

## Наредбе гранања

Под гранањем у програмирању подразумевамо ситуацију када се једна или више наредби може, али и не мора извршити у зависности од неког услова.

Наредбе гранања се, често, зову и наредбама преласка, па у том смислу, оне могу бити условне и безусловне. Ако извршење неке наредбе или групе наредби зависи од неког услова онда је то наредба условног преласка, у противном је то наредба безусловног преласка.

Наредба безусловног преласка је **GOTO** наредба, али она није структурисана.

Наредбе условног преласка су структурисане и могу бити:

- **If** наредба
- **Case** наредба

**If** наредба је наредба гранања са највише две гране, док **Case** наредба може имати највише 256 грана. Наредба **If** је општија, односно, сваки задатак са гранањем може се решити њеним коришћењем. Обрнуто не важи, тј. постоје задаци који се могу решити коришћењем **If** наредбе, а не могу се решити помоћу **Case** наредбе. **If** наредба се користи када у зависности од неког услова треба извршити један од највише два посла.

### Синтакса наредбе If

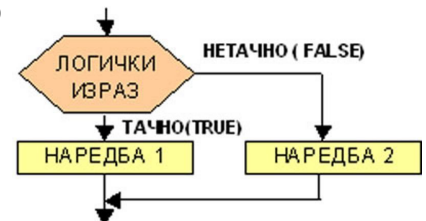
Наредба **If** може бити потпуна и непотпуна. Потпуна је ако има обе гране, а непотпуна ако има само једну грану.

Општи облик потпуне **If** наредбе је:

```
IF <uslov> THEN <naredba>
  ELSE <naredba>;
```

Општи облик непотпуне **If** наредбе је:

```
IF <uslov> THEN <naredba>;
```



Ако је потребно да се изврши група наредби иза службене речи **Then** и/или иза службене речи **Else** онда је обавезно употребити наредбу састављања за спајање те групе наредби, односно:

```
IF <uslov>
  THEN Begin <naredbe>
  End
  ELSE Begin <naredbe>
  End;
```

или

```
IF <uslov>
  THEN Begin <naredbe>
  End;
```

У случају да је услов испуњен извршава се наредба или група наредби иза службене речи **Then**, а ако услов није испуњен извршава се наредба или група наредби иза службене речи **Else**. Ако нема потребе да се изврши нека наредба у случају неиспуњења услова иза службене речи **Else** стоји празна наредба или се овај део наредбе изоставља. Ако нема потребе да се изврши нека наредба у случају испуњења услова иза службене речи **Then** обавезно стоји празна наредба или се услов преформулише тако да изостане потреба за службеном речи **Else**. Другим речима грана која почиње службеном речи **Then** је обавезна, тј. не може се изоставити, док се грана која почиње службеном речи **Else** може изоставити, ако се тиме не утиче на даљи ток програма.

Непосредно пре резервисане речи **Else** не сме се употребити граничник ; већ се као граничник наредбе употребљава размак.

Након извршења наредбе или групе наредби иза службене речи **Then** прелази се на прву наредбу испод **If** наредбе, тј. не постоји начин да се изврши наредба или група наредби које стоје иза **Then**, а затим и наредба или група наредби која стоји иза **Else**.

У оквиру наредби било које од грана **IF** наредбе може се наћи друга или више других **IF** наредби. У том случају треба водити рачуна о томе да се програмски преводилац, када наиђе на службену реч **Else**, враћа све до прве претходне службене речи **IF** и затвара је. Ако су све **IF** наредбе потпуне неће бити проблема са ужљебљењем. Али ако су неке од наредби непотпуне, (код програмера) може доћи до забуне (што може довести до алгоритамске грешке или до добијања *чудних*, неочекиваних резултата у раду програма). У таквим случајевима, ужљебљене непотпуну **IF** наредбу у **Then** грани, обавезно, треба писати у оквиру наредбе састављања (без обзира на то што се, можда, у тој грани извршава само једна наредба):

```
IF <uslov1>
  THEN Begin IF <uslov2>
            THEN <naredba1>;
            End
        ELSE <naredba2>;
        <naredba3>;
```

Ако би се изоставила наредба састављања у случају да није испуњен <услов2> извршила би се <наредба2> иза службене речи **Else**, а у случају неиспуњења првог услова извршила би се прва наредба испод службене речи **Else** што би било логичка грешка и довело би до грешке у неким резултатима програма, овако, у случају да тај <услов2> није испуњен, прелази се на <наредбу3>.

У случају да се непотпуна **IF** наредба налази иза службене речи **Else** наредба састављања није неопходна (али се може користити), јер је онда сасвим јасно која се наредба извршава у случају да неки од услова није испуњен:

```
IF <uslov>
  THEN <naredba1>
  ELSE IF <uslov2>
        THEN <naredba2>;
        <naredba3>;
```

### Синтакса наредбе Case

Када је услов за гранање сложенији, тј. када захтева неколико грана (више од две) јер има више случајева који захтевају извршење различитих наредби, проблем се може решити вишеструком употребом **IF** наредбе уз преформулисање самог услова у неколико једноставнијих услова, али је, увек, много лакше, практичније и једноставније употребити наредбу **CASE**.

Општи облик ове наредбе је:

```
CASE <izraz> OF
  <slucaj1> : <naredba>;
  <slucaj2> : <naredba>;
  <slucaj3>, <slucaj4>, ... , <slucajK> : <naredba>;
  <slucajK+1> .. <slucajK+M> : <naredba>;
  :
  <slucajN> : <naredba>;
  ELSE <naredba>;
END;
```

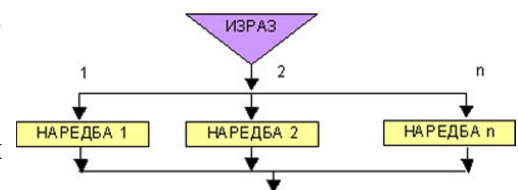
Службена реч **End**, на крају **CASE** наредбе је обавезна и не сме се изоставити. Број случајева зависи од броја вредности селектора које захтевају различите активности (наредбе), а највише их може бити 255. У случају да селектор <izraz> може имати више од 255 вредности морамо комбиновати две или више **CASE** наредби.

У **CASE** наредби селектор <izraz> мора бити простог, уређеног типа (*boolean, char, integer*), а случајеви тј. вредности <izraz>-а могу бити задате као

- појединачне вредности (<slucaj1> или <slucaj2>),
- групе (<slucaj3>, <slucaj4>, ... , <slucajK>) или
- интервали вредности (<slucajK+1> .. <slucajK+M>).

Код групних случајева морају се навести све вредности које захтевају извршење исте наредбе. Код интервала се подразумева да се наредба извршава за све вредности од почетне до крајње вредности интервала, тј. не могу се изузимати неке вредности из унетог интервала.

**Else** грана обухвата све случајеве који нису претходно поменути. Важно је поменути да вредности у случајевима не морају бити уређени у растући или опадајући поредак, али није дозвољено да се нека вредност појављује у два или више случајева.



Ако у неким од случајева (појединачним, групама или интервалима) треба извршити више од једне наредбе онда те наредбе, **обавезно**, групишемо наредбом састављања:

```

CASE <izraz> OF
  <slucaj1> : <naredba>;
  <slucaj2> : Begin <naredbe>;
              End;
  <slucaj3>, <slucaj4>, ... , <slucajK> : Begin <naredbe>;
              End;
  :
  <slucajN> : <naredba>;
  ELSE Begin <naredbe>;
          End
END;
```

Ако потребе алгоритма захтевају, може се у оквиру једне **CASE** наредбе наћи још једна или више **CASE** наредби, но како свака од ових наредби има своје **END** не може доћи до забуне који случај припада којем изразу.

До забуне (код програмера) може доћи само код комбиноване употребе **CASE** и непотпуне **IF** наредбе у последњем случају **CASE** наредбе, када се **Else** грана може схватити као део **CASE** наредбе или као део **IF** наредбе. Проблем се разрешава тако што се пре **Else** гране које припада **CASE** наредби обавезно ставља као граничник ; што у **IF** наредби није дозвољено, а може се и непотпуна **IF** наредба писати у оквиру наредбе састављања:

```

CASE <izraz> OF
  <slucaj1> : CASE <izraz> OF
                <slucaj1> : <naredbe>;
                <slucaj2> : <naredbe>;
                ...
                <slucajN> : <naredbe>;
                ELSE <naredbe>;
            END;
  <slucaj2> : <naredbe>;
  :
  <slucajM> : If <uslov>
              then <naredba>
              else <naredba>;
  :
  <slucajN> : If <uslov>
              then <naredba>;
  ELSE <naredbe>;
END;
```

## Алгоритми са гранањем

### • одређивање минимума и максимума два и три броја

Одређивање већег или мањег од два уписана броја је најједноставнији пример употребе **If** наредбе. Може се решити коришћењем додатне променљиве *min* (*max*) или без ње. Једноставније је објаснити пример са додатном променљивом.

- Одређивање већег од два броја:

```

If a>b
  then max:=a
  else max:=b;
```

- Одређивање мањег од два броја:

```

If a<b
  then min:=a
  else min:=b;
```

Ако треба одредити и мањи и већи од два броја онда ћемо спојити ове две наредбе у једну:

```

If a>b
  then begin max:=a;min:=b;
        end
  else begin min:=a;max:=b;
        end;
```

Нешто је тежи случај када треба одредити најмањи или највећи од три уписана броја јер се морају упоређивати три броја сваки са сваким. То значи да у овом случају имамо најмање две **If** наредбе. И овде ћемо користити додатну променљиву јер је поступак једноставније објаснити.

- Одређивање највећег од три броја:

```

If a>b // већи од бројева а и б проглашавамо највећим
  then max:=a
  else max:=b;
```

```
If c>max // проверавамо да ли је трећи број већи од највећег
then max:=c;
```

- Одређивање најмањег од три броја:

```
If a<b // мањи од бројева а и b проглашавамо најмањим
then min:=a
else min:=b;
If c<min // проверавамо да ли је трећи број мањи од најмањег
then min:=c;
```

- Ако треба одредити и најмањи и највећи од три броја онда ћемо то урадити на следећи начин:

```
If a>b // већи од бројева а и b проглашавамо највећим, а мањи најмањим
then begin max:=a;min:=b;
end
else begin min:=a;max:=b;
end;
If c>max // проверавамо да ли је трећи број већи од највећег
then max:=c
else If c<min // проверавамо да ли је трећи број мањи од најмањег
then min:=c;
```

Могући су и неки другачији алгоритми, али, чини се, да је овај врло једноставан и прилично разумљив.

- **уређивање два и три броја у монотono неопadaјући или нерастући поредак**

Ако је потребно три броја уредити по величини онда није довољно само упоредити сваки са сваким, већ се морају и размењивати вредности између променљивих које нису у жељеном поретку. Зато је овај пример комбинација одређивања максимума и минимума и размене вредности два броја. Проблем ћемо решити тако што ћемо, најпре, упоређивати два броја и наместити да први буде мањи од другог, а затим ћемо трећи број упоређивати са сваким од претходна два и, по потреби, размењивати вредности. Записаћемо то овако:

```
If a>b
then begin pom:=a;a:=b;b:=pom;
end;
If a>c
then begin pom:=a;a:=c;c:=pom;
end;
If b>c
then begin pom:=b;b:=c;c:=pom;
end;
```

Овакво уређење чини растући (прецизније, неопadaјући поредак јер нико не гарантује да неки од ова три броја нису једнаки). Посматрајмо то на једноставном, конкретном примеру у табели у којој сваки ред представља један алгоритамски корак:

a	b	c	шта се дешава
7	3	6	
3	7	6	упоређујемо прва два броја и пошто је први већи мењамо им места
3	7	5	упоређујемо први и трећи број и пошто је трећи већи ништа се не мења
3	5	7	упоређујемо други и трећи број и пошто је други већи мењамо им места

Ако би требало бројеве уредити у опадајући (односно, нерастући) поредак, једина измена била би промена релације веће у мање (или се бројеви уреде као у претходном примеру, а испишују се обрнутим редоследом). Следећи низ наредби уређује бројеве у такав поредак:

```
If a<b
then begin pom:=a;a:=b;b:=pom;
end;
If a<c
then begin pom:=a;a:=c;c:=pom;
end;
If b<c
then begin pom:=b;b:=c;c:=pom;
end;
```

- **одређивање сутрашњег и јучерашњег датума**

Код задатака са датумима мора се водити рачуна о томе да немају сви месеци једнак број дана и да је број дана у години ограничен на 365, односно, 366 код преступне године. Максимални број дана у месецу сви знамо, зато ћемо се подсетити правила која дефинишу да ли је година преступна. Већина би сваку четврту годину прогласила преступном, односно, ако је година дељива са четири она је преступна. Међутим то је погрешно, јер је познато да је 1900. година број који је дељив са 4, али та година није била преступна. Зато није довољно

