      ok := false ;
      for ik := 1 to round(muarvo[taso,0]) do
      if SamaTeksti(Str2,muuttu[taso,ik]) then
      begin
        ok := true ;
        muuttu[taso,ik] := Str2 ;
        mulaus[taso,ik] := Str3 ;
        muarvo[taso,ik] := arv2 ;
      end ;
    end ;
    if not ok then
    begin
      nk := round(muarvo[taso,0]) ;
      if nk<sataa then
      muarvo[taso,0] := muarvo[taso,0]+1 ;
```

```

        ik := round(muarvo[taso,0]) ;
        muuttu[taso,ik] := Str2 ;
        mulous[taso,ik] := Str3 ;
        muarvo[taso,ik] := arv2 ;
    end ;
end
else
begin
    muarvo[taso,0] := 1 ;
    muuttu[taso,1] := Str2 ;
    mulous[taso,1] := Str3 ;
    muarvo[taso,1] := arv2 ;
end ;
end ;
end ;

```

Muuttujien arvojen käyttö voidaan toteuttaa lisäämällä ohjelmakoodia arvo-funktioon seuraavasti:

```

else if round(muarvo[taso,0])>0 then
begin
    ok := false ;
    for ik := 1 to round(muarvo[taso,0]) do
        if SamaTeksti(Str1,muuttu[taso,ik]) then
            begin
                ok := true ;
                arvo := muarvo[taso,ik] ;
            end ;
            if not ok then arvo := StrToInt(Str1) ;
        end
    end
else
    arvo := StrToInt(Str1) ;
end

```

Ohjelmatulkin kokonaisrakenne tulee seuraavan mukaiseksi:

```

procedure TForm1.FormClick(Sender: TObject);
const sataa = 400 ;
      tasoja = 3 ;
      isoluku = 1e300 ;
var ipa,jpa,kpa,npa,m,nmu,nte,taso : integer ;
...

function arvo(var lause : string) : real ;
begin
end ;

function teksti(var lause : string) : string ;
begin
end ;

procedure pistos(var lause : string;var x,y : real) ;
begin
end ;

procedure janat(var lause:string;var x,y,x2,y2,vp:real;var vari:integer) ;
begin

```

```

end ;

procedure tulkinta(var lause : string) ;
begin
end ;

procedure rivit(Listaus : TStringList) ;
begin
end ;

procedure tulkit(Listaus : TStringList) ;
begin
end ;

begin
  Lista := TStringList.Create ;
  MuuLista := TStringList.Create ;
  nimi := 'muuttujat.txt' ;
  if FileExists(nimi) then MuuLista.LoadFromFile(nimi) ;
  nimi := 'lauseet.txt' ;
  if FileExists(nimi) then Lista.LoadFromFile(nimi) ;
  if MuuLista.Count>0 then
  begin
    m := MuuLista.Count ;
    rivit(MuuLista) ;
    tulkit(MuuLista) ;
  end ;
  if Lista.Count>0 then
  begin
    m := Lista.Count ;
    rivit(Lista) ;
    tulkit(Lista) ;
    Kuvasvg.Clear ;
    svgotsa(4) ;
    for ipa := 0 to tasoja do
    begin
      taso := ipa ;
      piirra ;
    end ;
    svghanta ;
  end ;
end ;

```

Ohjelma on sijoitettu lomakkeen Click-tapahtumankäsittelijään.

Tulkit-proceduuri käy lävitse ohjelmarivit välittäen ne tarpeen mukaan tulkinta-proceduurille. Tätä ennen käydään lävitse ehdollisten hyppykäskyjen mahdolliset osoitteet rivit-proceduurilla. Tulkinta-proceduuri kutsuu tarpeen mukaan janat-proceduuria, pistos-proceduuria, teksti-functiota tai arvo-functiota. Janat-proceduuri kutsuu taas pistos-proceduuria tai arvo-functiota. Pistos-proceduuri kutsuu arvo-functiota tai itseään. Arvo-functio voi kutsua myös itseään samoin kuin teksti-functio.

Ohessa tulkit-proceduurin ohjelmakoodi:

```

procedure tulkit(Listaus : TStringList) ;
var Str1,Str2,Str3,Str4 : string ;

```

```

    ji,nk,isohko,loo,tassi : integer ;
    ok : boolean ;
    arv1,arv2 : real ;
begin
    nk := Listaus.Count ;
    isohko := 100000 ;
    loo := 0 ;
    ji := 0 ;
    repeat
        Str4 := Listaus[ji] ;
        Str2 := Str4 ;
        Str4 := poisalku(Str4) ;
        Str3 := Str4 ;
        Str1 := pop(Str3) ;
        if SamaTeksti(Str1,'jos') then
            begin
                arv1 := arvo(Str3) ;
                arv2 := arvo(Str3) ;
                if arv1>0 then
                    ji := round(arv2)
                else ji := ji + 1 ;
            end
        else if SamaTeksti(Str1,'taso') then
            begin
                tassi := round(arvo(Str3)) ;
                taso := tassi ;
                ji := ji + 1 ;
            end
        else if SamaTeksti(Str1,'peri') then
            begin
                arv2 := arvo(Str3) ;
                ita := round(arv2) ;
                if ((ita>-1) and (ita<tasoja+1)) then
                    begin
                        nkt := Perinta[ita].Count ;
                        if nkt>0 then
                            for ipa := 0 to nkt-1 do
                                begin
                                    Str4 := Perinta[ita][ipa] ;
                                    tulkinta(Str4) ;
                                    loo := loo + 1 ;
                                end ;
                            end ;
                        ji := ji + 1 ;
                    end
                else if SamaTeksti(Str1,'tulk') then
                    begin
                        Str2 := teksti(Str3) ;
                        tulkinta(Str2) ;
                        ji := ji + 1 ;
                    end
                else
                    begin
                        if SamaTeksti(Str1,'--') then
                            begin
                                end
                            else
                                begin
                                    try
                                        tulkinta(Str4) ;
                                    except
                                        VirheLista.Append('Virhe: '+Str2) ;
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

```

```

        end ;
        ji := ji + 1 ;
    end ;
end ;
loo := loo + 1 ;
until ((ji = nk) or (loo>isohko)) ;
end ;

```

Tulkit-proceduurissa käydään läpi ohjelmarivit. 'isohko'- ja 'loo'-muuttuja ovat ohjelmointivirheiden varalle määriteltyjä arvoja. Jos ohjelmassa toistetaan samoja rivejä isohkoa enemmän, niin ollaan jossakin kohtaa harhateillä. Seuraavaksi ohjelmakoodia tutkitaan järjestelmällisesti repeat(toista)-komennon avulla. Ensimmäisenä tarkistetaan, onko komento ehdollinen hyppykäskey. Käskey 'jos' tutkii, onko annettu lauseke tosi vai vale. Totuus ansaitsee hypyn seuraavana olevaan rivinumeroon. Seuraavana oleva 'taso' tarkoittaa piirtotason vaihtoa. 'peri' taas viittaa uuden tason perintään vanhalta tasolta. Perintä tarkoittaa kaikkien muuttujien siirtoa arvoineen uudelle tasolle. 'tulk' käskää lähettämään seuraavana olevan tekstin suoraan tulkintaan. Jokaisen komennon kohdalla huolehditaan siirtymisestä seuraavalle tai hyppykäskyn määräämälle riville. Pelkkä kommentti ohitetaan. Lopuksi yritetään vain tulkita riviä. Muistetaan myös vaihtaa riviä.

Seuraavassa rivit-proceduurin ohjelmakoodi:

```

procedure rivit(Listaus : TStringList) ;
var Str1,Str2,Str3,Str4 : string ;
    ik,nk : integer ;
    ok : boolean ;
    arv1,arv2 : real ;
begin
    nk := Listaus.Count ;
    for ik := 0 to nk-1 do
    begin
        Str4 := Listaus[ik] ;
        Str4 := poisalku(Str4) ;
        Str1 := pop(Str4) ;
        SamaTeksti(Str1,'rivi') then
        begin
            Str2 := pop(Str4) ;
            Str3 := StrLet+' '+Str2+' '+IntToStr(ik) ;
            tulkinta(Str3) ;
        end ;
    end ;
end ;
end ;

```

Tämän ohjelmanpätkän tarkoitus on tutkia kohdat, mihin ehdollisella hyppykäskyllä hypätään. 'rivi'-komennon kertomaan muuttujaan sijoitetaan senhetkinen rivinnumero.

Ja janat-proceduurin ohjelmakoodi:

```

procedure janat(var lause:string;var x,y,x2,y2,vp:real;var vari:integer) ;
var dump : real ;
begin
    pistos(lause,x,y) ;
    pistos(lause,x2,y2) ;
    if length(lause)>0 then vp := arvo(lause) else vp := 1 ;
    if length(lause)>0 then vari := round(arvo(lause)) else vari := 0 ;
end ;

```

Ensin tutkitaan janan pisteiden arvot sijoittaen koordinaatit muuttujiin x ja y sekä x2 ja y2. Tarvittaessa etsitään arvot viivanpaksuudelle arvo-functiolla ja värille pyöristämällä arvo-function antama tulos kokonaisluvuksi.

Pistos-proceduurin ohjelmakoodi:

```
procedure pistos(var lause : string;var x,y : real) ;
var Str1,Str2,Str3,Str4,Str5 : string ;
    ik,nk : integer ;
    ok : boolean ;
    arv1,arv2,arv3,arv4,arv5,arv6,arv7 : real ;
    si1,si2,co1,co2,r1,r2,det : real ;
begin
    Str2 := lause ;
    Str1 := pop(lause) ;
    ik := length(Str1) ;
    if SamaTeksti(Str1,'psx') then
    begin
        pistos(lause,arv2,arv3) ;
        arv5 := arvo(lause)*arctan(1)/45 ;
        arv1 := arvo(lause) ;
        x := arv1 ;
        if cos(arv5)<>0 then
            y := arv3 - (arv2 - x) * sin(arv5) / cos(arv5)
        else
            y := arv3 ;
        end
    end
    else if round(pisx[taso,0])>0 then
    begin
        ok := false ;
        for ik := 1 to round(pisx[taso,0]) do
            if SamaTeksti(Str1,piste[taso,ik]) then
            begin
                ok := true ;
                x := pisx[taso,ik] ;
                y := pisy[taso,ik] ;
            end ;
            if not ok then
            begin
                lause := Str2 ;
                x := arvo(lause) ;
                y := arvo(lause) ;
            end ;
        end
    end
    else
    begin
        lause := Str2 ;
        x := arvo(lause) ;
        y := arvo(lause) ;
    end ;
end ;
```

Pisteen määrittely voidaan tehdä useilla tavoilla. Esimerkissä on 'psx' -määrittely on

<piste> <suunta> <x-arvo>

Muita vaihtoehtoja ovat esimerkiksi 'psy', 'psp' ja 'psps'.

<piste> <suunta> <y-arvo>

<piste> <suunta> <pituus>

<piste> <suunta> <piste> <suunta>

Piste-muuttuja voi myös olla vaihtoehto. Tietysti pelkät koordinaatitkin riittävät.

Teksti-function ohjelmakoodi:

```
function teksti(var lause : string) : string ;
var Str1,Str2,Str3,Str4,Str5 : string ;
    ik,nk : integer ;
    ok : boolean ;
    arv1,arv2,arv3,arv4,arv5,arv6,arv7 : real ;
    si1,si2,co1,co2,r1,r2,det : real ;
begin
    Str2 := lause ;
    Str1 := pop(lause) ;
    ik := length(Str1) ;
    if SamaTeksti(Str1,'te+') then
    begin
        Str4 := teksti(lause) ;
        Str5 := teksti(lause) ;
        teksti := Str4+Str5 ;
    end
    else if SamaTeksti(Str1,'tes') then
    begin
        arv1 := arvo(lause) ;
        nk := round(arvo(lause)) ;
        ik := round(arvo(lause)) ;
        Str3 := ReaaliTex(arv1,nk,ik) ;
        teksti := Str3 ;
    end
    else if SamaTeksti(Str1,'istr') then
    begin
        arv1 := arvo(lause) ;
        Str3 := IntToStr(round(arv1)) ;
        teksti := Str3 ;
    end
    else if round(pisy[taso,0])>0 then
    begin
        ok := false ;
        for ik := 1 to round(pisy[taso,0]) do
            if SamaTeksti(Str1,temuut[taso,ik]) then
            begin
                ok := true ;
                teksti := tearvo[taso,ik] ;
            end ;
        end ;
    end ;
```



```

        if not ok then teksti := Str2 ;
    end
    else teksti := Str2 ;
end ;

```

Koodissa etsitään tekstien yhdistämistä 'te+' -komennolla. 'tes' ja 'istr' muotoilevat luvut tekstiksi. Teksti-muuttujien arvoja haetaan tietysti lopuksi.

Teksti-muuttujien arvojen käyttö voidaan toteuttaa lisäämällä ohjelmakoodia tulkinta-poceduuriin seuraavasti:

```

else if SamaTeksti(Str1,'tex') then
begin
    Str2 := pop(lause) ;
    Str3 := lause ;
    Str4 := teksti(lause) ;
    if round(pisy[taso,0])>0 then
    begin
        ok := false ;
        for ik := 1 to round(pisy[taso,0]) do
            if SamaTeksti(Str2,temuut[taso,ik]) then
            begin
                ok := true ;
                temuut[taso,ik] := Str2 ;
                telaus[taso,ik] := Str3 ;
                tearvo[taso,ik] := Str4 ;
            end ;
            if not ok then
            begin
                nk := round(pisy[taso,0]) ;
                if nk<sataa then
                pisy[taso,0] := pisy[taso,0]+1 ;
                ik := round(pisy[taso,0]) ;
                temuut[taso,ik] := Str2 ;
                telaus[taso,ik] := Str3 ;
                tearvo[taso,ik] := Str4 ;
            end ;
        end ;
    end ;
end
else if SamaTeksti(Str1,'sana') then
begin
    Str2 := pop(lause) ;
    Str3 := lause ;
    Str4 := pop(lause) ;
    pistos(lause,arv2,arv3) ;
    arv1 := arvo(lause) ;
    arv4 := 0 ;
    arv5 := 0 ;
    if length(lause)>0 then
    arv4 := arvo(lause) ;
    if length(lause)>0 then
    arv5 := arvo(lause) ;
    Str5 := teksti(Str4) ;
    if round(tepix[taso,0])>0 then
    begin
        ii := round(tepix[taso,0])+1 ;
        for ik := 1 to round(tepix[taso,0]) do

```

```

    if SamaTeksti(Str2,tepi[taso,ik]) then
        ii := ik ;
        tepix[taso,0] := ii ;
        tepilaus[taso,ii] := Str3 ;
        tepix[taso,ii] := arv2 ;
        tepiy[taso,ii] := arv3 ;
        tepiv[taso,ii] := round(arv4) ;
        tepiko[taso,ii] := arv1 ;
        tepiku[taso,ii] := arv5 ;
        tepi[taso,ii] := Str2 ;
        tepite[taso,ii] := Str5 ;
    end
else
begin
    ii := 1 ;
    tepix[taso,0] := ii ;
    tepilaus[taso,ii] := Str3 ;
    tepix[taso,ii] := arv2 ;
    tepiy[taso,ii] := arv3 ;
    tepiv[taso,ii] := round(arv4) ;
    tepiko[taso,ii] := arv1 ;
    tepiku[taso,ii] := arv5 ;
    tepi[taso,ii] := Str2 ;
    tepite[taso,ii] := Str5 ;
end ;
end
else if SamaTeksti(Str1,'teksti') then
begin
    Str2 := pop(lause) ;
    Str3 := lause ;
    if round(pisy[taso,0])>0 then
begin
    ok := false ;
    for ik := 1 to round(pisy[taso,0]) do
        if SamaTeksti(Str2,temuut[taso,ik]) then
begin
            ok := true ;
            temuut[taso,ik] := Str2 ;
            telaus[taso,ik] := Str3 ;
            tearvo[taso,ik] := Str3 ;
        end ;
        if not ok then
begin
            nk := round(pisy[taso,0]) ;
            if nk<sataa then
                pisy[taso,0] := pisy[taso,0]+1 ;
            ik := round(pisy[taso,0]) ;
            temuut[taso,ik] := Str2 ;
            telaus[taso,ik] := Str3 ;
            tearvo[taso,ik] := Str3 ;
        end ;
    end ;
end
else
begin
    pisy[taso,0] := 1 ;
    temuut[taso,1] := Str2 ;
    telaus[taso,1] := Str3 ;
    tearvo[taso,1] := Str3 ;
end
end ;
end ;

```

Koodissa etsitään ensin tekstien sijoittamista 'tex' -komennolla. Tekstimuuttujaan sijoitetaan lausekkeita, muuttujia tai lopuksi tekstiä.

Seuraavaksi 'sana'-komento luo piirtoon haluttuun asentoon, kokoon ja väriin tulevan tekstin.

Kyseessä on tekstiobjekti, joka on hankala ohjelmoida Wintoosassa. Muissa alustoissa ja tiedostoformaateissa ovat omat ongelmansa.

Viimein 'teksti'-komento sijoittaa muuttujaan kirjoitetun tekstin. Tekstien käsittelyssä ei käytetä hipsuja. Tämä yksinkertaistaa ohjelmakoodia ja auttaa sisällyttämään kaikki merkit tekstimuuttujien arvoihin. Tämä muuttujien käsittelytapa mahdollistaa eri tyyppisten muuttujien nimeämisen samoihin nimiin. On kuitenkin tilanteita, joissa syntyy ristiriitoja syntaksissa. Esimerkiksi pisteen määrittelyssä voi tulla ristiriita muuttujan <piste> ja muuttujan <x-arvo> kesken. Asia voidaan huomioida käyttämällä <x-arvo> tilanteessa lauseketta '+ 0 <x-arvo>'.

Seuraavaksi tarkka syntaksi:

```
<BNF> ::= Backus-Naurin-muoto

BNF-esitys kielestä:

<menokohde> ::= rivi <muuttuja>

<hyppykäskey> ::= jos <arvo> <rivinro>

<tekstin tulkinta> ::= tulk <teksti>

<tason valinta> ::= taso <arvo>

<tason perintä> ::= peri <arvo>

<rivinro> ::= <arvo>

<kommentti> ::= -- <merkkejä>

<janasijoitus> ::= jana <janamuuttuja> <piste> <piste> <viivan paksuus> <väri> |
                    jana <janamuuttuja> <piste> <piste> <viivan paksuus>

<pistesijoitus> ::= piste <pistemuuttuja> <piste>

<muuttujasijoitus> ::= laita <muuttuja> <arvo>

<piste> ::= <x-arvo> <y-arvo> |
            <pistemuuttuja> |
            psx <piste> <suunta> <x-arvo> |
            psy <piste> <suunta> <y-arvo> |
            psp <piste> <suunta> <piste> <suunta> |
            psp <piste> <suunta> <pituus> |
            p1 <janamuuttuja> |
            p2 <janamuuttuja>