проверити да ли је година дељива са 4 да би се утврдило да ли је преступна, већ не сме бити последња година века, тј. не сме бити дељива са 100. Но и то није комплетан услов јер је 2000. година дељива са 100, а била је преступна. Зато, комплетан услов, да би нека година била преступна, гласи:

**година је преступна ако је дељива са 4 и није дељива са 100 или ако је дељива са 400.**

Када смо ово утврдили можемо одредити сутрашњи датум ако је дат данашњи. У већини случајева, сутрашњи датум је за један дан већи од данашњег. Ово правило не важи само ако је данас последњи дан у месецу. У том случају сутра је први дан следећег месеца. Да бисмо знали да ли је последњи дан у месецу морамо тестирати редни број месеца и редни број дана из данашњег датума. Редни број месеца може да се повећа за један само ако данашњи датум није 31.12. неке године. У том случају сутрашњи датум је 1.1. следеће године. Сада ћемо написати, уз објашњења у коментарима, главни део програма који прати описани алгоритам:

```
If (g mod 4=0)and(g mod 100<>0) // ако је година дељива са 4 и ако није дељива са 100
  or(g mod 400=0)           // или ако је дељива са 400
  then p:=1                 // година је преступна
  else p:=0;                // у противном година није преступна
Case m of                    // сада одређујемо максимални број дана у месецу
  2 : bd:=28+p;             // ако је фебруар има 28 или 29 дана
  4,6,9,11 : bd:=30;       // четири месеца у години имају по 30 дана
  else bd:=31;             // остали месеци имају по 31 дан
end;
d:=d+1;                     // сутрашњи датум је за 1 већи
If d>bd                       // ако је редни број дана већи од максималног броја дана
  then begin d:=1;         // прешли смо у други месец, односно, датум је 1.
        m:=m+1;           // у том случају месец је за 1 већи
  end;
If m>12                       // ако смо добили редни број месеца већи од 12
  then begin m:=1;         // прешли смо у другу годину, односно, месец је 1.
        g:=g+1;           // у том случају година је за 1 већа
  end;
```

Сада треба исписати сутрашњи датум :  $d$ ,  $m$ ,  $g$ .

Алгоритам за одређивање јучерашњег датума ћемо исписати без објашњења јер је прилично јасан, ако смо овај претходни добро разумели:

```
If (g mod 4=0)and(g mod 100<>0)or(g mod 400=0)
  then p:=1
  else p:=0;
Case m-1 of
  2 : bd:=28+p;
  4,6,9,11 : bd:=30;
  else bd:=31;
end;
d:=d-1;
If d<1
  then begin d:=bd;m:=m-1;
  end;
If m<1
  then begin m:=12;g:=g-1;
  end;
```

**Case** наредбу коју смо користили у оба алгоритма, лако смо могли заменити са две **If** наредбе, али је овакво решење ефектније. У проблемима са датумима требало би, пре извршења приказаних и сличних алгоритама, тестирати унете податке. Било би бесмислено тражити сутрашњи или јучерашњи ако је унети датум 33.13. било које године.

- **приказ назива дана у недељи на основу учитаног редног броја дана**

Овај пример је демонстрација предности **Case** у односу на **If** наредбу (у сличним ситуацијама). Редни број дана у недељи једнозначно одређује назив дана. Будући да је ово врло једноставан пример, нећемо се задржавати на непотребним објашњењима, али ћемо навести прво решење са **Case**, а затим и са **If** наредбом ради упоређења.

```
Case rb of
  1 : ime:='Prvi dan u nedelji je ponedeljak.';
  2 : ime:='Drugi dan u nedelji je utorak.';
  3 : ime:='Treci dan u nedelji je sreda.';
  4 : ime:='Cetvrti dan u nedelji je четвртак.';
  5 : ime:='Peti dan u nedelji je петак.';
  6 : ime:='Sesti dan u nedelji je subota.';
  7 : ime:='Sedmi dan u nedelji je nedelja.';
  else ime:='Pogresan unos rednog broja dana u nedelji.';
end;
```

```

If rb = 1
  then ime:='Prvi dan u nedelji je ponedeljak.'
  else If rb = 2 then ime:='Drugi dan u nedelji je utorak.'
  else If rb = 3 then ime:='Treci dan u nedelji je sreda.'
  else If rb = 4 then ime:='Cetvrti dan u nedelji je cetvrtak.'
  else If rb = 5 then ime:='Peti dan u nedelji je petak.'
  else If rb = 6 then ime:='Sesti dan u nedelji je subota.'
  else If rb = 7 then ime:='Sedmi dan u nedelji je nedelja.'
  else ime:='Pogresan unos rednog broja dana u nedelji.';
end;

```

• приказ назива дана у недељи на основу учитаног редног броја дана

Овај пример је још једна демонстрација предности *Case* у односу на *If* наредбу. Редни број месеца у години једнозначно одређује назив месеца. Будући да је ово врло једноставан пример, нећемо се задржавати на непотребним објашњењима, али ћемо и овде навести прво решење са *Case*, а затим и са *If* наредбом ради упоређења.

```

Case rb of
  1 : ime:='Prvi mesec u godini je januar.';
  2 : ime:='Drugi mesec u godini je februar.';
  3 : ime:='Treci mesec u godini je mart.';
  4 : ime:='Cetvrti mesec u godini je april.';
  5 : ime:='Peti mesec u godini je maj.';
  6 : ime:='Sesti mesec u godini je jun.';
  7 : ime:='Sedmi mesec u godini je jul.';
  8 : ime:='Osmi mesec u godini je avgust.';
  9 : ime:='Deveti mesec u godini je septembar.';
 10: ime:='Deseti mesec u godini je oktobar.';
 11: ime:='Jedanaesti mesec u godini je novembar.';
 12: ime:='Dvanaesti mesec u godini je decembar.';
  else ime:='Pogresan unos rednog broja meseca u godini.';
end;

If rb = 1
  then ime:='Prvi mesec u godini je januar.'
  else If rb = 2 then ime:='Drugi mesec u godini je februar.'
  else If rb = 3 then ime:='Treci mesec u godini je mart.'
  else If rb = 4 then ime:='Cetvrti mesec u godini je april.'
  else If rb = 5 then ime:='Peti mesec u godini je maj.'
  else If rb = 6 then ime:='Sesti mesec u godini je jun.'
  else If rb = 7 then ime:='Sedmi mesec u godini je jul.'
  else If rb = 8 then ime:='Osmi mesec u godini je avgust.';
  else If rb = 9 then ime:='Deveti mesec u godini je septembar.';
  else If rb = 10 then ime:='Deseti mesec u godini je oktobar.';
  else If rb = 11 then ime:='Jedanaesti mesec u godini je novembar.';
  else If rb = 12 then ime:='Dvanaesti mesec u godini je decembar.';
  else ime:='Pogresan unos rednog broja meseca u godini.';
end;

```

У примерима са редним бројевима дана и месеца, тестирали смо и исправност унетих података.

### Питања на која треба обратити пажњу

- Шта је гранање?
- Врсте гранања.
- Синтакса *If* наредбе
- Ужљебљене *If* наредбе
- Непотпуна *If* наредба
- *Else* грана у *If* наредби
- Синтакса *Case* наредбе
- Селектор у *Case* наредби
- Случајеви у *Case* наредби
- *Else* грана у *Case* наредби
- *Case* наредба у *Case* наредби
- *If* наредба у *Case* наредби



## Задаци са гранањем

### • Саставити програм који испитује да ли је уписани број паран.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 310; ClientWidth = 350; Color = Teal; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, лабелу за комуникацију са корисником, едит за унос, лабелу за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Да ли је број паран?; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едит за унос:

**Font = Bold, Black; Left = 200; Top = 80; Width = 120.**

Лабела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = 22, Lime; Height = 80; Layout = tlCenter.**

Лабела за комуникацију са корисником:

**AutoSize = False; Caption = Упишите цео број; Height = 36; Layout = tlCenter; Left = 20; Top = 80; Width = 190.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (20, 210, 210); Top = (150, 150, 210); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Paran*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Label3.Caption:=' ';
      Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    b:integer;
begin b:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(b);
      If Odd(b)
      then a:='Број '+IntToStr(b)+' је непаран.'
      else a:='Број '+IntToStr(b)+' је паран.';
      Label3.Caption:=a;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

### • Саставити програм који испитује да ли су два уписана броја дељива.

За два броја кажемо да су дељива ако један од њих при дељењу другим даје остатак 0.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 350; ClientWidth = 350; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, лабелу за комуникацију са корисником, два едита за унос, лабелу за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Дељивост бројева; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едити за унос:

**Font = Bold, Black; Left = 190; Top = (80, 130); Width = 120.**

Лабела за испис резултата:

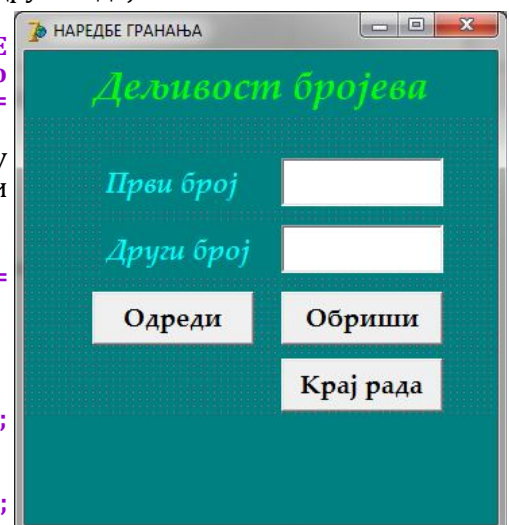
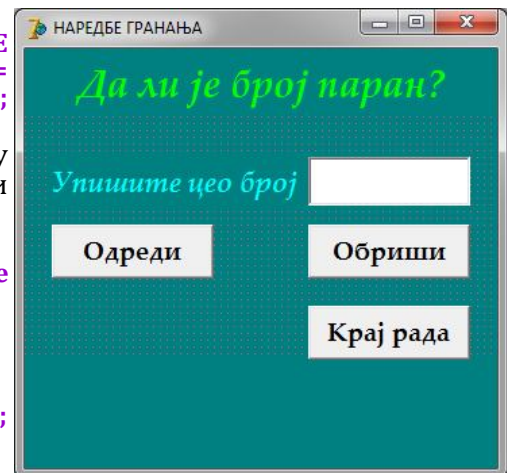
**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = 22, Lime; Height = 80; Layout = tlCenter.**

Лабеле за комуникацију са корисником:

**AutoSize = False; Caption = (Први број, Други број); Height = 36; Layout = tlCenter; Left = 60; Top = (80, 130); Width = 190.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (50, 190, 190); Top = (180, 180, 230); Width = 120.**



Сачуваћемо дизајнирану форму у креирани фолдер *Paran*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Label4.Caption:=' ';
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    b,c:integer;
begin
  b:=StrToIntDef(Edit1.Text,0); Edit1.Text:=IntToStr(b);
  c:=StrToIntDef(Edit2.Text,0); Edit2.Text:=IntToStr(c);
  If b mod c=0
  then a:='Број '+IntToStr(b)+' је дељив бројем '+IntToStr(c)+'.'
  else If c mod b=0
  then a:='Број '+IntToStr(c)+' је дељив бројем '+IntToStr(b)+'.'
  else a:='Бројеви нису дељиви.';
  Label4.Caption:=a;
  Edit1.ReadOnly:=true;
  Edit2.ReadOnly:=true;
  Button2.SetFocus;
end;
```

- **Саставити програм који тестира да ли је на тастатури притиснуто слово, цифра или знак.**

На тастатури је велики број тастера, да не бисмо користили јако сложене **if** наредбе или велики број наредби у решењу овог задатка користимо предности скуповног типа. Тастере ћемо груписати у скуп слова (при чему ћемо користити само велика слова да бисмо поједноставили запис решења), скуп цифара и остале.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 190; ClientWidth = 400; Font = Palatino Linotype, 22; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, лабелу за испис резултата и тастер за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Притисните неки тастер; Height = 50; Layout = tlCenter.**

Лабела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Height = 90; Layout = tlCenter.**

Тастер за управљање:

**Caption = Крај рада; Font = Bold; Height = 40; Left = (50, 190, 190); Top = (180, 180, 230); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Taster*, а затим, двокликом мишем на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

У прозору *Object Inspector*, најпре треба селектовати форму (објекат *Form1*), а затим прећи на листић *Events* и двокликом на догађај **OnKeyPress** декларисати и дефинисати следећу процедуру:

```
procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
var a:string;
begin
  If UpCase(key) in ['A'..'Z']
  then a:='Притиснуто је слово '+key
  else if key in ['0'..'9']
  then a:='Притиснута је цифра '+key
  else a:='Притиснут је знак '+key;
  Label2.Caption:=a;
end;
```

- **Саставити програм који за унету годину одређује да ли је година преступна.**

Година је преступна ако је дељива са 4 и није дељива са 100 или ако је дељива са 400. То значи да је 2000. била преступна, 1900. није, свака четврта између је била преступна, а 2100. неће бити.

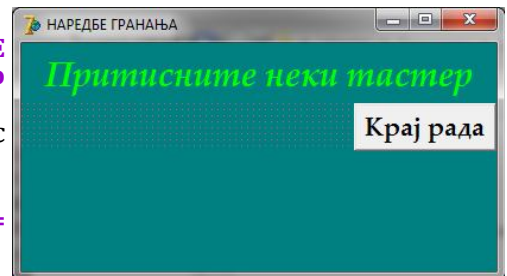
Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 310; ClientWidth = 350; Color = Teal; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, лабелу за комуникацију са корисником, едит за унос, лабелу за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Преступна година; Font = 22, Lime; Height = 50; Layout = tlCenter.**





Едит за унос:

**Font = Bold, Black; Left = 200; MaxLength = 2; Top = 80; Width = 120.**

Лабела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = 22, Lime; Height = 80; Layout = tlCenter.**

Лабела за комуникацију са корисником:

**AutoSize = False; Caption = Упишите цео број; Height = 36; Layout = tlCenter; Left = 20; Top = 80; Width = 190.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (20, 210, 210); Top = (150, 150, 210); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Prestupna*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Label3.Caption:=' ';
      Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var a:string;g:integer;
begin g:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(g);
      If (g mod 4=0)and(g mod 100<>0)or(g mod 400=0)
      then a:=IntToStr(g)+' година'+#13+'је преступна.'
      else a:=IntToStr(g)+' година'+#13+'није преступна.';
      Label3.Caption:=a;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

end;

### • Саставити програм који одређује решење линеарне једначине са једном непознатом.

Линеарну једначину можемо решити на следећи начин:

Ако је линеарни коефицијент једначине различит од нуле онда једначина има јединствено решење:

$$a \cdot x + b = 0 \quad a \cdot x = -b \quad / : a \neq 0 \quad x = -\frac{b}{a}$$

Ако је линеарни коефицијент једнак нули онда имамо овакву ситуацију  $0 \cdot x + b = 0$  и два случаја:

- ако је слободни члан  $b = 0$  решење једначине је било који број, односно једначина је неодређена,

- ако је слободни члан  $b \neq 0$  различит од нуле једначина нема решења, односно, једначина је немогућа.

Значи, након уноса коефицијената, најпре тестирамо линеарни члан, а затим слободни члан, па затим исписујемо одговарајући коментар и решење једначине.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 330; ClientWidth = 350; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, лабелу за комуникацију са корисником, два едита за унос, лабелу за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Линеарна једначина; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едити за унос:

**Font = Bold, Black; Left = (70, 220); Top = 80; Width = 100.**

Лабела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = 22, Lime; Height = 110; Layout = tlCenter.**

Лабеле за комуникацију са корисником:

**AutoSize = False; Caption = (a =, b =); Height = 36; Layout = tlCenter; Left = (30, 180); Top = 80; Width = 40.**

Тастери за управљање:

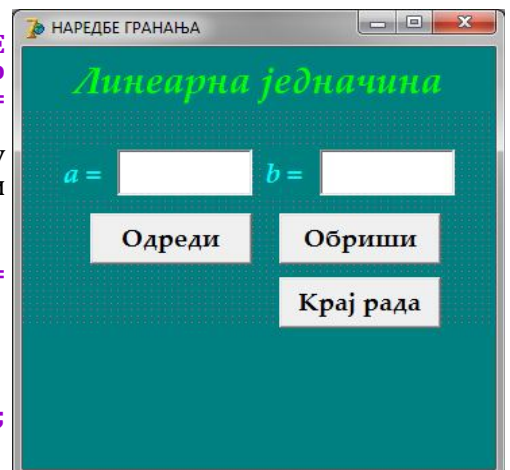
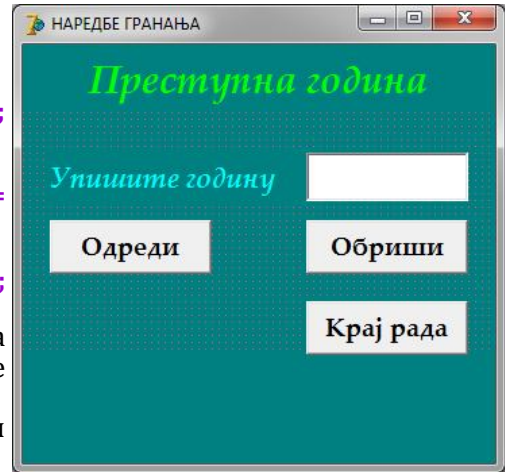
**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (50, 190, 190); Top = (130, 130, 180); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Linearna*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
```



```

Edit2.Clear;Edit2.ReadOnly:=false;
Label4.Caption:=' ';
Edit1.SetFocus;

```

end;

Двоклик на тастер *Одреди*:

```

procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    b,c:real;
begin
b:=StrToFloatDef(Edit1.Text,1);Edit1.Text:=FloatToStr(b);
c:=StrToFloatDef(Edit2.Text,2);Edit2.Text:=FloatToStr(c);
a:=Format('%-27s'+#13+'%1.2f', ['Једначина',b])+#183+'x';
If c>=0 then a:=a+' + ';
a:=a+Format('%1.2f', [Abs(c)])+' = 0'+#13;
If b=0
then If c=0
then a:=a+'је неодређена.'
else a:=a+'је немогућа.'
else a:=a+'има решење x = '+Format('%1.2f', [-c/b]);
Label4.Caption:=a;
end;

```

У испису решењу смо користили *ASCII* кодове #13, за прелазак у нови ред и #183 за ознаку множења, као и формат '*%-30s*' за лево уређење стринга.

• **Саставити програм који одређује решење система од две линеарне једначине са две непознате.**

Посматраћемо систем од две линеарне једначине са две непознате у општем облику. За налажење решења применићемо методу детерминанти (исте формуле за решење добијају се и применом неке друге методе, али је овако једноставније објаснити поступак решавања система).

Ако је детерминанта система различита од нуле ( $D \neq 0$ ) систем има јединствено решење:  $x = Dx / D$  и  $y = Dy / D$ .

Ако је детерминанта система једнака нули ( $D = 0$ ) и бар једна од детерминанти по  $x$  или по  $y$  је различита од нуле ( $Dx \neq 0$  или  $Dy \neq 0$ )

систем је немогућ, односно, нема решења, у противном систем је неодређен, односно, има бесконачно много решења.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 380; ClientWidth = 400; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, шест лабела за комуникацију са корисником, шест едита за унос, лабелу за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Линеарни систем; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едити за унос:

**Font = Bold, Black; Left = (65, 190, 315); Top = (80, 120); Width = 70.**

Лабела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = 22, Lime; Height = 170; Layout = tlCenter.**

Лабеле за комуникацију са корисником:

**AutoSize = False; Caption = (a1 =, b1 =, c1 =, a2 =, b2 =, c2 =); Height = 36; Layout = tlCenter; Left = (20, 145, 270); Top = (80, 120); Width = 40.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (15, 140, 265); Top = 170; Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Sistem*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```

procedure TForm1.Button2Click(Sender: TObject);
begin
Edit1.Clear;Edit1.ReadOnly:=false;
Edit2.Clear;Edit2.ReadOnly:=false;
Edit3.Clear;Edit3.ReadOnly:=false;
Edit4.Clear;Edit4.ReadOnly:=false;
Edit5.Clear;Edit5.ReadOnly:=false;
Edit6.Clear;Edit6.ReadOnly:=false;
Label8.Caption:=' ';
Edit1.SetFocus;
end;

```

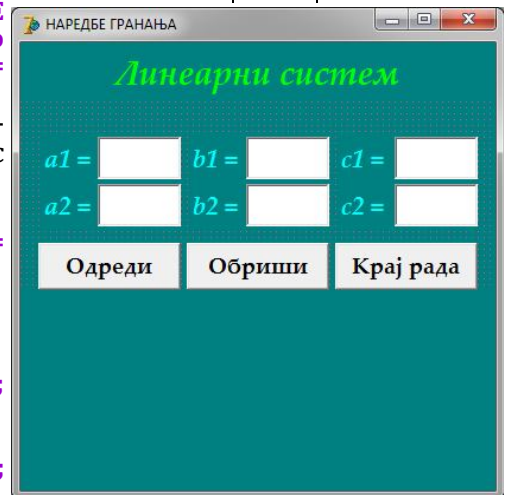
end;

$$\begin{cases} a_1 \cdot x + b_1 \cdot y = c_1 \\ a_2 \cdot x + b_2 \cdot y = c_2 \end{cases}$$

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1 \cdot b_2 - a_2 \cdot b_1$$

$$Dx = \begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix} = c_1 \cdot b_2 - c_2 \cdot b_1$$

$$Dy = \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix} = a_1 \cdot c_2 - a_2 \cdot c_1$$



Двоклик на тастер *Одреди*:

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    a1,b1,c1,a2,b2,c2,d,dx,dy:real;
begin
  a1:=StrToFloatDef(Edit1.Text,1);Edit1.Text:=FloatToStr(a1);
  b1:=StrToFloatDef(Edit2.Text,2);Edit2.Text:=FloatToStr(b1);
  c1:=StrToFloatDef(Edit3.Text,3);Edit3.Text:=FloatToStr(c1);
  a2:=StrToFloatDef(Edit4.Text,4);Edit4.Text:=FloatToStr(a2);
  b2:=StrToFloatDef(Edit5.Text,5);Edit5.Text:=FloatToStr(b2);
  c2:=StrToFloatDef(Edit6.Text,6);Edit6.Text:=FloatToStr(c2);
  s:=Format('%-27s'#13'%1.2f'#183'?', ['Линеарни систем',a1]);
  If b1>=0 then s:=s+'+';
  s:=s+Format('%1.2f'#183'y=%1.2f'#13,[b1,c1]);
  s:=s+Format('%1.2f'#183'?', [a2]);
  If b2>=0 then s:=s+'+';
  s:=s+Format('%1.2f'#183'y=%1.2f'#13,[b2,c2]);
  d:=a1*b2-a2*b1;
  dx:=c1*b2-c2*b1;
  dy:=a1*c2-a2*c1;
  If d=0
  then If (dx=0)and(dy=0)
        then s:=s+' је неодређен.'
        else s:=s+' је немогућ.'
  else s:=s+Format('%-27s'#13'?=%-11.2f'#13'y=%-11.2f', ['има решење:',dx/d,dy/d]);
  Label8.Caption:=s;
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
  Edit4.ReadOnly:=true;Edit5.ReadOnly:=true;Edit6.ReadOnly:=true;
  Button2.SetFocus;
end;

```

- **Саставити програм који одређује решење система од три линеарне једначине са три непознате.**

Посматраћемо систем од три линеарне једначине са три непознате у општем облику:

За налажење решења система применићемо методу детерминанти. Одредићемо детерминанту система и детерминанте променљивих. Ако је детерминанта система различита од нуле ( $D \neq 0$ ) систем има јединствено решење.

Ако је детерминанта система једнака нули ( $D = 0$ ) и бар једна од детерминанти  $D_x$ ,  $D_y$  и  $D_z$  је различита од нуле ( $D_x \neq 0$ ,  $D_y \neq 0$  или  $D_z \neq 0$ ) систем је немогућ.