<x-arvo> ::= <arvo> |
            x <pistemuuttuja>

<y-arvo> ::= <arvo> |
            y <pistemuuttuja>

<suunta> ::= <arvo> |
            suunta <piste> <piste>

<pituus> ::= <arvo> |
            ero <piste> <piste>

<viivan paksuus> ::= <arvo> |
                    paks <janamuuttuja>

<väri> ::= <arvo> |
          vari <janamuuttuja>

<arvo> ::= <kokonaisluku> |
          <muuttuja> |
```

```

+ <arvo> <arvo>      |
- <arvo> <arvo>      |
* <arvo> <arvo>      |
/ <arvo> <arvo>      |
^ <arvo> <arvo>      |
** <arvo> <arvo>     |
: <arvo> <arvo>      |
> <arvo> <arvo>      |
< <arvo> <arvo>      |
= <arvo> <arvo>      |
# <arvo> <arvo>      |
ja <arvo> <arvo>      |
tai <arvo> <arvo>      |
vai <arvo> <arvo>      |
ei <arvo>             |
de3 <arvo> <arvo>     |
de6 <arvo> <arvo>     |
rgb <arvo> <arvo> <arvo> |
pyta <arvo> <arvo>    |
pyte <arvo> <arvo>    |
asin <arvo>           |
acos <arvo>           |
atan <arvo>           |
sin <arvo>            |
cos <arvo>            |
tan <arvo>            |
sqrt <arvo>           |
sqr <arvo>            |
ln <arvo>             |
pi                   |
round <arvo>         |
trunc <arvo>         |
abs <arvo>           |

<janamuuttuja>      ::= <varaamaton sana>

<pistemuuttuja>     ::= <varaamaton sana>

<muuttuja>          ::= <varaamaton sana>

<kokonaisluku>      ::= (-)<luku>

<luku>              ::= <ei nolla>      |
                       <luku><numero>

<ei nolla>          ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<numero>            ::= 0 | <ei nolla>

<sanasijoitus>      ::= sana <sanamuuttuja> <tekstimuuttuja> <piste> <korkeus> <väri> <kulma> |
                       sana <sanamuuttuja> <tekstimuuttuja> <piste> <korkeus> <väri>      |
                       sana <sanamuuttuja> <tekstimuuttuja> <piste> <korkeus>

<tekstisijoitus>    ::= teksti <tekstimuuttuja> <merkkejä> |
                       tex <tekstimuuttuja> <teksti>

<teksti>            ::= te+ <teksti> <teksti>      |
                       <tekstimuuttuja>           |
                       tes <arvo> <luku> <luku>      |
                       istr <arvo>                 |
                       <merkkejä>

<sanamuuttuja>      ::= <varaamaton sana>

<tekstimuuttuja>    ::= <varaamaton sana>

<korkeus>           ::= <luku>      |
                       <arvo>      |
                       <muuttuja>

<merkkejä>          ::= chr(1)..chr(255)      |
                       <merkkejä>chr(1)..chr(255)

<varatut sanat>     ::= + | - | * | / | ** | ^ | : | pyta | pyte | asin | acos |
                       > | < | = | # | ja | tai | vai | ei | de3 | de6 |
                       rgb | atan | sin | cos | tan | sqrt | sqr | exp | ln |
                       pi | round | trunc | abs | ero | suunta | x | y |
                       te+ | tes | psx | psy | psp | laita | piste |
                       jana | sana | tex | teksti | rivi | jos | taso |
                       paks | vari | p1 | p2 | | -- | istr | tulk

```

Seuraavaksi koko koodi hieman paranneltuna:

Tässä on Lazarus-versio:

```
unit polsk;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs,
  LCLType, LCLIntf, Math;

type

  { TForm1 }

  TForm1 = class(TForm)
    procedure FormClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { private declarations }
  public
    { public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.lfm}

{ TForm1 }

procedure TForm1.FormClick(Sender: TObject);
const sataa = 400 ;
      tasoja = 3 ;
      isoluku = 1e300 ;
var ipa,jpa,kpa,npa,m,nmu,nte,taso,vaaka : integer ;
    rivi,mita,nimi : string ;
    jotain,xx,yy,zz : real ;
    temuut,tearvo,telaus : array[0..tasoja,1..sataa] of string ;
    muuttu,mulaus,piste,pilaus,jana,janaus : array[0..tasoja,1..sataa] of string ;
    muarvo,pihx,pihy,pihx1,pihy1,pihx2,pihy2,pihp : array[0..tasoja,0..sataa] of real ;
    pishv,tepihv : array[0..tasoja,0..sataa] of integer ;
    tepix,tepihv,tepiko,tepiku : array[0..tasoja,0..sataa] of real ;
    tepilaus,tepite,tepi : array[0..tasoja,1..sataa] of string ;
    Lista,Kuvasvg,Kuvahtm,VirheLista,RiviLista,MuuLista,Kirja : TStringList ;
    Perinta : array[0..tasoja] of TStringList ;
    StrPlus, StrMiinus, StrKerto, StrJako, StrPotel, StrPote2, StrSuhde : string ;
    StrPy, StrPe, StrAS, StrAC, StrRGB, StrAT, StrSi, StrCo, StrTa : string ;
    StrSqrt, StrSqr, StrExp, StrLn, StrPi, StrRound, StrTrunc, StrAbs : string ;
    StrDif, StrDir, StrKox, StrKoy, StrTeplu, StrTes, StrPsh, StrPsy : string ;
    StrPshs, StrPsp, StrLet, StrPiste, StrJana, StrTex, StrTeksti : string ;
    StrStr, StrLabel, StrIf, StrJaPa, StrJaVa, StrJaPl, StrJaP2 : string ;
    StrCom, StrGt, StrLt, StrEq, StrNe, StrAn, StrOr, StrXo, StrNo : string ;
    StrDe3, StrDe6, StrEval, StrI2Str, StrTaso, StrInher : string ;
    stabu,StrTmp : string ;

procedure kieli ;
begin
  StrPlus := '+' ;
  StrMiinus := '-' ;
  StrKerto := '*' ;
  StrJako := '/' ;
  StrPotel := '**' ;
  StrPote2 := '^' ;
  StrSuhde := ':' ;
  StrPy := 'pyta' ;
  StrPe := 'pyte' ;
  StrAS := 'asin' ;
  StrAC := 'acos' ;
  StrRGB := 'rgb' ;
  StrAT := 'atan' ;
  StrSi := 'sin' ;
  StrCo := 'cos' ;
  StrTa := 'tan' ;
  StrSqrt := 'sqrt' ;
```

```

StrSqr := 'sqr' ;
StrExp := 'exp' ;
StrLn := 'ln' ;
StrPi := 'pi' ;
StrRound := 'round' ;
StrTrunc := 'trunc' ;
StrAbs := 'abs' ;
StrDif := 'ero' ;
StrDir := 'suunta' ;
StrKox := 'x' ;
StrKoy := 'y' ;
StrTeplu := 'te+' ;
StrTes := 'tes' ;
StrI2Str := 'istr' ;
StrPsx := 'psx' ;
StrPsy := 'psy' ;
StrPsp := 'psps' ;
StrPsp := 'psp' ;
StrLet := 'laita' ;
StrPiste := 'piste' ;
StrJana := 'jana' ;
StrStr := 'sana' ;
StrTex := 'tex' ;
StrTeksti := 'teksti' ;
StrLabel := 'rivi' ;
StrIf := 'jos' ;
StrJaPa := 'paks' ;
StrJaVa := 'vari' ;
StrJaP1 := 'p1' ;
StrJaP2 := 'p2' ;
StrCom := '--' ;
StrGt := '>' ;
StrLt := '<' ;
StrEq := '=' ;
StrNe := '#' ;
StrAn := 'ja' ;
StrOr := 'tai' ;
StrXo := 'vai' ;
StrNo := 'ei' ;
StrDe3 := 'de3' ;
StrDe6 := 'de6' ;
StrEval := 'tulk' ;
StrTaso := 'taso' ;
StrInher := 'peri' ;
end ;

function ReaaliTex(xk: real ; nk,mk: integer) : string ;
var iik,rk : integer ;
    Str1,Str2 : string ;
begin
    Str1 := FloatToStrF(xk,ffFixed,nk,mk) ;
    rk := length(Str1) ;
    iik := pos(',',Str1) ;
    Str2 := Str1 ;
    if iik<>0 then
    begin
        if iik=1 then Str2 := '.'+copy(Str1,2,rk-1) ;
        if iik=rk then Str2 := copy(Str1,1,rk-1)+'.' ;
        if ((iik<>1) and (iik<>rk)) then
        begin
            Str2 := copy(Str1,1,iik-1)+'.'+copy(Str1,iik+1,rk-iik) ;
        end;
    end ;
    ReaaliTex := Str2 ;
{    ReaaliTex := FloatToStrF(xk,ffGeneral,nk,mk) ;    }
end ;

function ReaaliTeksti(xk: real ; nk,mk: integer) : string ;
begin
    ReaaliTeksti := FloatToStrF(xk,ffFixed,nk,mk) ;
end ;

function SamaTeksti(Str1,Str2 : string) : boolean ;
begin
    { SamaTeksti := AnsiSameText(Str1,Str2) ; }
    if Str1=Str2 then SamaTeksti := true else SamaTeksti := false ;
end ;