Ако је детерминанта система једнака нули ( $D = 0$ ) и ако су све детерминанте променљивих једнаке нули ( $D_x = 0$ ,  $D_y = 0$  и  $D_z = 0$ ) онда се мора применити метода елиминације (Гаусова или нека друга) да би смо утврдили да ли систем има бесконачно решења или нема решења. Овај део анализе је (у овом тренутку за нас) мало теже испрограмирати, па ћемо у том случају само исписати поруку да систем нема једнозначно решење, односно, да је можда неодређен или немогућ.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 380; ClientWidth = 400; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лателу за наслов, дванаест латела за комуникацију са корисником, дванаест едита за унос, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Линеарни систем; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едити за унос:

**Font = Bold, Black; Left = (55, 160, 265, 370); Top = (80, 120, 160); Width = 50.**

Латела за испис резултата:

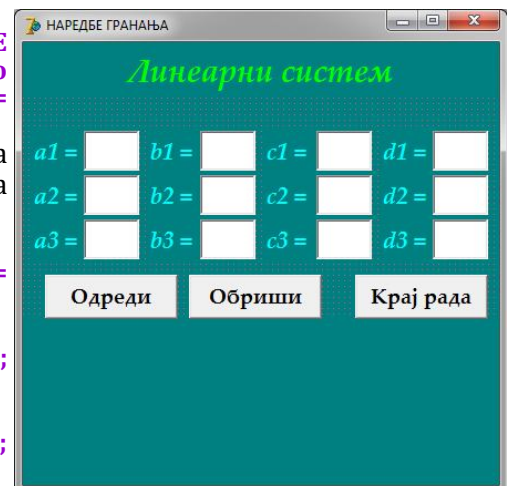
**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 150; Layout = tlCenter.**

Лателе за комуникацију са корисником:

**AutoSize = False; Caption = (a1 =, b1 =, c1 =, d1 =, a2 =, b2 =, c2 =, d2 =, a3 =, b3 =, c3 =, d3); Height = 36; Layout = tlCenter; Left = (10, 115, 220, 315); Top = (80, 120, 160); Width = 45.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (15, 140, 265); Top = 170; Width = 120.**





Сачуваћемо дизајнирану форму у креирани фолдер *Sistem*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit2.Clear;Edit3.Clear;
  Edit4.Clear;Edit5.Clear;Edit6.Clear;
  Edit7.Clear;Edit8.Clear;Edit9.Clear;
  Edit10.Clear;Edit11.Clear;Edit12.Clear;
  Label14.Caption:='';
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    a1,b1,c1,d1,a2,b2,c2,d2,a3,b3,c3,d3,d,dx,dy,dz:real;
begin
  a1:=StrToFloatDef(Edit1.Text,1); Edit1.Text:=FloatToStr(a1);
  b1:=StrToFloatDef(Edit2.Text,2); Edit2.Text:=FloatToStr(b1);
  c1:=StrToFloatDef(Edit3.Text,3); Edit3.Text:=FloatToStr(c1);
  d1:=StrToFloatDef(Edit4.Text,4); Edit4.Text:=FloatToStr(d1);
  a2:=StrToFloatDef(Edit5.Text,5); Edit5.Text:=FloatToStr(a2);
  b2:=StrToFloatDef(Edit6.Text,6); Edit6.Text:=FloatToStr(b2);
  c2:=StrToFloatDef(Edit7.Text,7); Edit7.Text:=FloatToStr(c2);
  d2:=StrToFloatDef(Edit8.Text,8); Edit8.Text:=FloatToStr(d2);
  a3:=StrToFloatDef(Edit9.Text,9); Edit9.Text:=FloatToStr(a3);
  b3:=StrToFloatDef(Edit10.Text,1);Edit10.Text:=FloatToStr(b3);
  c3:=StrToFloatDef(Edit11.Text,2);Edit11.Text:=FloatToStr(c3);
  d3:=StrToFloatDef(Edit12.Text,3);Edit12.Text:=FloatToStr(d3);
  s:=Format('%-30s'+#13+'%1.2f', ['Линеарни систем',a1])+#183+'x';
  If b1>=0 then s:=s+'+';
  s:=s+Format('%1.2f'#183'y', [b1]);
  If c1>=0 then s:=s+'+';
  s:=s+Format('%1.2f'#183'z=%1.2f'#13'%1.2f'#183'x', [c1,d1,a2]);
  If b2>=0 then s:=s+'+';
  s:=s+Format('%1.2f'#183'y', [b2]);
  If c2>=0 then s:=s+'+';
  s:=s+Format('%1.2f'#183'z=%1.2f'#13'%1.2f'#183'x', [c2,d2,a3]);
  If b3>=0 then s:=s+'+';
  s:=s+Format('%1.2f'#183'y', [b3]);
  If c3>=0 then s:=s+'+';
  s:=s+Format('%1.2f'#183'z=%1.2f'#13', [c3,d3]);
  d:=a1*b2*c3-a1*b3*c2-a2*b1*c3+a2*b3*c1+a3*b1*c2-a3*b2*c1;
  dx:=d1*b2*c3-d1*b3*c2-d2*b1*c3+d2*b3*c1+d3*b1*c2-d3*b2*c1;
  dy:=a1*d2*c3-a1*d3*c2-a2*d1*c3+a2*d3*c1+a3*d1*c2-a3*d2*c1;
  dz:=a1*b2*d3-a1*b3*d2-a2*b1*d3+a2*b3*d1+a3*b1*d2-a3*b2*d1;
  If d=0
  then If (dx<>0)or(dy<>0)or(dz<>0)
        then s:=s+'је немогућ.'
        else s:=s+'је неодређен или немогућ.'
  else s:=s+Format('%-30s'#13'x=%1.2f'#13'y=%1.2f'#13'z=%1.2f',
    ['има решење',dx/d,dy/d,dz/d]);
  Label3.Caption:=s;
end;
```

### • Саставити програм који одређује решење квадратне једначине са једном непознатом.

Одредићемо дискриминанту и на основу њеног знака одредити врсту решења, а затим ћемо одредити решење квадратне једначине. Комплексна решења ћемо исписати као збир реалног и имагинарног дела. Подесићемо карактеристике форме на следећи начин:

**BorderIcons = [];** **BorderStyle = bsSingle;** **Caption = НАРЕДБЕ ГРАНАЊА;** **ClientHeight = 330;** **ClientWidth = 380;** **Font = Palatino Linotype, BoldItalic, 16, Aqua;** **FormStyle = fsStayOnTop;** **Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, три лабеле за комуникацију са корисником, три едита за унос, лабелу за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop;** **Alignment = taCenter;** **AutoSize = False;** **Caption = Квадратна једначина;** **Font = 22, Lime;** **Height = 50;** **Layout = tlCenter.**

Едити за унос:

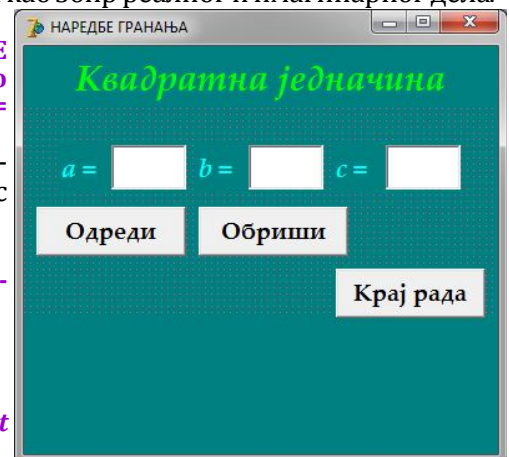
**Font = Bold, Black;** **Left = (70, 180, 290);** **Top = 80;** **Width = 50.**

Лабела за испис резултата:

**Align = alTop;** **Alignment = taCenter;** **AutoSize = False;** **Caption = ;** **Font = Lime;** **Height = 110;** **Layout = tlCenter.**

Лабеле за комуникацију са корисником:

**AutoSize = False;** **Caption = (a =, b =, c =);** **Height = 36;** **Layout = tlCenter;** **Left = (30, 140, 250);** **Top = 80;** **Width = 40.**



Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (10, 140, 250); Top = (130, 130, 180); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Kvadratna*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Label5.Caption:=' ';
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    a,b,c,d,x:real;
begin
  a:=StrToFloatDef(Edit1.Text,1); Edit1.Text:=FloatToStr(a);
  b:=StrToFloatDef(Edit2.Text,2); Edit2.Text:=FloatToStr(b);
  c:=StrToFloatDef(Edit3.Text,1); Edit3.Text:=FloatToStr(c);
  s:=Format('%-27s'#13'%1.2f'#183'x2', ['Решења једначине',a]);
  If b>=0 then s:=s+'+' ;
  s:=s+Format('%1.2f'#183'x', [b]);
  If c>=0 then s:=s+'+' ;
  s:=s+Format('%1.2f = 0'#13,[c]);
  d:=b*b-4*a*c; x:=Sqrt(Abs(d));
  If d=0
  then s:=s+Format('%-27s'#13'%-10s%1.2f', ['су реална и једнака:', 'x1 = x2 =', -b/2/a])
  else If d>0
    then s:=s+'су реална и различита: '+#13+' x1 = '
      +Format('%1.2f'#13' x2 = %1.2f', [(-b+x)/2/a, (-b-x)/2/a])
    else s:=s+'су коњуговано комплексна: '+#13+' x1 = '
      +Format('%1.2f+%1.2f'#183'i'#13' x2 = ', [-b/2/a,x/2/a])
      +Format('%1.2f-%1.2f'#183'i', [-b/2/a,x/2/a]);
  Label5.Caption:=s;
  Edit1.ReadOnly:=true;
  Edit2.ReadOnly:=true;
  Edit3.ReadOnly:=true;
  Button2.SetFocus;
end;
```

### • Саставити програм који одређује квадрант у коме се налази дата тачка у равни.

Раван је двема ортогоналним правима подељена на четири дела, квадранта. У зависности од знака координата тачке пронаћи ћемо њен положај.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 380; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лателу за наслов, две лателе за комуникацију са корисником, два едита за унос, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Координатна раван; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едита за унос:

**Font = Bold, Black; Left = 70; Top = (80, 130); Width = 50.**

Латела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.**

Лателе за комуникацију са корисником:

**AutoSize = False; Caption = (x =, y =); Height = 36; Layout = tlCenter; Left = 30; Top = (80, 130); Width = 40.**

Тастери за управљање:

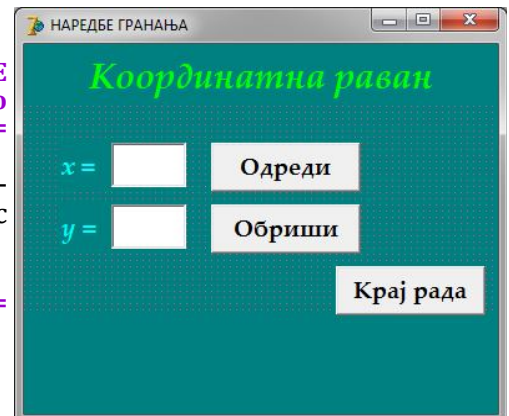
**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (150, 150, 250); Top = (80, 130, 180); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Kvadranti*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Label3.Caption:=' ';
  Edit1.SetFocus;
end;
```





Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    x,y:real;
begin x:=StrToFloatDef(Edit1.Text,1);Edit1.Text:=FloatToStr(x);
      y:=StrToFloatDef(Edit2.Text,2);Edit2.Text:=FloatToStr(y);
      s:=Format('%1s (%1.2f;%1.2f) '#13'%11s', ['Тачка А',x,y, 'налази се ']);
      If (x=0)and(y<>0)
        then s:=s+'на x-оси.'
        else If (x<>0)and(y=0)
          then s:=s+'на y-оси.'
          else If (x=0)and(y=0)
            then s:=s+'у координатном почетку.'
            else If y>0
              then If x>0
                then s:=s+'у првом квадранту.'
                else s:=s+'у другом квадранту.'
              else If x<0
                then s:=s+'у четвртном квадранту.'
                else s:=s+'у трећем квадранту.';

      Label4.Caption:=s;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

• **Саставити програм који одређује октант у коме се налази дата тачка у простору.**

Простор је трима ортогоналним равнима подељен на осам делова, октанта. У пресеку сваке две равни налазе се координатне осе. У зависности од знака координата тачке одредићемо и записати њен положај у координатном простору.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 380; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, три лабеле за комуникацију са корисником, три едита за унос, лабелу за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Координатни простор; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едити за унос:

**Font = Bold, Black; Left = 70; Top = (80, 130, 180); Width = 50.**

Лабела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.**

Лабеле за комуникацију са корисником:

**AutoSize = False; Caption = (x=, y=, z=); Height = 36; Layout = tlCenter; Left = 30; Top = (80, 130, 180); Width = 40.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (150, 150, 250); Top = (80, 130, 180); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Oktanti*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

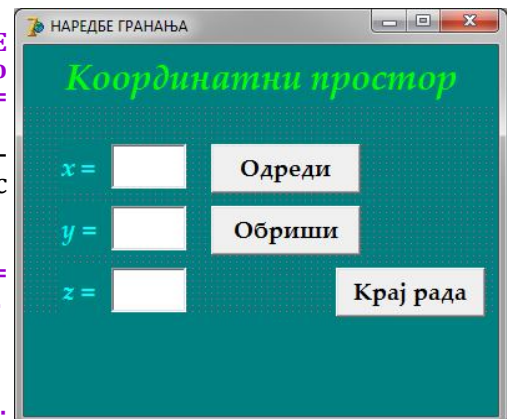
Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit3.ReadOnly:=false;
      Label5.Caption:='';
      Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    x,y,z:real;
begin x:=StrToFloatDef(Edit1.Text,1); Edit1.Text:=FloatToStr(x);
      y:=StrToFloatDef(Edit2.Text,2); Edit2.Text:=FloatToStr(y);
      z:=StrToFloatDef(Edit3.Text,3); Edit3.Text:=FloatToStr(z);
      a:=Format('Тачка А(%1.2f;%1.2f;%1.2f) '#13'налази се ', [x,y,z]);
      If x=0
        then If y=0
          then If z=0
            then a:=a+'у координатном почетку.'
            else a:=a+'на z-оси.'
```



```

else If z=0
    then a:=a+'на у-оси.'
    else a:=a+'у у-z равни.'
else If y=0
    then If z=0
        then a:=a+'на х-оси.'
        else a:=a+'у х-z равни.'
    else If z=0
        then a:=a+'у х-у равни.'
        else If z>0
            then If x>0
                then If y>0
                    then a:=a+'у првом октанту.'
                    else a:=a+'у другом октанту.'
                else If y>0
                    then a:=a+'у четвртом октанту.'
                    else a:=a+'у трећем октанту.'
            else If x>0
                then If y>0
                    then a:=a+'у петом октанту.'
                    else a:=a+'у шестом октанту.'
                else If y>0
                    then a:=a+'у осмом октанту.'
                    else a:=a+'у седмом октанту.';
Label5.Caption:=s;
Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
Button2.SetFocus;
end;