function EssiSameText(Str1,Str2 : string) : boolean ;
begin

```

```

    EssiSameText := AnsiSameText(Str1,Str2) ;
{ if Str1=Str2 then SamaTeksti := true else SamaTeksti := false ; }
end ;

function nor2utf(var Str1 : string) : string ;
var ik,nk : integer ;
    Str2,Str3 : string ;
begin
    nk := length(Str1) ;
    Str3 := '' ;
    for ik := 1 to nk do
    begin
        Str2 := Str1[ik] ;
        if Str2=chr(229) then Str2 := chr(195)+chr(165)
        else if Str2=chr(228) then Str2 := chr(195)+chr(164)
        else if Str2=chr(246) then Str2 := chr(195)+chr(182)
        else if Str2=chr(128) then Str2 := chr(226)+chr(130)+chr(172)
        else if Str2=chr(181) then Str2 := chr(194)+chr(181)
        else if Str2=chr(197) then Str2 := chr(195)+chr(133)
        else if Str2=chr(196) then Str2 := chr(195)+chr(132)
        else if Str2=chr(214) then Str2 := chr(195)+chr(150) ;
        Str3 := Str3 + Str2 ;
    end ;
    nor2utf := Str3 ;
end ;

function nor2utfd(var Str1 : string) : string ;
var ik,nk : integer ;
    Str2,Str3 : string ;
begin
    nk := length(Str1) ;
    Str3 := Str1 ;
    nor2utfd := Str3 ;
end ;

function skax(xs : real ; ns : integer) : real ;
var ik,nk : integer ;
    ker,ke2 : real ;
begin
    ik := 480 ;
    ke2 := 30 ;
    if vaaka=1 then
    begin
        ik := 960 ;
        ke2 := 0 ;
    end ;
    ker := 1 ;
    if ns>0 then ker := 1/(ke2+20) else ik := 0 ;
    skax := xs*ker + ik / 2 ;
end ;

function skay(ys : real ; ns : integer) : real ;
var ik,nk : integer ;
    ker,ke2 : real ;
begin
    ik := 680 ;
    ker := 1 ;
    ke2 := 30 ;
    if vaaka=1 then ke2 := 0 ;
    skay := ik - ys*ker ;
    if ns>0 then
    begin
        ker := 1/(ke2+20) ;
        skay := ik - ys*ker - ik / ns ;
        if ns=1 then skay := ik / 3 - ys*ker ;
    end ;
end ;

function skar(rs : real ; ns : integer) : real ;
var ik,nk : integer ;
begin
    ik := -1 ;
    skar := ik * rs ;
end ;

function pop(var Str1 : string) : string ;
var ik,nk : integer ;
begin
    ik := Pos(' ',Str1) ;
    nk := length(Str1) ;

```

```

    if ik>0 then
    begin
        pop := copy(Str1,1,ik-1) ;
        if ik=nk then Str1 := '' else Str1 := copy(Str1,ik+1,nk-ik) ;
    end
    else
    begin
        pop := Str1 ;
        Str1 := '' ;
    end ;
end ;

```

```

procedure pinoa(Str1 : string ; var Str2 : string) ;
begin
    if length(Str2)=0 then Str2 := Str1 else
        Str2 := Str1+' '+Str2 ;
    end ;

```

```

function popta(var Str1 : string) : string ;
var ik,nk : integer ;
begin
    ik := Pos(stabu,Str1) ;
    nk := length(Str1) ;
    if ik>0 then
    begin
        popta := copy(Str1,1,ik-1) ;
        if ik=nk then Str1 := '' else Str1 := copy(Str1,ik+1,nk-ik) ;
    end
    else
    begin
        popta := Str1 ;
        Str1 := '' ;
    end ;
end ;

```

```

procedure pinota(Str1 : string ; var Str2 : string) ;
begin
    if length(Str2)=0 then Str2 := Str1 else
        Str2 := Str1+stabu+Str2 ;
    end ;

```

```

function poisalku(Str1 : string) : string ;
var ik,jk,nk : integer ;
    tabu : string ;
begin
    ik := Pos(' ',Str1) ;
    tabu := chr(9) ;
    jk := Pos(tabu,Str1) ;
    nk := length(Str1) ;
    if ((ik=1) or (jk=1)) and (nk>1)) then
    begin
        Str1 := copy(Str1,ik+1,nk-ik) ;
        Str1 := poisalku(Str1) ;
    end ;
    if nk=1 then Str1 := '' ;
    poisalku := Str1 ;
end ;

```

```

procedure purargb(ii : integer; var ir,ig,ib : integer) ;
var k8,k16,ktmp : integer ;
begin
    k8 := 256 ;
    k16 := k8*k8 ;
    ib := ii div k16 ;
    ktmp := ii mod k16 ;
    ig := ktmp div k8 ;
    ir := ktmp mod k8 ;
end ;

```

```

function hexa(ij,ik,jk : integer) : string ;
var Str1 : string ;
begin
    Str1 := '#' + IntToHex(ij,2) + IntToHex(ik,2) + IntToHex(jk,2) ;
    hexa := Str1 ;
end ;

```

```

function arvo(var lause : string) : real ;
var Str1,Str2,Str3,Str4,Str5,Str6 : string ;
    ik,jk,ij,nk : integer ;
    ok : boolean ;

```



```

    arv1, arv2, arv3, arv4, arv5 : real ;
begin
    arvo := 0 ;
    Str1 := pop(lause) ;
    ik := length(Str1) ;
    if (SamaTeksti(Str1, StrPlus) or SamaTeksti(Str1, StrMiinus)) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if SamaTeksti(Str1, StrPlus) then arvo := arv2+arv3 else arvo := arv2-arv3;
    end
    else if (SamaTeksti(Str1, StrKerto) or SamaTeksti(Str1, StrJako)) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if SamaTeksti(Str1, StrKerto) then arvo := arv2*arv3 else
        if arv3<>0 then arvo := arv2/arv3 else
        if arv2<>0 then arvo := arv2/abs(arv2)*isoluku else arvo := 1 ;
    end
    else if (SamaTeksti(Str1, StrPotel) or SamaTeksti(Str1, StrPote2)) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        try
            arvo := Power(arv2, arv3) ;
        except
            arvo := Power(abs(arv2), arv3) ;
        end ;
    end
    else if SamaTeksti(Str1, StrSuhde) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        arvo := ArcTan2(arv2, arv3)*45/ArcTan(1) ;
    end
    else if SamaTeksti(Str1, StrGt) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if arv2>arv3 then arvo := 1 else arvo := 0 ;
    end
    else if SamaTeksti(Str1, StrLt) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if arv2<arv3 then arvo := 1 else arvo := 0 ;
    end
    else if SamaTeksti(Str1, StrEq) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if arv2=arv3 then arvo := 1 else arvo := 0 ;
    end
    else if SamaTeksti(Str1, StrNe) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if arv2<>arv3 then arvo := 1 else arvo := 0 ;
    end
    else if SamaTeksti(Str1, StrAn) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if ((arv2>0) and (arv3>0)) then arvo := 1 else arvo := 0 ;
    end
    else if SamaTeksti(Str1, StrOr) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if ((arv2>0) or (arv3>0)) then arvo := 1 else arvo := 0 ;
    end
    else if SamaTeksti(Str1, StrXo) then
    begin
        arv2 := arvo(lause) ;
        arv3 := arvo(lause) ;
        if ((arv2>0) xor (arv3>0)) then arvo := 1 else arvo := 0 ;
    end
    else if SamaTeksti(Str1, StrNo) then
    begin
        arv2 := arvo(lause) ;

```

```

    if arv2>0 then arvo := 0 else arvo := 1 ;
end
else if SamaTeksti(Str1,StrDe3) then
begin
    arv2 := arvo(lause) ;
    arv3 := arvo(lause) ;
    arvo := arv2+(arv3/1000) ;
end
else if SamaTeksti(Str1,StrDe6) then
begin
    arv2 := arvo(lause) ;
    arv3 := arvo(lause) ;
    arvo := arv2+(arv3/1000000) ;
end
else if SamaTeksti(Str1,StrPy) then
begin
    arv2 := arvo(lause) ;
    arv3 := arvo(lause) ;
    arvo := sqrt(sqr(arv2)+sqr(arv3)) ;
end
else if SamaTeksti(Str1,StrPe) then
begin
    arv2 := arvo(lause) ;
    arv3 := arvo(lause) ;
    arvo := sqrt(sqr(arv2)-sqr(arv3)) ;
end
else if SamaTeksti(Str1,StrAS) then
begin
    arv2 := arvo(lause) ;
    arvo := ArcTan(arv2/sqrt(1-sqr(arv2))) ;
end
else if SamaTeksti(Str1,StrAC) then
begin
    arv2 := arvo(lause) ;
    arvo := ArcTan(sqrt(1-sqr(arv2))/arv2) ;
end
else if SamaTeksti(Str1,StrRGB) then
begin
    ij := 255 ;
    jk := 0 ;
    ik := 127 ;
    ij := trunc(arvo(lause));
    ik := trunc(arvo(lause));
    jk := trunc(arvo(lause)) ;
    arvo := ij+256*ik+256*256*jk ;
end
else if SamaTeksti(Str1,StrAT) then arvo := ArcTan(arvo(lause))*45/ArcTan(1)
else if SamaTeksti(Str1,StrSi) then arvo := Sin(arvo(lause)/45*ArcTan(1))
else if SamaTeksti(Str1,StrCo) then arvo := Cos(arvo(lause)/45*ArcTan(1))
else if SamaTeksti(Str1,StrTa) then arvo := Tan(arvo(lause)/45*ArcTan(1))
else if SamaTeksti(Str1,StrSqrt) then arvo := sqrt(arvo(lause))
else if SamaTeksti(Str1,StrSqr) then arvo := sqr(arvo(lause))
else if SamaTeksti(Str1,StrExp) then arvo := exp(arvo(lause))
else if SamaTeksti(Str1,StrLn) then arvo := ln(arvo(lause))
else if SamaTeksti(Str1,StrPi) then arvo := pi
else if SamaTeksti(Str1,StrRound) then arvo := round(arvo(lause))
else if SamaTeksti(Str1,StrTrunc) then arvo := trunc(arvo(lause))
else if SamaTeksti(Str1,StrAbs) then arvo := abs(arvo(lause))
else if SamaTeksti(Str1,StrDif) then
begin
    ok := false ;
    arv3 := 0 ;
    arv4 := 0 ;
    arv5 := 0 ;
    arv2 := 0 ;
    Str2 := pop(lause) ;
    Str3 := pop(lause) ;
    if round(pisx[taso,0])>0 then
    begin
        ok := true ;
        for ik := 1 to round(pisx[taso,0]) do
            if SamaTeksti(Str2,piste[taso,ik]) then
            begin
                ok := true ;
                arv2 := pisx[taso,ik] ;
                arv4 := pisy[taso,ik] ;
            end ;
        end ;
        if not ok then
        begin
            arv2 := 0 ;

```