```

• **Саставити програм који три унета броја уређује у неоппадајући (нерастући) поредак.**

Да бисмо решили овај задатак одредићемо најпре најмањи од унетих бројева, а затим ћемо преостала два уредити по величини. Идеја је да најмањи број буде први, зато ћемо упоређивати први и други, па, ако је први већи од другог, заменићемо им места. Затим ћемо упоређивати први и трећи и, ако је први већи, заменићемо им место. Сада је први број са најмањи (јер је мањи и од другог и од трећег), па сада треба упоређивати други и трећи број и, ако је други већи од трећег, заменићемо им места. За размену вредности две променљиве користићемо помоћну променљиву која ће привремено чувати вредност прве променљиве да би она могла преузети вредност друге, а да се при томе не изгуби њена првобитна вредност. То значи да се замена вредности две променљиве обавља кроз три наредбе: (помоћна променљива узима вредност прве променљиве)  $p:=a$ ; (прва променљива узима вредност друге променљиве)  $a:=b$ ; (друга променљива узима вредност помоћне променљиве, тј. претходну вредност прве променљиве)  $b:=p$ ; Ово је стандардна процедура замене вредности две променљиве.

Подесићемо карактеристике форме на следећи начин:

```

BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ
ГРАНАЊА; ClientHeight = 300; ClientWidth = 380; Font = Palatino
Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position =
poScreenCenter.

```

На форму ћемо ставити лателу за наслов, три лателе за комуникацију са корисником, три едита за унос, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

```

Align = alTop; Alignment = taCenter; AutoSize = False; Caption =
Сортирање; Font = 22, Lime; Height = 50; Layout = tlCenter.

```

Едити за унос:

```

Font = Bold, Black; Left = 70; Top = (80, 130, 180); Width = 50.

```

Латела за испис резултата:

```

Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ;
Font = Lime; Height = 70; Layout = tlCenter.

```

Лателе за комуникацију са корисником:

```

AutoSize = False; Caption = (a =, b =, c =); Height = 36; Layout = tlCenter; Left = 30; Top = (80, 130, 180); Width
= 40.

```

Тастери за управљање:

```

Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (150, 150, 250); Top = (80, 130,
180); Width = 120.

```

Сачуваћемо дизајнирану форму у креирани фолдер *Rastuci*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

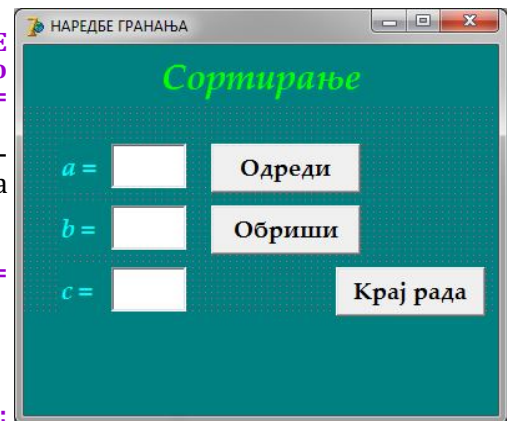
Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```

procedure TForm1.Button2Click(Sender: TObject);
begin
    Edit1.Clear;Edit1.ReadOnly:=false;
    Edit2.Clear;Edit2.ReadOnly:=false;
    Edit3.Clear;Edit3.ReadOnly:=false;
    Label5.Caption:='';
    Edit1.SetFocus;
end;

```



Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    a,b,c,p:integer;
begin a:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,2);Edit2.Text:=IntToStr(b);
      c:=StrToIntDef(Edit3.Text,3);Edit3.Text:=IntToStr(c);
      s:='Неоподајући поредак'+#13;
      If a>b
        then begin p:=a;a:=b;b:=p;
              end;
      If a>c
        then begin p:=a;a:=c;c:=p;
              end;
      If b>c
        then begin p:=b;b:=c;c:=p;
              end;
      Label5.Caption:=s+Format('%2d, %2d, %2d',[a,b,c]);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Ако желимо да бројеве уредимо у нерастући поредак потребно је неједнакост  $>$  у све три if наредбе променити у  $<$ , или код исписа резултата наредбу *Format* написати овако *Format('%2d, %2d, %2d',[c,b,a])*.

- **Саставити програм који одређује дужину трајања емисије у минутима, ако је дато време почетка и краја емисије у сатима и минутима.**

Да би се задатак коректно решио требало би увести контролу уноса времена и водити рачуна о преласку у нови дан. Време почетка и краја емисије претворићемо у минуте и одузети да бисмо добили решење.

Подесићемо карактеристике форме на следећи начин:

*BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 330; ClientWidth = 360; Color = Teal; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.*

На форму ћемо ставити лабелу за наслов, три лабеле за комуникацију са корисником, четири едита за унос, едит за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Сабирање времена; Font = 22, Lime; Height = 50; Layout = tlCenter.*

Едити за унос:

*Font = Bold, Black; Left = (230, 280); MaxLength = 2; Top = (80, 130); Width = 40.*

Едит за испис резултата:

*Font = Bold, Black; Left = 230; ReadOnly = True; Top = 180; Width = 90.*

Лабеле за комуникацију са корисником:

*AutoSize = False; Caption = (Почетак емисије ; Завршетак емисије ; Трајање емисије :); Height = 36; Layout = tlCenter; Left = 60; Top = (100, 150, 200); Width = 210.*

Тастери за управљање:

*Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (20, 150, 230); Top = (230, 230, 280); Width = 120.*

Сачуваћемо дизајнирану форму у креирани фолдер *Emisija*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

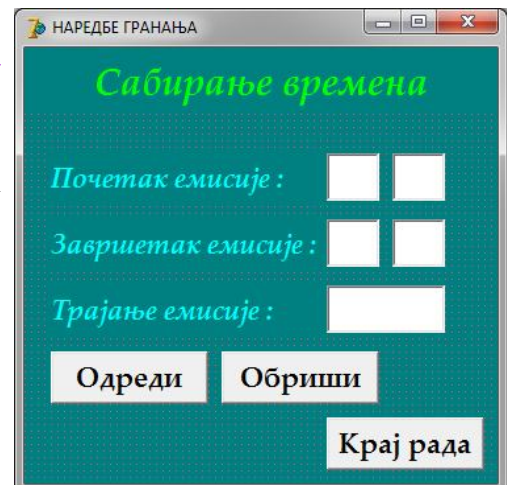
Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit3.ReadOnly:=false;
      Edit4.Clear;Edit4.ReadOnly:=false;
      Edit5.Clear;
      Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
begin If sp>sk
      then sk:=sk+24;
      t:=sk*60+mk-(sp*60+mp);Edit5.Text:=IntToStr(t);
      Edit1.ReadOnly:=true;Edit1.ReadOnly:=true;
      Edit2.ReadOnly:=true;Edit2.ReadOnly:=true;
      Edit3.ReadOnly:=true;Edit3.ReadOnly:=true;
      Edit4.ReadOnly:=true;Edit4.ReadOnly:=true;
      Button2.SetFocus;
end;
```



Све променљиве које смо употребили декларисане су као глобалне у заглављу програма:

```
var
  Form1: TForm1;
  sp, mp, sk, mk, t: integer;
```

Контролу уноса времена остварили смо у процедурама које су реакција на излазак из одговарајућег едита (селекујемо сваки едит, у прозору *ObjectInspector* на листићу *Events* двоклик на поље вредности догађаја *OnExit* декларише одговарајућу процедуру коју дефинишемо):

```
procedure TForm1.Edit1Exit(Sender: TObject);
begin sp:=StrToIntDef(Edit1.Text,0);
  If (sp<0) or (sp>23)
  then begin Edit1.Clear;Edit1.SetFocus;
  end
  else Edit1.Text:=IntToStr(sp);
end;
procedure TForm1.Edit2Exit(Sender: TObject);
begin mp:=StrToIntDef(Edit2.Text,0);
  If (mp<0) or (mp>59)
  then begin Edit2.Clear;Edit2.SetFocus;
  end
  else Edit2.Text:=IntToStr(mp);
end;
procedure TForm1.Edit3Exit(Sender: TObject);
begin sk:=StrToIntDef(Edit3.Text,0);
  If (sk<0) or (sk>23)
  then begin Edit3.Clear;Edit3.SetFocus;
  end
  else Edit3.Text:=IntToStr(sk);
end;
procedure TForm1.Edit4Exit(Sender: TObject);
begin mk:=StrToIntDef(Edit4.Text,0);
  If (mk<0) or (mk>59) or (sp=sk) and (mp>mk)
  then begin Edit4.Clear;Edit4.SetFocus;
  end
  else Edit4.Text:=IntToStr(mk);
end;
```

Уведене контроле нису толерантне. Податак који се уноси мора бити логично исправан у односу на претходно унете, чак и ако је претходни податак погрешно унет. Тек када се унесе исправан податак може се променити претходни.

- **Саставити програм који одређује комплемент и суплемент угла датог у степенима минутима и секундама.**

Да би се коректно решио задатак потребно је увести контролу уноса угла (мора бити мањи или једнак 90 степени да би се могао наћи комплемент). Угао ћемо претворити у секунде, а затим ћемо комплемент добити као допуну до  $90 \cdot 3600$  секунди. Суплемент је за 90 степени већи.

Подесићемо карактеристике форме на следећи начин:

```
BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ
ГРАНАЊА; ClientHeight = 330; ClientWidth = 370; Color = Teal; Font
= Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop;
Position = poScreenCenter.
```

На форму ћемо ставити лателу за наслов, три лателе за комуникацију са корисником, три едита за унос, шест едита за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

```
Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Комп
лемент и суплемент; Font = 22, Lime; Height = 50; Layout = tlCenter.
```

Едита за унос:

```
Font = Bold, Black; Left = (170, 230, 270); MaxLength = 2; Top = 80;
Width = (50, 40, 40).
```

Едит за испис резултата:

```
Font = Bold, Black; Left = (170, 230, 270); ReadOnly = True; Top =
(130, 180); Width = (50, 40, 40).
```

Лателе за комуникацију са корисником:

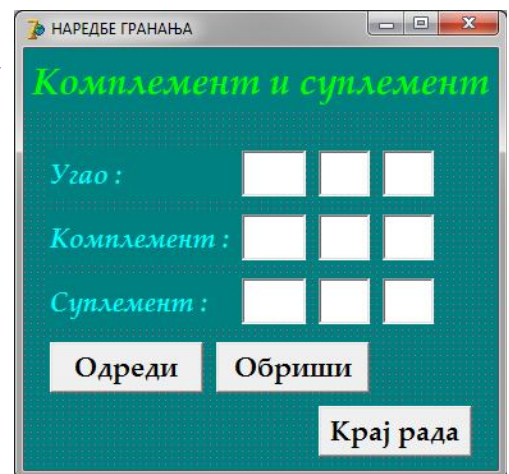
```
AutoSize = False; Caption = (Угао :, Комплемент :, Суплемент :); Height = 36; Layout = tlCenter; Left = 20;
Top = (80, 130, 180); Width = 150.
```

Тастери за управљање:

```
Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (20, 150, 230); Top = (230, 230,
280); Width = 120.
```

Сачуваћемо дизајнирану форму у креирани фолдер *КомпСур*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.





Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Edit4.Clear;Edit5.Clear;Edit6.Clear;
  Edit7.Clear;Edit8.Clear;Edit9.Clear;
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var c1,c2,c3:integer;
begin
  c3:=90*3600-(a1*3600+a2*60+a3);
  c1:=c3 div 3600;
  c2:=c3 div 60 mod 60;
  c3:=c3 mod 60;
  Edit4.Text:=IntToStr(c1);
  Edit5.Text:=IntToStr(c2);
  Edit6.Text:=IntToStr(c3);
  Edit7.Text:=IntToStr(c1+90);
  Edit8.Text:=IntToStr(c2);
  Edit9.Text:=IntToStr(c3);
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
  Button2.SetFocus;
end;
```

Променљиве за унос угла декларисане су као глобалне у заглављу програма:

```
var
  Form1: TForm1;
  a1,a2,a3:integer;
```

Контролу уноса угла остварили смо у процедурама које су реакција на излазак из одговарајућег едита (селекујемо сваки едит, у прозору *ObjectInspector* на листићу *Events* двоклик на поље вредности догађаја *OnExit* декларише одговарајућу процедуру коју дефинишемо):

```
procedure TForm1.Edit1Exit(Sender: TObject);
begin
  a1:=StrToIntDef(Edit1.Text,0);
  If (a1<0) or (a1>90)
  then begin Edit1.Clear;Edit1.SetFocus;
  end
  else Edit1.Text:=IntToStr(a1);
end;

procedure TForm1.Edit2Exit(Sender: TObject);
begin
  a2:=StrToIntDef(Edit2.Text,0);
  If (a2<0) or (a2>59) or (a1=90) and (a2>0)
  then begin Edit2.Clear;Edit2.SetFocus;
  end
  else Edit2.Text:=IntToStr(a2);
end;

procedure TForm1.Edit3Exit(Sender: TObject);
begin
  a3:=StrToIntDef(Edit3.Text,0);
  If (a3<0) or (a3>59) or (a1=90) and (a3>0)
  then begin Edit3.Clear;Edit3.SetFocus;
  end
  else Edit3.Text:=IntToStr(a3);
end;
```

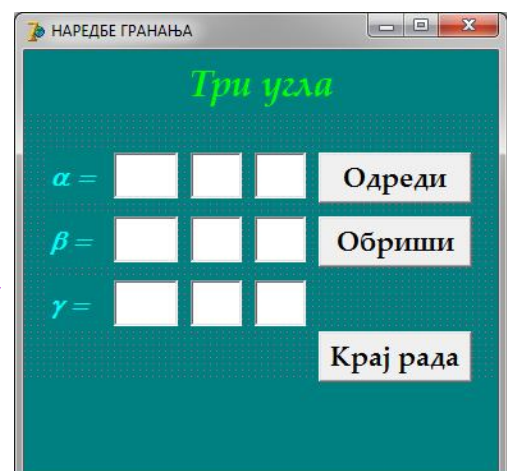
- **Дата су три угла у степенима минутима и секундама. Саставити програм који одређује да ли они могу бити углови троугла и ако могу да ли је троугао оштроугли, правоугли или тупоугли.**

Први услов који три угла морају испуњавати је да им је збир 180 степени, али то није довољно. Да би био оштроугли троугао сва три угла морају бити оштра. Да би троугао био правоугли један угао мора бити прав, а остала два оштра. Да би троугао био тупоугли један угао мора бити туп, а остала два оштра. Код уноса углова није потребно уводити претходну контролу уноса јер је садржана у каснијим условима.

Подесићемо карактеристике форме на следећи начин:

```
BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ
ГРАНАЊА; ClientHeight = 330; ClientWidth = 370; Color = Teal; Font
= Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop;
Position = poScreenCenter.
```

На форму ћемо ставити лателу за наслов, три лателе за комуникацију са корисником, девет едита за унос, лателу за испис резултата и три тастера за управљање апликацијом.





Лабела за наслов:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Три угла; Font = 22, Lime; Height = 50; Layout = tlCenter.*

Едити за унос:

*Font = Bold, Black; Left = (170, 230, 270); MaxLength = (3, 2, 2); Top = (80, 130, 180); Width = (50, 40, 40).*

Лабела за испис резултата:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.*

Лабеле за комуникацију са корисником:

*AutoSize = False; Caption = (a =, b =, g =); Font = Symbol; Height = 36; Layout = tlCenter; Left = 20; Top = (80, 130, 180); Width = 50.*

Тастери за управљање:

*Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 230; Top = (80, 130, 220); Width = 120.*

Сачуваћемо дизајнирану форму у креирани фолдер *Uglovi*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Edit4.Clear;Edit4.ReadOnly:=false;
  Edit5.Clear;Edit5.ReadOnly:=false;
  Edit6.Clear;Edit6.ReadOnly:=false;
  Edit7.Clear;Edit7.ReadOnly:=false;
  Edit8.Clear;Edit8.ReadOnly:=false;
  Edit9.Clear;Edit9.ReadOnly:=false;
  Label5.Caption:='';
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    a1,b1,c1,a2,b2,c2,a3,b3,c3:integer;
begin
  a1:=StrToIntDef(Edit1.Text,60);Edit1.Text:=IntToStr(a1);
  a2:=StrToIntDef(Edit2.Text,0); Edit2.Text:=IntToStr(a2 mod 60);
  a3:=StrToIntDef(Edit3.Text,0); Edit3.Text:=IntToStr(a3 mod 60);
  b1:=StrToIntDef(Edit4.Text,60);Edit4.Text:=IntToStr(b1);
  b2:=StrToIntDef(Edit5.Text,0); Edit5.Text:=IntToStr(b2 mod 60);
  b3:=StrToIntDef(Edit6.Text,0); Edit6.Text:=IntToStr(b3 mod 60);
  c1:=StrToIntDef(Edit7.Text,60);Edit7.Text:=IntToStr(c1);
  c2:=StrToIntDef(Edit8.Text,0); Edit8.Text:=IntToStr(c2 mod 60);
  c3:=StrToIntDef(Edit9.Text,0); Edit9.Text:=IntToStr(c3 mod 60);
  a3:=a1*3600+a2*60+a3;
  b3:=b1*3600+b2*60+b3;
  c3:=c1*3600+c2*60+c3;
  s:='Не могу бити углови троугла.';
  If a3+b3+c3=180*3600
  then If (a3=90*3600) and (b3<90*3600) and (c3<90*3600) xor
        (a3<90*3600) and (b3=90*3600) and (c3<90*3600) xor
        (a3<90*3600) and (b3<90*3600) and (c3=90*3600)
  then s:='Могу бити углови троугла, '+#13+'- правоугли.'
  else If (a3>90*3600) and (b3<90*3600) and (c3<90*3600) xor
        (a3<90*3600) and (b3>90*3600) and (c3<90*3600) xor
        (a3<90*3600) and (b3<90*3600) and (c3>90*3600)
  then s:='Могу бити углови троугла, '+#13+'- тупоугли.'
  else If (a3<90*3600) and (b3<90*3600) and (c3<90*3600)
  then s:='Могу бити углови троугла, '+#13+'- оштроугли.';
  Label5.Caption:=s;
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
  Edit4.ReadOnly:=true;Edit5.ReadOnly:=true;Edit6.ReadOnly:=true;
  Edit7.ReadOnly:=true;Edit8.ReadOnly:=true;Edit9.ReadOnly:=true;
  Button2.SetFocus;
end;
```

- **Дата су два угла у степенима минутима и секундама. Саставити програм који одређује да ли постоји угао који са ова два чини троугао и ако постоји одређује да ли је троугао оштроугли, правоугли или тупоугли.**

Услови су слични као и у претходном задатку, једино што по услову задатка збир углова мора бити 180 степени, па тај услов не морамо проверавати. Овде ће и остали услови бити нешто једноставнији јер ћемо пре одређивања трећег угла проверити да ли је збир прва два мањи од 180 степени и да ли су оба угла позитивно оријентисана.

Подесићемо карактеристике форме на следећи начин:

*BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 330; ClientWidth = 370; Color = Teal; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.*

На форму ћемо ставити лателу за наслов, три лателе за комуникацију са корисником, шест едита за унос, три едита и лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Три угла; Font = Bold, Lime; Height = 50; Layout = tlCenter.*

Едита за унос:

*Font = Bold, Black; Left = (170, 230, 270); MaxLength=(3, 2, 2); Top = (80, 130); Width = (50, 40, 40).*

Едита за испис резултата:

*Font = Bold, Black; Left = (170, 230, 270); ReadOnly = True; Top = 200; Width = (50, 40, 40).*

Латела за испис резултата:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.*

Лателе за комуникацију са корисником:

*AutoSize = False; Caption = (a =, b =, g =); Font = Symbol; Height = 36; Layout = tlCenter; Left = 20; Top = (80, 130, 200); Width = 50.*

Тастери за управљање:

*Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 230; Top = (80, 130, 200); Width = 120.*

Сачуваћемо дизајнирану форму у креирани фолдер *Trougao*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Edit4.Clear;Edit4.ReadOnly:=false;
  Edit5.Clear;Edit5.ReadOnly:=false;
  Edit6.Clear;Edit6.ReadOnly:=false;
  Edit7.Clear;Edit8.Clear;Edit9.Clear;
  Label5.Caption:='';
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

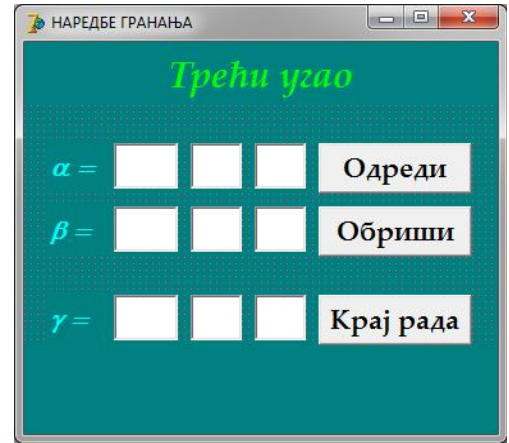
```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    a1,b1,c1,a2,b2,c2,a3,b3,c3:integer;
begin
  a1:=StrToIntDef(Edit1.Text,60);Edit1.Text:=IntToStr(a1);
  a2:=StrToIntDef(Edit2.Text,0);Edit2.Text:=IntToStr(a2 mod 60);
  a3:=StrToIntDef(Edit3.Text,0);Edit3.Text:=IntToStr(a3 mod 60);
  b1:=StrToIntDef(Edit4.Text,60);Edit4.Text:=IntToStr(b1);
  b2:=StrToIntDef(Edit5.Text,0);Edit5.Text:=IntToStr(b2 mod 60);
  b3:=StrToIntDef(Edit6.Text,0);Edit6.Text:=IntToStr(b3 mod 60);
  a3:=a1*3600+a2*60+a3;
  b3:=b1*3600+b2*60+b3;
  If (a3+b3<180*3600) and (a3>0) and (b3>0)
    then begin
      c3:=180*3600-a3-b3;
      If (a3=90*3600) or (b3=90*3600) or (c3=90*3600)
        then s:='Могу бити углови троугла,'#13+'- правоугли.'
        else If (a3>90*3600) or (b3>90*3600) or (c3>90*3600)
          then s:='Могу бити углови троугла,'#13+'- тупоугли.'
          else s:='Могу бити углови троугла,'#13+'- оштроугли.';
      c1:=c3 div 3600; Edit7.Text:=IntToStr(c1);
      c2:=c3 div 60 mod 60;Edit8.Text:=IntToStr(c2);
      c3:=c3 mod 60; Edit9.Text:=IntToStr(c3);
    end
    else begin
      s:='Не могу бити углови троугла.';
      Edit7.Clear;Edit8.Clear;Edit9.Clear;
    end;
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
  Edit4.ReadOnly:=true;Edit5.ReadOnly:=true;Edit6.ReadOnly:=true;
  Label5.Caption:=s;
  Button2.SetFocus;
end;
```

- **Саставити програм који одређује да ли три уписана броја могу бити мерни бројеви страница троугла и ако могу, какав је тај троугао.**

Подесићемо карактеристике форме на следећи начин:

*BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 350; ClientWidth = 350; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.*

На форму ћемо ставити лателу за наслов, три лателе за комуникацију са корисником, три едита за унос, лателу за испис резултата и три тастера за управљање апликацијом.