```

    arv4 := 0 ;
end ;
ok := false ;
for ik := 1 to round(pisx[taso,0]) do
if SamaTeksti(Str3,piste[taso,ik]) then
begin
    ok := true ;
    arv3 := pisx[taso,ik] ;
    arv5 := pisy[taso,ik] ;
end ;
if not ok then
begin
    arv3 := 0 ;
    arv5 := 0 ;
end ;
end ;
arvo := sqrt(sqr(arv2-arv3)+sqr(arv4-arv5)) ;
end
else if SamaTeksti(Str1,StrDir) then
begin
    ok := false ;
    arv3 := 0 ;
    arv4 := 0 ;
    arv5 := 0 ;
    arv2 := 0 ;
    Str2 := pop(lause) ;
    Str3 := pop(lause) ;
    if round(pisx[taso,0])>0 then
begin
    ok := false ;
    for ik := 1 to round(pisx[taso,0]) do
    if SamaTeksti(Str2,piste[taso,ik]) then
begin
        ok := true ;
        arv2 := pisx[taso,ik] ;
        arv4 := pisy[taso,ik] ;
end ;
if not ok then
begin
        arv2 := 0 ;
        arv4 := 0 ;
end ;
end ;
ok := false ;
for ik := 1 to round(pisx[taso,0]) do
if SamaTeksti(Str3,piste[taso,ik]) then
begin
        ok := true ;
        arv3 := pisx[taso,ik] ;
        arv5 := pisy[taso,ik] ;
end ;
if not ok then
begin
        arv3 := 0 ;
        arv5 := 0 ;
end ;
end ;
end ;
if arv2-arv3=0 then
begin
    arvo := 90 ;
    if arv4-arv5<0 then arvo := -90 ;
end
else arvo := ArcTan2(arv4-arv5,arv2-arv3)*180/pi ;
end
else if SamaTeksti(Str1,StrKox) then
begin
    ok := false ;
    Str2 := pop(lause) ;
    if round(pisx[taso,0])>0 then
begin
    ok := false ;
    for ik := 1 to round(pisx[taso,0]) do
    if SamaTeksti(Str2,piste[taso,ik]) then
begin
        ok := true ;
        arvo := pisx[taso,ik] ;
end ;
end ;
end ;
if not ok then arvo := 0 ;
end
else if SamaTeksti(Str1,StrKoy) then

```

```

begin
    ok := false ;
    Str2 := pop(lause) ;
    if round(pisx[taso,0])>0 then
    begin
        ok := false ;
        for ik := 1 to round(pisx[taso,0]) do
            if SamaTeksti(Str2,piste[taso,ik]) then
            begin
                ok := true ;
                arvo := pisy[taso,ik] ;
            end ;
        end ;
        if not ok then arvo := 0 ;
    end
    else if SamaTeksti(Str1,StrJaPa) then
    begin
        ok := false ;
        Str2 := pop(lause) ;
        if round(pisx2[taso,0])>0 then
        begin
            ok := false ;
            for ik := 1 to round(pisx2[taso,0]) do
                if SamaTeksti(Str2,jana[taso,ik]) then
                begin
                    ok := true ;
                    arvo := pisp[taso,ik] ;
                end ;
            end ;
            if not ok then arvo := 0 ;
        end
        else if SamaTeksti(Str1,StrJaVa) then
        begin
            ok := false ;
            Str2 := pop(lause) ;
            if round(pisx2[taso,0])>0 then
            begin
                ok := false ;
                for ik := 1 to round(pisx2[taso,0]) do
                    if SamaTeksti(lause,jana[taso,ik]) then
                    begin
                        ok := true ;
                        arvo := pisv[taso,ik] ;
                    end ;
                end ;
                if not ok then arvo := 0 ;
            end
            else if round(muarvo[taso,0])>0 then
            begin
                ok := false ;
                for ik := 1 to round(muarvo[taso,0]) do
                    if SamaTeksti(Str1,muuttu[taso,ik]) then
                    begin
                        ok := true ;
                        arvo := muarvo[taso,ik] ;
                    end ;
                end ;
                try
                    if not ok then arvo := StrToInt(Str1) ;
                except
                    Virhelista.Append(lause) ;
                    Virhelista.Append('Ei ole kokonaisluku: '+Str1) ;
                end ;
            end
            else
            begin
                try
                    arvo := StrToInt(Str1) ;
                except
                    Virhelista.Append(lause) ;
                    Virhelista.Append('Ei ole kokonaisluku: '+Str1) ;
                end ;
            end ;
        end ;
    end ;

function teksti(var lause : string) : string ;
var Str1,Str2,Str3,Str4,Str5 : string ;
    ik,nk : integer ;
    ok : boolean ;
    arv1,arv2,arv3,arv4,arv5,arv6,arv7 : real ;
    sil,si2,col,co2,r1,r2,det : real ;

```

```

begin
  Str2 := lause ;
  Str1 := pop(lause) ;
  ik := length(Str1) ;
  if SamaTeksti(Str1,StrTeplu) then
    begin
      Str4 := teksti(lause) ;
      Str5 := teksti(lause) ;
      teksti := Str4+Str5 ;
    end
  else if SamaTeksti(Str1,StrTes) then
    begin
      arv1 := arvo(lause) ;
      nk := round(arvo(lause)) ;
      ik := round(arvo(lause)) ;
      Str3 := ReaaliTeksti(arv1,nk,ik) ;
      teksti := Str3 ;
    end
  else if SamaTeksti(Str1,StrI2Str) then
    begin
      arv1 := arvo(lause) ;
      Str3 := IntToStr(round(arv1)) ;
      teksti := Str3 ;
    end
  else if round(pisy[taso,0])>0 then
    begin
      ok := false ;
      for ik := 1 to round(pisy[taso,0]) do
        if SamaTeksti(Str1,temuut[taso,ik]) then
          begin
            ok := true ;
            teksti := tearvo[taso,ik] ;
          end ;
        if not ok then teksti := Str2 ;
      end
    else teksti := Str2 ;
  end ;

procedure pistos(var lause : string;var x,y : real) ;
var Str1,Str2,Str3,Str4,Str5 : string ;
    ik,nk : integer ;
    ok : boolean ;
    arv1,arv2,arv3,arv4,arv5,arv6,arv7 : real ;
    si1,si2,col,co2,r1,r2,det : real ;
begin
  Str3 := lause ;
  Str1 := pop(lause) ;
  ik := length(Str1) ;
  if SamaTeksti(Str1,StrPsx) then
    begin
      pistos(lause,arv2,arv3) ;
      arv5 := arvo(lause)*arctan(1)/45 ;
      arv1 := arvo(lause) ;
      x := arv1 ;
      if cos(arv5)<>0 then
        y := arv3 - (arv2 - x) * sin(arv5) / cos(arv5)
      else
        y := arv3 ;
      end
    else if SamaTeksti(Str1,StrPsy) then
      begin
        pistos(lause,arv2,arv3) ;
        arv5 := arvo(lause)*arctan(1)/45 ;
        arv1 := arvo(lause) ;
        if sin(arv5)<>0 then
          x := arv2 - (arv3 - arv1) * cos(arv5) / sin(arv5)
        else
          x := arv2 ;
        y := arv1 ;
      end
    else if SamaTeksti(Str1,StrPsp) then
      begin
        pistos(lause,arv1,arv2) ;
        arv3 := arvo(lause)*arctan(1)/45 ;
        pistos(lause,arv4,arv5) ;
        arv6 := arvo(lause)*arctan(1)/45 ;
        x := (arv1+arv4)/2 ;
        y := (arv2+arv5)/2 ;
        if arv3 <> arv6 then
          begin

```