Лабела за наслов:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Сортирање; Font = 22, Lime; Height = 50; Layout = tlCenter.*

Едити за унос:

*Font = Bold, Black; Left = 80; Top = (80, 130, 180); Width = 100.*

Лабела за испис резултата:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 80; Layout = tlCenter.*

Лабеле за комуникацију са корисником:

*AutoSize = False; Caption = (a =, b =, c =); Height = 36; Layout = tlCenter; Left = 40; Top = (80, 130, 180); Width = 40.*

Тастери за управљање:

*Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 200; Top = (80, 130, 180); Width = 120.*

Сачуваћемо дизајнирану форму у креирани фолдер *Stranice*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Label5.Caption:='';
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    b,c,d,p:real;
begin
  b:=StrToFloatDef(Edit1.Text,1);Edit1.Text:=FloatToStr(b);
  c:=StrToFloatDef(Edit2.Text,3);Edit2.Text:=FloatToStr(c);
  d:=StrToFloatDef(Edit3.Text,2);Edit3.Text:=FloatToStr(d);
  If (b+c>d) and (c+d>b) and (d+b>c)
  then begin If b>c
              then begin p:=c;c:=b;b:=p;
                    end;
              If c>d
              then begin p:=c;c:=d;d:=p;
                    end;
              If d*d<c*c+b*b
              then a:='Ово су странице оштроуглог троугла.'
              else If d*d=c*c+b*b
              then a:='Ово су странице правоуглог троугла.'
              else a:='Ово су странице тупоуглог троугла.';
            end
  else a:='Не могу бити странице троугла.';
  Label3.Caption:=a;
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
  Button2.SetFocus;
end;
```

• **Саставити програм који одређује углове (у степенима, минутима и секундама) троугла ако су дате странице *a*, *b* и *c*.**

Искористићемо косинусну теорему да одредимо два угла јер имамо све странице троугла. Трећи угао ћемо добити као допуну до 180°. Ове једноставне математичке формуле лако можемо написати у духу нашег програмског језика и решити задатак. Треба водити рачуна о томе да све тригонометријске функције раде са угловима у радијанима, па ове формуле треба допунити конверзијом радијана у степене, минуте и секунде.

Подесићемо карактеристике форме на следећи начин:

*BorderIcons = []; BorderStyle = bsSingle; Caption = РЕАЛНИ ТИП; ClientHeight = 370; ClientWidth = 400; Color = Teal; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.*

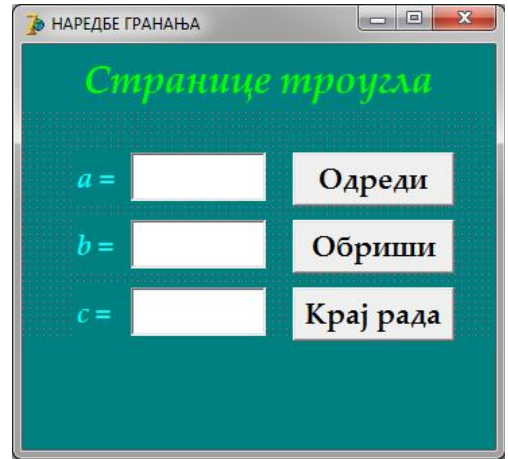
На форму ћемо ставити лабелу за наслов, шест лабела за комуникацију са корисником, три едита за унос, девет едита за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Углови троугла; Font = 22, Lime; Height = 50; Layout = tlCenter.*

Едити за унос:

*Font = Bold, Black; Left = (65, 185, 305); Top = 80; Width = 70.*



$$a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos \alpha$$

$$\cos \alpha = \frac{b^2 + c^2 - a^2}{2 \cdot b \cdot c}$$

$$\sin \alpha = \sqrt{1 - \cos^2 \alpha}$$

$$\alpha = \text{ArcTg}\left(\frac{\sin \alpha}{\cos \alpha}\right)$$

$$b^2 = c^2 + a^2 - 2 \cdot c \cdot a \cdot \cos \beta$$

$$\cos \beta = \frac{c^2 + a^2 - b^2}{2 \cdot c \cdot a}$$

$$\sin \beta = \sqrt{1 - \cos^2 \beta}$$

$$\beta = \text{ArcTg}\left(\frac{\sin \alpha}{\cos \alpha}\right)$$

$$\gamma = 180^\circ - \alpha - \beta$$

Едити за испис резултата:

**Font = Bold, Black; Left = (80, 135, 180); ReadOnly = True; Top = (200, 240, 280); Width = (50, 40, 40).**

Лабеле за комуникацију са корисником:

**AutoSize = False; Caption = (a =, b =, c =); Height = 36; Layout = tlCenter; Left = (30, 150, 270); Top = 80; Width = 35.**

Лабеле за угао:

**AutoSize = False; Caption = (a =, b =, g =); Font = Symbol, BoldItalic, 16, Aqua; Height = 36; Layout = tlCenter; Left = 30; Top = (200, 240, 280); Width = 50.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (65, 200, 260); Top = (140, 140, 310); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *SveStranice*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Edit4.Clear;Edit5.Clear;Edit6.Clear;
  Edit7.Clear;Edit8.Clear;Edit9.Clear;
  Edit10.Clear;Edit11.Clear;Edit12.Clear;
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,p:real;
    a1,a2,a3,b1,b2,b3,c1,c2,c3:integer;
begin
  a:=StrToFloatDef(Edit1.Text,3);Edit1.Text:=FloatToStr(a);
  b:=StrToFloatDef(Edit2.Text,3);Edit2.Text:=FloatToStr(b);
  c:=StrToFloatDef(Edit3.Text,3);Edit3.Text:=FloatToStr(c);
  If (a+b>c) and (b+c>a) and (c+a>b)
  then begin p:=(b*b+c*c-a*a)/(2*b*c);
    If p=0
    then a3:=0
    else a3:=Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600);
    a1:=a3 div 3600; Edit4.Text:=IntToStr(a1);
    a2:=a3 div 60 mod 60;Edit5.Text:=IntToStr(a2);
    a3:=a3 mod 60; Edit6.Text:=IntToStr(a3);
    p:=(a*a+c*c-b*b)/(2*a*c);
    If p=0
    then b3:=0
    else b3:=Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600);
    b1:=b3 div 3600; Edit7.Text:=IntToStr(b1);
    b2:=b3 div 60 mod 60;Edit8.Text:=IntToStr(b2);
    b3:=b3 mod 60; Edit9.Text:=IntToStr(b3);
    c3:=180*3600-(a1*3600+b1*3600+a2*60+b2*60+a3+b3);
    c1:=c3 div 3600; Edit10.Text:=IntToStr(c1);
    c2:=c3 div 60 mod 60;Edit11.Text:=IntToStr(c2);
    c3:=c3 mod 60; Edit12.Text:=IntToStr(c3);
    Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
    Button2.SetFocus;
  end
  else begin Application.MessageBox('Не могу бити странице троугла!', 'ГРЕШКА У УНОСУ', 0);
    Button2.Click;
  end;
end;
```

- **Саставити програм који одређује трећу страницу, два висине и углове троугла ако су дате две странице и висина на једну од њих.**

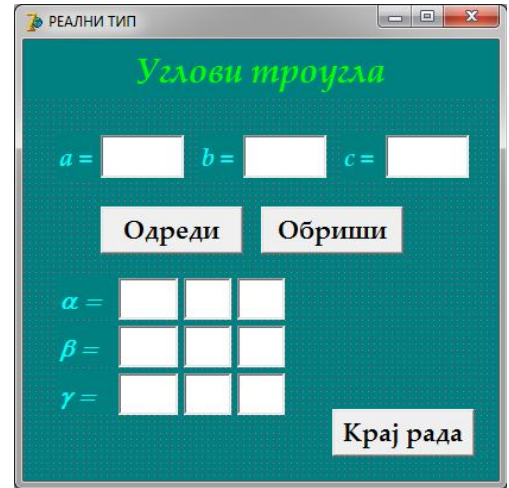
Применићемо Питагорину теорему за два правоугла троугла на које почетни троугао дели висина и добићемо трећу страницу. Одредићемо висине троугла из обрасца за површину, а затим и обим и површину троугла. За углове ћемо прво одредити *sin* и *cos*, а затим помоћу *ArcTan* углове у секундама. Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = РЕАЛНИ ТИП; ClientHeight = 370; ClientWidth = 440; Color = Teal; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, четрнаест лабела за комуникацију са корисником, три едита за унос, седамнаест едита за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Решавање троугла; Font = 22, Lime; Height = 50; Layout = tlCenter.**





Едити за унос:

**Font = Bold, Black; Left = 75; Top = (70, 110, 150); Width = 70.**

Едити за испис страница и висина:

**Font = Bold, Black; Left = (215, 345); ReadOnly = True; Top = (70, 110, 150); Width = 70.**

Едити за испис обима и површине:

**Font = Bold, Black; Left = (215, 335); ReadOnly = True; Top = 190; Width = 80.**

Едити за испис углова:

**Font = Bold, Black; Left = (215, 270, 315); ReadOnly = True; Top = (230, 270, 310); Width = (50, 40, 40).**

Лабеле за комуникацију са корисником:

**AutoSize = False; Caption = (a =, b =, ha =, a =, b =, c =, ha =, hb =, hc =, O =, P =); Height = 36; Layout = tlCenter; Left = (75, 150, 270); Top = (70, 110, 150, 70, 110, 150, 70, 110, 150, 190, 190); Width = 35.**

Лабеле углова:

**AutoSize = False; Caption = (a =, b =, g =); Font = Symbol, BoldItalic, 16, Aqua; Height = 36; Layout = tlCenter; Left = 180; Top = (230, 270, 310); Width = 35.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 30; Top = (205, 255, 305); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *DveStranice*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

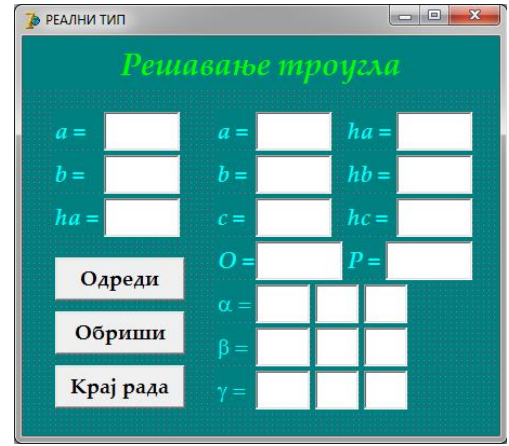
Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Edit4.Clear;Edit5.Clear;Edit6.Clear;Edit7.Clear;Edit8.Clear;
  Edit9.Clear; Edit10.Clear;Edit11.Clear;Edit12.Clear;
  Edit13.Clear;Edit14.Clear;Edit15.Clear;Edit16.Clear;
  Edit17.Clear;Edit18.Clear;Edit19.Clear;Edit20.Clear;
  Edit1.SetFocus;;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,h1,h2,h3,p,o,x,y:real;
    a1,a2,a3,b1,b2,b3,c1,c2,c3:integer;
begin
  a:=StrToFloatDef(Edit1.Text,6);Edit1.Text:=FloatToStr(a);
  b:=StrToFloatDef(Edit2.Text,6);Edit2.Text:=FloatToStr(b);
  h1:=StrToFloatDef(Edit3.Text,3);Edit3.Text:=FloatToStr(h1);
  If b<h1 // страница b је хипотенуза у помоћном троуглу и мора бити већа од катета
  then begin Application.MessageBox('Не може бити троугао!', 'ГРЕШКА У УНОСУ',0);
        Button2.Click;
      end
  else begin x:=Sqrt(b*b-h1*h1); // одсечци на које висина дели страницу a
            y:=a-x;
            Edit4.Text:=Format('%7.3f', [a]);
            Edit5.Text:=Format('%7.3f', [b]);
            c:=Sqrt(y*y+h1*h1);Edit6.Text:=Format('%7.3f', [c]);
            Edit7.Text:=Format('%7.3f', [h1]);
            h2:=a*h1/b;Edit8.Text:=Format('%7.3f', [h2]);
            h3:=a*h1/c;Edit9.Text:=Format('%7.3f', [h3]);
            o:=a+b+c;Edit10.Text:=Format('%7.3f', [o]);
            p:=a*h1/2;Edit11.Text:=Format('%7.3f', [p]);
            If y=0 // странице a и b су нормалне, угао γ је прав
            then b3:=90*3600
            else b3:=Round(ArcTan(h1/y)/pi*180*3600);
            If x=0 // странице a и c су нормалне, угао β је прав
            then c3:=90*3600
            else c3:=Round(ArcTan(h1/x)/pi*180*3600);
            a3:=180*3600-b3-c3;
            a1:=a3 div 3600; Edit12.Text :=IntToStr(a1);
            a2:=a3 div 60 mod 60;Edit13.Text:=IntToStr(a2);
            a3:=a3 mod 60; Edit14.Text:=IntToStr(a3);
            b1:=b3 div 3600; Edit15.Text:=IntToStr(b1);
            b2:=b3 div 60 mod 60;Edit16.Text:=IntToStr(b2);
            b3:=b3 mod 60; Edit17.Text:=IntToStr(b3);
            c1:=c3 div 3600; Edit18.Text:=IntToStr(c1);
            c2:=c3 div 60 mod 60;Edit19.Text:=IntToStr(c2);
            c3:=c3 mod 60; Edit20.Text:=IntToStr(c3);
            Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
            Button2.SetFocus;
          end;
end;
```





- Саставити програм који одређује две странице, висину и углове троугла ако су дате две висине и трећа страница троугла

Подесићемо карактеристике форме на следећи начин:

```
BorderIcons = []; BorderStyle = bsSingle; Caption = РЕАЛНИ ТИП;
ClientHeight = 370; ClientWidth = 440; Color = Teal; Font = Palatino
Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position =
poScreenCenter.
```

На форму ћемо ставити лателу за наслов, четрнаест латела за комуникацију са корисником, три едита за унос, седамнаест едита за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

```
Align = alTop; Alignment = taCenter; AutoSize = False; Caption =
Решавање троугла; Font = 22, Lime; Height = 50; Layout = tlCenter.
```

Едита за унос:

```
Font = Bold, Black; Left = 75; Top = (70, 110, 150); Width = 70.
```

Едита за испис страница и висина:

```
Font = Bold, Black; Left = (215, 345); ReadOnly = True; Top = (70,
110, 150); Width = 70.
```

Едита за испис обима и површине:

```
Font = Bold, Black; Left = (215, 335); ReadOnly = True; Top = 190; Width = 80.
```

Едита за испис углава:

```
Font = Bold, Black; Left = (215, 270, 315); ReadOnly = True; Top = (230, 270, 310); Width = (50, 40, 40).
```

Лателе за комуникацију са корисником:

```
AutoSize = False; Caption = (a =, hb =, hc =, a =, b =, c =, ha =, hb =, hc =, O =, P =); Height = 36; Layout = tlCenter; Left =
(75, 150, 270); Top = (70, 110, 150, 70, 110, 150, 70, 110, 150, 190, 190); Width = 35.
```

Лателе углава:

```
AutoSize = False; Caption = (a =, b =, g =); Font = Symbol, BoldItalic, 16, Aqua; Height = 36; Layout = tlCenter;
Left = 180; Top = (230, 270, 310); Width = 35.
```

Тастери за управљање:

```
Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 30; Top = (205, 255, 305); Width = 120.
```

Сачуваћемо дизајнирану форму у креирани фолдер *DveVisine*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear; Edit1.ReadOnly:=false;
  Edit2.Clear; Edit2.ReadOnly:=false;
  Edit3.Clear; Edit3.ReadOnly:=false;
  Edit4.Clear; Edit5.Clear; Edit6.Clear; Edit7.Clear; Edit8.Clear;
  Edit9.Clear; Edit10.Clear; Edit11.Clear; Edit12.Clear;
  Edit13.Clear; Edit14.Clear; Edit15.Clear; Edit16.Clear;
  Edit17.Clear; Edit18.Clear; Edit19.Clear; Edit20.Clear;
  Edit1.SetFocus; ;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,h1,h2,h3,p,o,x,y:real;
    a1,a2,a3,b1,b2,b3,c1,c2,c3:integer;
begin
  a:=StrToFloatDef(Edit1.Text,6); Edit1.Text:=FloatToStr(a);
  h2:=StrToFloatDef(Edit2.Text,6); Edit2.Text:=FloatToStr(h2);
  h3:=StrToFloatDef(Edit3.Text,3); Edit3.Text:=FloatToStr(h3);
  If (a<h3)or(a<h2) // a је хипотенуза у помоћним троугловима и мора бити већа од катета
  then begin Application.MessageBox('Не може бити троугао!', 'ГРЕШКА У УНОСУ', 0);
    Button2.Click;
  end
  else begin
    x:=Sqrt(a*a-h3*h3); // одсечак на страници c
    y:=Sqrt(a*a-h2*h2); // одсечак на страници b
    If x=0 // странице a и c су нормалне, угао β је прав
    then begin
      p:=pi/2-ArcTan(h2/y);
      b3:=90*3600;
      c3:=Round(ArcTan(h2/y)/pi*180*3600);
    end
    else If y=0 // странице a и b су нормалне, угао γ је прав
    then begin
      p:=pi/2-ArcTan(h3/x);
      b3:=Round(ArcTan(h3/x)/pi*180*3600);
      c3:=90*3600;
    end
    else begin
      p:=pi-ArcTan(h3/x)-ArcTan(h2/y);
      b3:=Round(ArcTan(h3/x)/pi*180*3600);
      c3:=Round(ArcTan(h2/y)/pi*180*3600);
    end;
  end;
  Edit4.Text:=Format('%7.3f', [a]);
  b:=h3/sin(p); Edit5.Text:=Format('%7.3f', [b]);
  c:=h2/sin(p); Edit6.Text:=Format('%7.3f', [c]);
```



```

h1:=b*h2/a;Edit7.Text:=Format('%7.3f',[h1]);
Edit8.Text:=Format('%7.3f',[h2]);
Edit9.Text:=Format('%7.3f',[h3]);
o:=a+b+c;Edit10.Text:=Format('%7.3f',[o]);
p:=a*h1/2;Edit11.Text:=Format('%7.3f',[p]);
a3:=180*3600-c3-b3;
a1:=a3 div 3600;      Edit12.Text :=IntToStr(a1);
a2:=a3 div 60 mod 60;Edit13.Text:=IntToStr(a2);
a3:=a3 mod 60;      Edit14.Text:=IntToStr(a3);
b1:=b3 div 3600;      Edit15.Text:=IntToStr(b1);
b2:=b3 div 60 mod 60;Edit16.Text:=IntToStr(b2);
b3:=b3 mod 60;      Edit17.Text:=IntToStr(b3);
c1:=c3 div 3600;      Edit18.Text:=IntToStr(c1);
c2:=c3 div 60 mod 60;Edit19.Text:=IntToStr(c2);
c3:=c3 mod 60;      Edit20.Text:=IntToStr(c3);
Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
Button2.SetFocus;

```

end;

end;

- Саставити програм који одређује странице, две висине и трећи угао троугла ако су дата два угла и висина из трећег темена троугла.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons** = []; **BorderStyle** = bsSingle; **Caption** = РЕАЛНИ ТИП;  
**ClientHeight** = 370; **ClientWidth** = 440; **Color** = Teal; **Font** = Palatino  
 Linotype, BoldItalic, 16, Aqua; **FormStyle** = fsStayOnTop; **Position** =  
 poScreenCenter.

На форму ћемо ставити лабелу за наслов, дванаест лабела за комуникацију са корисником, седам едита за унос, једанаест едита за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align** = alTop; **Alignment** = taCenter; **AutoSize** = False; **Caption** =  
 Решавање троугла; **Font** = 22, Lime; **Height** = 50; **Layout** = tlCenter.

Едит за унос висине:

**Font** = Bold, Black; **Left** = 65; **Top** = 150; **Width** = 70.

Едити за испис страница и висина:

**Font** = Bold, Black; **Left** = 255; **ReadOnly** = True; **Top** = (70, 110, 150,  
 190, 230, 270); **Width** = 70.

Едити за испис обима и површине:

**Font** = Bold, Black; **Left** = 380; **ReadOnly** = True; **Top** = (70, 110); **Width** = 80.

Едити за унос и испис углова:

**Font** = Bold, Black; **Left** = (65, 115, 155, 65, 115, 155, 215, 270, 315); **ReadOnly** = True; **Top** = (70, 70, 70, 110,  
 110, 110, 310, 310, 310); **Width** = (50, 40, 40, 50, 40, 40, 50, 40, 40).

Лабеле за комуникацију са корисником:

**AutoSize** = False; **Caption** = (hc =, a =, b =, c =, ha =, hb =, hc =, O =, P =); **Height** = 36; **Layout** = tlCenter; **Left** = (20, 210,  
 335); **Top** = (150, 70, 110, 150, 190, 230, 270, 70, 110); **Width** = 45.

Лабеле углова:

**AutoSize** = False; **Caption** = (a =, b =, g =); **Font** = Symbol, BoldItalic, 16, Aqua; **Height** = 36; **Layout** = tlCenter;  
**Left** = (20, 20, 210); **Top** = (70, 110, 310); **Width** = 45.

Тастери за управљање:

**Caption** = (Одреди, Обриши, Крај рада); **Font** = Bold; **Height** = 40; **Left** = 20; **Top** = (205, 255, 305); **Width** = 120.

Сачуваћемо дизајнирану форму у креирани фолдер *DvaUgla*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```

procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Edit4.Clear;Edit5.Clear;Edit6.Clear;Edit7.Clear;Edit8.Clear;
  Edit9.Clear; Edit10.Clear;Edit11.Clear;Edit12.Clear;Edit13.Clear;
  Edit14.Clear;Edit15.Clear;Edit16.Clear;Edit17.Clear;Edit18.Clear;
  Edit1.SetFocus;;
end;

```

Двоклик на тастер *Одреди*:

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,h1,h2,h3,p,o:real;
    a1,a2,a3,b1,b2,b3,c1,c2,c3:integer;
begin
  a1:=StrToIntDef(Edit1.Text,9);Edit1.Text:=IntToStr(a1 mod 180);
  a2:=StrToIntDef(Edit2.Text,0);Edit2.Text:=IntToStr(a2 mod 60);
  a3:=StrToIntDef(Edit3.Text,0);Edit3.Text:=IntToStr(a3 mod 60);
  a3:=a1*3600+a2*60+a3;
  b1:=StrToIntDef(Edit4.Text,9);Edit4.Text:=IntToStr(b1 mod 180);
  b2:=StrToIntDef(Edit5.Text,0);Edit5.Text:=IntToStr(b2 mod 60);
  b3:=StrToIntDef(Edit6.Text,0);Edit6.Text:=IntToStr(b3 mod 60);

```



```

b3:=b1*3600+b2*60+b3;
h3:=StrToFloatDef(Edit7.Text,0);Edit7.Text:=FloatToStr(h3);
c3:=180*3600-b3-a3;
If (c3<0)or(a3<=0)or(c3<=0)// збир два унутрашња угла троугла је мањи од 180 степени
then begin Application.MessageBox('Не може бити троугао!', 'ГРЕШКА У УНОСУ',0);
      Button2.Click;
    end
else begin a:=h3/Sin(b3/180/3600*pi);Edit8.Text:=Format('%7.3f',[a]);
      b:=h3/Sin(a3/180/3600*pi);Edit9.Text:=Format('%7.3f',[b]);
      c:=Sqrt(b*b-h3*h3)+Sqrt(a*a-h3*h3);Edit10.Text:=Format('%7.3f',[c]);
      h1:=c*Sin(b3/180/3600*pi);Edit11.Text:=Format('%7.3f',[h1]);
      h2:=c*Sin(a3/180/3600*pi);Edit12.Text:=Format('%7.3f',[h2]);
      Edit13.Text:=Format('%7.3f',[h3]);
      o:=a+b+c; Edit14.Text:=Format('%7.3f',[o]);
      p:=a*h1/2;Edit15.Text:=Format('%7.3f',[p]);
      c1:=c3 div 3600; Edit16.Text:=IntToStr(c1);
      c2:=c3 div 60 mod 60;Edit17.Text:=IntToStr(c2);
      c3:=c3 mod 60; Edit18.Text:=IntToStr(c3);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
      Button2.SetFocus;
    end;
end;

```

• **Саставити програм који одређује странице и углове троугла ако су дате све три висине троугла**

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = РЕАЛНИ ТИП;  
ClientHeight = 370; ClientWidth = 440; Color = Teal; Font = Palatino  
Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position =  
poScreenCenter.**

На форму ћемо ставити лабелу за наслов, четрнаест лабела за комуникацију са корисником, три едита за унос, седамнаест едита за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption =  
Решавање троугла; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едити за унос:

**Font = Bold, Black; Left = 75; Top = (70, 110, 150); Width = 70.**

Едити за испис страница и висина:

**Font = Bold, Black; Left = (215, 345); ReadOnly = True; Top = (70,  
110, 150); Width = 70.**

Едити за испис обима и површину:

**Font = Bold, Black; Left = (215, 335); ReadOnly = True; Top = 190; Width = 80.**

Едити за испис углова:

**Font = Bold, Black; Left = (215, 270, 315); ReadOnly = True; Top = (230, 270, 310); Width = (50, 40, 40).**

Лабеле за комуникацију са корисником:

**AutoSize = False; Caption = (ha =, hb =, hc =, a =, b =, c =, ha =, hb =, hc =, O =, P =); Height = 36; Layout = tlCenter; Left  
= (75, 150, 270); Top = (70, 110, 150, 70, 110, 150, 70, 110, 150, 190, 190); Width = 35.**

Лабеле углова:

**AutoSize = False; Caption = (a =, b =, g =); Font = Symbol, BoldItalic, 16, Aqua; Height = 36; Layout = tlCenter;  
Left = 180; Top = (230, 270, 310); Width = 35.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 30; Top = (205, 255, 305); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *TriVisine*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```

procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit3.ReadOnly:=false;
      Edit4.Clear;Edit5.Clear;Edit6.Clear;Edit7.Clear;Edit8.Clear;
      Edit9.Clear; Edit10.Clear;Edit11.Clear