```

sil := sin(arv3);
si2 := sin(arv6);
col := cos(arv3);
co2 := cos(arv6);
r1 := col * arv2 - sil * arv1;
r2 := co2 * arv5 - si2 * arv4;
det := sil * co2 - si2 * col;
if det <> 0 then
begin
    x := (col * r2 - co2 * r1) / det;
    y := (sil * r2 - si2 * r1) / det;
end;
end;
end
else if SamaTeksti(Str1,StrPsp) then
begin
    pistos(lause,arv2,arv3) ;
    arv5 := arvo(lause)*arctan(1)/45 ;
    arv1 := arvo(lause) ;
    x := arv2+arv1*cos(arv5) ;
    y := arv3+arv1*sin(arv5) ;
end
else if SamaTeksti(Str1,StrJaP1) then
begin
    ok := false ;
    Str2 := pop(lause) ;
    if round(pisx2[taso,0])>0 then
    begin
        ok := false ;
        for ik := 1 to round(pisx2[taso,0]) do
        if SamaTeksti(Str2,jana[taso,ik]) then
        begin
            ok := true ;
            x := pisx1[taso,ik] ;
            y := pisy1[taso,ik] ;
        end ;
        end ;
    if not ok then
    begin
        y := 0 ;
        x := 0 ;
    end ;
end
else if SamaTeksti(Str1,StrJaP2) then
begin
    ok := false ;
    Str2 := pop(lause) ;
    if round(pisx2[taso,0])>0 then
    begin
        ok := false ;
        for ik := 1 to round(pisx2[taso,0]) do
        if SamaTeksti(Str2,jana[taso,ik]) then
        begin
            ok := true ;
            x := pisx2[taso,ik] ;
            y := pisy2[taso,ik] ;
        end ;
        end ;
    if not ok then
    begin
        y := 0 ;
        x := 0 ;
    end ;
end
else if round(pisx[taso,0])>0 then
begin
    ok := false ;
    for ik := 1 to round(pisx[taso,0]) do
    if SamaTeksti(Str1,piste[taso,ik]) then
    begin
        ok := true ;
        x := pisx[taso,ik] ;
        y := pisy[taso,ik] ;
    end ;
    if not ok then
    begin
        lause := Str3 ;
        x := arvo(lause) ;
        y := arvo(lause) ;
    end ;
end ;

```

```

end
else
begin
    lause := Str3 ;
    x := arvo(lause) ;
    y := arvo(lause) ;
end ;
end ;

procedure janat(var lause:string;var x,y,x2,y2,vp:real;var vari:integer) ;
var dump : real ;
begin
    pistos(lause,x,y) ;
    pistos(lause,x2,y2) ;
    if length(lause)>0 then vp := arvo(lause) else vp := 1 ;
    if length(lause)>0 then vari := round(arvo(lause)) else vari := 0 ;
end ;

procedure tulkinta(var lause : string) ;
var Str1,Str2,Str3,Str4,Str5 : string ;
    ik,ii,nk,nn : integer ;
    ok : boolean ;
    arv1,arv2,arv3,arv4,arv5,arv6,arv7 : real ;
    si1,si2,col,co2,r1,r2,det : real ;
begin
    Perinta[taso].Append(lause) ;
    Str1 := pop(lause) ;
    if SamaTeksti(Str1,StrLet) then
    begin
        Str2 := pop(lause) ;
        Str3 := lause ;
        arv2 := arvo(lause) ;
        if round(muarvo[taso,0])>0 then
        begin
            ok := false ;
            for ik := 1 to round(muarvo[taso,0]) do
                if SamaTeksti(Str2,muuttu[taso,ik]) then
                begin
                    ok := true ;
                    muuttu[taso,ik] := Str2 ;
                    mulous[taso,ik] := Str3 ;
                    muarvo[taso,ik] := arv2 ;
                end ;
            end ;
            if not ok then
            begin
                nk := round(muarvo[taso,0]) ;
                if nk<sataa then
                    muarvo[taso,0] := muarvo[taso,0]+1 ;
                ik := round(muarvo[taso,0]) ;
                muuttu[taso,ik] := Str2 ;
                mulous[taso,ik] := Str3 ;
                muarvo[taso,ik] := arv2 ;
            end ;
        end
        else
        begin
            muarvo[taso,0] := 1 ;
            muuttu[taso,1] := Str2 ;
            mulous[taso,1] := Str3 ;
            muarvo[taso,1] := arv2 ;
        end ;
    end
    else if SamaTeksti(Str1,StrPiste) then
    begin
        Str2 := pop(lause) ;
        Str3 := lause ;
        pistos(lause,arv2,arv3) ;
        if round(pisx[taso,0])>0 then
        begin
            ok := false ;
            for ik := 1 to round(pisx[taso,0]) do
                if SamaTeksti(Str2,piste[taso,ik]) then
                begin
                    ok := true ;
                    piste[taso,ik] := Str2 ;
                    pilaus[taso,ik] := Str3 ;
                    pisx[taso,ik] := arv2 ;
                    pisy[taso,ik] := arv3 ;
                end ;
            end ;
            if not ok then

```

```

begin
    nk := round(pisx[taso,0]) ;
    if nk<sataa then
        pisx[taso,0] := pisx[taso,0]+1 ;
        ik := round(pisx[taso,0]) ;
        piste[taso,ik] := Str2 ;
        pilaus[taso,ik] := Str3 ;
        pisx[taso,ik] := arv2 ;
        pisy[taso,ik] := arv3 ;
    end ;
end
else
begin
    pisx[taso,0] := 1 ;
    piste[taso,1] := Str2 ;
    pilaus[taso,1] := Str3 ;
    pisx[taso,1] := arv2 ;
    pisy[taso,1] := arv3 ;
end ;
end
else if SamaTeksti(Str1,StrJana) then
begin
    Str2 := pop(lause) ;
    Str3 := lause ;
    janat(lause,arv2,arv3,arv4,arv5,arv6,nn) ;
    if round(pisx2[taso,0])>0 then
begin
    ok := false ;
    for ik := 1 to round(pisx2[taso,0]) do
        if SamaTeksti(Str2,jana[taso,ik]) then
begin
            ok := true ;
            jana[taso,ik] := Str2 ;
            janaus[taso,ik] := Str3 ;
            pisx1[taso,ik] := arv2 ;
            pisy1[taso,ik] := arv3 ;
            pisx2[taso,ik] := arv4 ;
            pisy2[taso,ik] := arv5 ;
            pisp[taso,ik] := arv6 ;
            pispv[taso,ik] := nn ;
        end ;
    end if not ok then
begin
        nk := round(pisx2[taso,0]) ;
        if nk<sataa then
            pisx2[taso,0] := pisx2[taso,0]+1 ;
            ik := round(pisx2[taso,0]) ;
            jana[taso,ik] := Str2 ;
            janaus[taso,ik] := Str3 ;
            pisx1[taso,ik] := arv2 ;
            pisy1[taso,ik] := arv3 ;
            pisx2[taso,ik] := arv4 ;
            pisy2[taso,ik] := arv5 ;
            pisp[taso,ik] := arv6 ;
            pispv[taso,ik] := nn ;
        end ;
    end
end
else
begin
    pisx2[taso,0] := 1 ;
    jana[taso,1] := Str2 ;
    janaus[taso,1] := Str3 ;
    pisx1[taso,1] := arv2 ;
    pisy1[taso,1] := arv3 ;
    pisx2[taso,1] := arv4 ;
    pisy2[taso,1] := arv5 ;
    pisp[taso,1] := arv6 ;
    pispv[taso,1] := nn ;
end ;
end
else if SamaTeksti(Str1,StrTex) then
begin
    Str2 := pop(lause) ;
    Str3 := lause ;
    Str4 := teksti(lause) ;
    if round(pisy[taso,0])>0 then
begin
        ok := false ;
        for ik := 1 to round(pisy[taso,0]) do
            if SamaTeksti(Str2,temuut[taso,ik]) then

```



```

begin
    ok := true ;
    temuut[taso,ik] := Str2 ;
    telaus[taso,ik] := Str3 ;
    tearvo[taso,ik] := Str4 ;
end ;
if not ok then
begin
    nk := round(pisy[taso,0]) ;
    if nk<sataa then
        pisy[taso,0] := pisy[taso,0]+1 ;
        ik := round(pisy[taso,0]) ;
        temuut[taso,ik] := Str2 ;
        telaus[taso,ik] := Str3 ;
        tearvo[taso,ik] := Str4 ;
    end ;
end ;
end
else if SamaTeksti(Str1,StrStr) then
begin
    Str2 := pop(lause) ;
    Str3 := lause ;
    Str4 := pop(lause) ;
    pistos(lause,arv2,arv3) ;
    arv1 := arvo(lause) ;
    arv4 := 0 ;
    arv5 := 0 ;
    if length(lause)>0 then
        arv4 := arvo(lause) ;
        if length(lause)>0 then
            arv5 := arvo(lause) ;
        Str5 := teksti(Str4) ;
        if round(tepix[taso,0])>0 then
            begin
                ii := round(tepix[taso,0])+1 ;
                for ik := 1 to round(tepix[taso,0]) do
                    if SamaTeksti(Str2,tepi[taso,ik]) then
                        ii := ik ;
                    tepix[taso,0] := ii ;
                    tepilaus[taso,ii] := Str3 ;
                    tepix[taso,ii] := arv2 ;
                    tepiy[taso,ii] := arv3 ;
                    tepiv[taso,ii] := round(arv4) ;
                    tepiko[taso,ii] := arv1 ;
                    tepiku[taso,ii] := arv5 ;
                    tepi[taso,ii] := Str2 ;
                    tepite[taso,ii] := Str5 ;
                end
            end
        else
            begin
                ii := 1 ;
                tepix[taso,0] := ii ;
                tepilaus[taso,ii] := Str3 ;
                tepix[taso,ii] := arv2 ;
                tepiy[taso,ii] := arv3 ;
                tepiv[taso,ii] := round(arv4) ;
                tepiko[taso,ii] := arv1 ;
                tepiku[taso,ii] := arv5 ;
                tepi[taso,ii] := Str2 ;
                tepite[taso,ii] := Str5 ;
            end ;
        end
    end
else if SamaTeksti(Str1,StrTeksti) then
begin
    Str2 := pop(lause) ;
    Str3 := lause ;
    if round(pisy[taso,0])>0 then
        begin
            ok := false ;
            for ik := 1 to round(pisy[taso,0]) do
                if SamaTeksti(Str2,temuut[taso,ik]) then
                    begin
                        ok := true ;
                        temuut[taso,ik] := Str2 ;
                        telaus[taso,ik] := Str3 ;
                        tearvo[taso,ik] := Str3 ;
                    end ;
                if not ok then
                    begin
                        nk := round(pisy[taso,0]) ;

```

```

        if nk<sataa then
            pisy[taso,0] := pisy[taso,0]+1 ;
            ik := round(pisy[taso,0]) ;
            temuut[taso,ik] := Str2 ;
            telaus[taso,ik] := Str3 ;
            tearvo[taso,ik] := Str3 ;
        end ;
    end
else
begin
    pisy[taso,0] := 1 ;
    temuut[taso,1] := Str2 ;
    telaus[taso,1] := Str3 ;
    tearvo[taso,1] := Str3 ;
end
end ;
end ;

procedure rivit(Listaus : TStringList) ;
var Str1,Str2,Str3,Str4 : string ;
    ik,nk : integer ;
    ok : boolean ;
    arv1,arv2 : real ;
begin
    nk := Listaus.Count ;
    for ik := 0 to nk-1 do
        begin
            Str4 := Listaus[ik] ;
            Str4 := poisalku(Str4) ;
            Str1 := pop(Str4) ;
            if SamaTeksti(Str1,StrLabel) then
                begin
                    Str2 := pop(Str4) ;
                    Str3 := StrLet+ ' '+Str2+' '+IntToStr(ik) ;
                    tulkinta(Str3) ;
                end ;
            end ;
        end ;
    end ;

procedure ratko ;
var ik,jk,nk,nn : integer ;
    arv1,arv2,arv3,arv4,arv5 : real ;
    Str1,Str2,Str3 : string ;
begin
    nk := round(muarvo[taso,0]) ;
    if nk>0 then
        for ik := 1 to nk do
            begin
                Str1 := mulaus[taso,ik] ;
                arv1:= arvo(Str1) ;
                muarvo[taso,ik] := arv1 ;
            end ;
        nk := round(pisy[taso,0]) ;
        if nk>0 then
            for ik := 1 to nk do
                begin
                    Str2 := telaus[taso,ik] ;
                    Str1 := teksti(Str2) ;
                    tearvo[taso,ik] := Str1 ;
                end ;
            nk := round(pisx[taso,0]) ;
            if nk>0 then
                for ik := 1 to nk do
                    begin
                        Str1 := pilaus[taso,ik] ;
                        pistos(Str1,arv1,arv2) ;
                        pisx[taso,ik] := arv1 ;
                        pisy[taso,ik] := arv2 ;
                    end ;
                nk := round(pisx2[taso,0]) ;
                if nk>0 then
                    for ik := 1 to nk do
                        begin
                            Str1 := janaus[taso,ik] ;
                            janat(Str1,arv1,arv2,arv3,arv4,arv5,nn) ;
                            pisx1[taso,ik] := arv1 ;
                            pisyl[taso,ik] := arv2 ;
                            pisx2[taso,ik] := arv3 ;
                            pisy2[taso,ik] := arv4 ;
                            pisp[taso,ik] := arv5 ;

```

```

    pislv[taso,ik] := nn ;
end ;
nk := round(tepik[taso,0]) ;
if nk>0 then
for ik := 1 to nk do
begin
    Str1 := tepilaus[taso,ik] ;
    Str2 := pop(Str1) ;
    pistos(Str1,arv2,arv3) ;
    arv1 := arvo(Str1) ;
    arv4 := 0 ;
    arv5 := 0 ;
    if length(Str1)>0 then
    arv4 := arvo(Str1) ;
    if length(Str1)>0 then
    arv5 := arvo(Str1) ;
    Str3 := teksti(Str2) ;
    tepik[taso,ik] := arv2 ;
    tepiy[taso,ik] := arv3 ;
    tepiv[taso,ik] := round(arv4) ;
    tepiko[taso,ik] := arv1 ;
    tepiku[taso,ik] := arv5 ;
    tepite[taso,ik] := Str3 ;
    end ;
end ;

procedure tulkit(Listaus : TStringList) ;
var Str1,Str2,Str3,Str4 : string ;
    ji,nk,nkt,ita,ipa,isohko,loo,tassi : integer ;
    ok : boolean ;
    arv1,arv2 : real ;
begin
    nk := Listaus.Count ;
    isohko := 100000 ;
    loo := 0 ;
    ji := 0 ;
    repeat
        Str4 := Listaus[ji] ;
        Str2 := Str4 ;
        Str4 := poisalku(Str4) ;
        Str3 := Str4 ;
        Str1 := pop(Str3) ;
        if SamaTeksti(Str1,StrIf) then
        begin
            arv1 := arvo(Str3) ;
            arv2 := arvo(Str3) ;
            if arv1>0 then
            ji := round(arv2)
            else ji := ji + 1 ;
        end
        else if SamaTeksti(Str1,StrInher) then
        begin
            arv2 := arvo(Str3) ;
            ita := round(arv2) ;
            if ((ita>-1) and (ita<tasoja+1)) then
            begin
                nkt := Perinta[ita].Count ;
                if nkt>0 then
                for ipa := 0 to nkt-1 do
                begin
                    Str4 := Perinta[ita][ipa] ;
                    tulkinta(Str4) ;
                    loo := loo + 1 ;
                end ;
            end ;
            ji := ji + 1 ;
        end
        else if SamaTeksti(Str1,StrTaso) then
        begin
            tassi := round(arvo(Str3)) ;
            taso := tassi ;
            ji := ji + 1 ;
        end
        else if SamaTeksti(Str1,StrEval) then
        begin
            Str2 := teksti(Str3) ;
            tulkinta(Str2) ;
            ji := ji + 1 ;
        end
    else

```

```

begin
  if SamaTeksti(Str1,StrCom) then
    begin
      end
    else
      begin
        try
          tulkinta(Str4) ;
        except
          VirheLista.Append('Virhe: '+Str2) ;
        end ;
        ji := ji + 1 ;
      end ;
    end ;
    loo := loo + 1 ;
  until ((ji = nk) or (loo>isohko)) ;
end ;

procedure svgotsa(neli : integer) ;
var Str1,Str2,Str7 : string ;
begin
  Str7 := IntToStr(480*neli)+' '+IntToStr(680*neli)+'" ' ;
  if vaaka=1 then Str7 := IntToStr(960*neli)+' '+IntToStr(680*neli)+'" ' ;
  Str1 := '<svg width="20cm" height="28.235cm" viewBox="0 0 ' ;
  if vaaka=1 then Str1 := '<svg width="28.235cm" height="20cm" viewBox="0 0 ' ;
  Str2 := ' xmlns="http://www.w3.org/2000/svg" version="1.1">' ;
  Kuvasvg.Append(Str1+Str7) ;
  Kuvasvg.Append(Str2) ;
end ;

procedure svghanta ;
begin
  Kuvasvg.Append('</svg>') ;
end ;

function CreateRotatedFont(F : TFont; Angle : Integer) : hFont;
{-create a rotated font based on the font object F}
var
  LF : TLogFont;
begin
  FillChar(LF, SizeOf(LF), #0);
  with LF do begin
    lfHeight      := F.Height;
    lfWidth       := 0;
    lfEscapement   := Angle;
    lfOrientation  := 0;
    if fsBold in F.Style then
      lfWeight     := FW_BOLD
    else
      lfWeight     := FW_NORMAL;
    lfItalic       := Byte(fsItalic in F.Style);
    lfUnderline    := Byte(fsUnderline in F.Style);
    lfStrikeOut    := Byte(fsStrikeOut in F.Style);
    lfCharSet      := DEFAULT_CHARSET;
    StrPCopy(lfFaceName, F.Name);
    lfQuality      := DEFAULT_QUALITY;
    {everything else as default}
    lfOutPrecision := OUT_DEFAULT_PRECIS;
    lfClipPrecision := CLIP_DEFAULT_PRECIS;
    case F.Pitch of
      fpVariable : lfPitchAndFamily := VARIABLE_PITCH;
      fpFixed    : lfPitchAndFamily := FIXED_PITCH;
    else
      lfPitchAndFamily := DEFAULT_PITCH;
    end;
  end;
  CreateRotatedFont := CreateFontIndirect(LF);
end;

procedure piirra ;
var ik,jk,kk,nk,px,py,px2,py2,vv,ir,ig,ib,trw,nelja : integer ;
Str1,Str2,Str3,Str4,Str5,Str6,Str7,Str8,Str9 : string ;
kVanhaVontti,kUusiVontti : HFont ;
arv1,arv2 : real ;

begin
  nelja := 4 ;
  kk := round(tepix[taso,0]) ;
  nk := round(pisx2[taso,0]) ;
  if vaaka=1 then arv1 := 1 else arv1 := 2.5 ;

```

```

if kk > 0 then
begin
  for ik := 1 to kk do
  begin
    px := round(skax(tepik[taso,ik],taso)) ;
    py := round(skay(tepiy[taso,ik],taso)) ;
    vp := round(tepiko[taso,ik]/arv1) ;
    tr := round(skar(tepiku[taso,ik],taso)) ;
    arv2 := skar(tepiku[taso,ik],taso) ;
    vv := tepiv[taso,ik] ;
    purargb(vv,ir,ig,ib) ;
    canvas.Font.Size := round(0.6*vp) ;
    canvas.Font.Color := TColor(vv) ;
    if abs(arv2)>0 then
    begin
      trw := -round(10*skar(tepiku[taso,ik],0)) ;
      kUusiVontti := CreateRotatedFont(Canvas.Font, trw);
      kVanhaVontti := SelectObject(canvas.Handle,kUusiVontti) ;
      canvas.TextOut(px,py,nor2utf(tepite[taso,ik])) ;
      kUusiVontti := SelectObject(canvas.Handle,kVanhaVontti) ;
      DeleteObject(kUusiVontti) ;
    end
    else
    canvas.TextOut(px,py,nor2utf(tepite[taso,ik])) ;

    px := round(skax(tepik[taso,ik],taso)*nelja) ;
    py := round(skay(tepiy[taso,ik],taso)*nelja) ;
    vp := round(tepiko[taso,ik]*nelja/arv1) ;
    Str9 := ReaaliTex(arv2,4,2) ;
    Str6 := ''+IntToStr(vv)+' ' ;
    Str6 := 'rgb('+IntToStr(ir)+','+IntToStr(ig)+','+IntToStr(ib)+')' ;
    Str6 := '#'+IntToHex(vv,6)+' ' ;
    Str6 := ''+hexa(ir,ig,ib)+' ' ;
    Str7 := '<tspan dy=""'+IntToStr(vp)+'">' ;
    Str1 := '<text x=""'+IntToStr(px)+'" y=""'+IntToStr(py)+'" ' ;
    Str2 := 'fill='+Str6+' font-family="Verdana" font-size=""'+IntToStr(vp)+'" ' ;
    Str3 := 'transform="rotate('+Str9+' '+IntToStr(px)+' '+IntToStr(py)+')">' ;
    Str8 := nor2utf(tepite[taso,ik]) ;
    Str4 := Str7+Str8+'</tspan></text>' ;
    Str5 := Str1+Str2+Str3+Str4 ;
    Kuvasvg.Append(Str5) ;
  end ;
end ;
if nk > 0 then
begin
  for ik := 1 to nk do
  begin
    vp := round(pisp[taso,ik]) ;
    vv := pisp[taso,ik] ;
    canvas.Pen.Color := TColor(vv) ;
    canvas.Pen.Width := vp ;
    purargb(vv,ir,ig,ib) ;
    Str6 := 'rgb('+IntToStr(ir)+','+IntToStr(ig)+','+IntToStr(ib)+')' ;
    Str6 := '#'+IntToHex(vv,6) ;
    Str6 := hexa(ir,ig,ib) ;
    px := round(skax(pisx1[taso,ik],taso)) ;
    py := round(skay(pisy1[taso,ik],taso)) ;
    canvas.MoveTo(px,py) ;
    px := round(skax(pisx2[taso,ik],taso)) ;
    py := round(skay(pisy2[taso,ik],taso)) ;
    canvas.LineTo(px,py) ;
    px := round(skax(pisx1[taso,ik],taso)*nelja) ;
    py := round(skay(pisy1[taso,ik],taso)*nelja) ;
    Str1 := '<line x1=""'+IntToStr(px)+'" y1=""'+IntToStr(py)+'" x2="" ' ;
    px := round(skax(pisx2[taso,ik],taso)*nelja) ;
    py := round(skay(pisy2[taso,ik],taso)*nelja) ;
    Str2 := IntToStr(px)+'" y2=""'+IntToStr(py)+'" ' ;
    Str3 := 'style="stroke:'+Str6+';" ;
    Str4 := 'stroke-width:'+IntToStr(round(vp*nelja/arv1))+';stroke-linecap:round' />' ;
    Str5 := Str1+Str2+Str3+Str4 ;
    Kuvasvg.Append(Str5) ;
  end ;
end ;
end ;

begin
  Lista := TStringList.Create ;
  VirheLista := TStringList.Create ;
  RiviLista := TStringList.Create ;
  Kuvasvg := TStringList.Create ;

```

```

Kuvahtm := TStringList.Create ;
MuuLista := TStringList.Create ;
Kirja := TStringList.Create ;
for ipa := 0 to tasoja do Perinta[ipa] := TStringList.Create ;
if ClientWidth>800 then vaaka := 1 else
vaaka := 0 ;
if vaaka=1 then ClientWidth := 960 ;
Application.ProcessMessages;
stabu := chr(9) ;
kieli ;
taso := 0 ;
canvas.Pen.Color := clBlack ;
nimi := 'muuttujat.txt' ;
if FileExists(nimi) then MuuLista.LoadFromFile(nimi) ;
nimi := 'lauseet.txt' ;
if FileExists(nimi) then Lista.LoadFromFile(nimi) ;
for ipa := 0 to tasoja do
begin
    muarvo[taso,ipa] := 0 ;
    pisy[taso,ipa] := 0 ;
    pisy1[taso,ipa] := 0 ;
    pisy2[taso,ipa] := 0 ;
    tepix[taso,ipa] := 0 ;
    tepiy[taso,ipa] := 0 ;
end ;
if MuuLista.Count>0 then
begin
    m := MuuLista.Count ;
    rivit(MuuLista) ;
    tulkit(MuuLista) ;
end ;
if Lista.Count>0 then
begin
    m := Lista.Count ;
    rivit(Lista) ;
    tulkit(Lista) ;
    Kuvasvg.Clear ;
    svgotso(4) ;
    for ipa := 0 to tasoja do
    begin
        taso := ipa ;
        ratko ;
        piirra ;
    end ;
    taso := 0 ;
    svghanta ;
    Kuvasvg.SaveToFile('svg.svg') ;
{    Kuvahtm.Append('<html>') ; }
    Kuvahtm.Clear ;
    Kuvahtm.Append(chr(239)+chr(187)+chr(191)+'<html>') ;
    Kuvahtm.Append('<body>') ;
    Kuvahtm.AddStrings(Kuvasvg) ;
    Kuvahtm.Append('</body>') ;
    Kuvahtm.Append('</html>') ;
    Kuvahtm.SaveToFile('svg.htm') ;
end ;
VirheLista.SaveToFile('virheet.txt') ;
Lista.Clear ;
if pisy[taso,0]>0 then
for ipa := 1 to round(pisy[taso,0]) do
begin
    Lista.Append(temuut[taso,ipa]+stabu+telaus[taso,ipa]+stabu+tearvo[taso,ipa]) ;
end ;
Lista.SaveToFile('temuut.txt');
Lista.Clear ;
if muarvo[taso,0]>0 then
for ipa := 1 to round(muarvo[taso,0]) do
begin
    StrTmp := FloatToStrF(muarvo[taso,ipa],ffGeneral,10,3) ;
    Lista.Append(muuttu[taso,ipa]+stabu+mulaus[taso,ipa]+stabu+StrTmp) ;
end ;
Lista.SaveToFile('muuttu.txt') ;
for ipa := 0 to tasoja do
begin
    if Perinta[ipa].Count>0 then
    Perinta[ipa].SaveToFile('peri'+IntToStr(ipa)+'.txt') ;
    Perinta[ipa].Free ;

```

```

end ;
Lista.Free ;
VirheLista.Free ;
RiviLista.Free ;
Kuvavsvg.Free ;
Kuvahtm.Free ;
MuuLista.Free ;
Kirja.Free ;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  Left := 4 ;
  Top := 0 ;
  ClientHeight := 680 ;
  ClientWidth := 480 ;
  Caption := 'PN' ;
end;

end.

```

### Esimerkkinä tulkattava ohjelma:

```

laita xx1 50
laita xx2 + xx1 75
laita yy1 + 100 50
laita yy3 / yy1 4
laita at atan / 1 2
piste pp1 xx1 yy1
piste pp2 psx pp1 30 xx2
jana j11 pp1 pp2 8 rgb 255 0 0
jana jj2 10 399 399 0 1
jana jj3 10 399 10 10 1
jana jj4 10 10 399 10
teksti tx1 Voi
tex tx2 te+ tx1 ihme
tex tx4 tes at 6 3
tex tx3 te+ tx2 tx4
sana tt1 tx3 pp1 24 0 30

```

### Tiedosto svg.htm:

```

<html>
<body>
<svg width="20cm" height="28.235cm" viewBox="0 0 1920 2720"
xmlns="http://www.w3.org/2000/svg" version="1.1">
<text x="200" y="2120" fill="#000000" font-family="Verdana" font-size="96"
transform="rotate(-30 200,2120)"><tspan dy="96">Voi ihme 26,565</tspan></text>
<line x1="200" y1="2120" x2="500" y2="1947" style="stroke:#FF0000;stroke-width:32;stroke-linecap:round" />
<line x1="40" y1="1124" x2="1596" y2="2720" style="stroke:#000000;stroke-width:4;stroke-linecap:round" />
<line x1="40" y1="1124" x2="40" y2="2680" style="stroke:#000000;stroke-width:4;stroke-linecap:round" />
<line x1="40" y1="2680" x2="1596" y2="2680" style="stroke:#000000;stroke-width:4;stroke-linecap:round" />
</svg>
</body>
</html>

```

### Ja muodostuva kuva:

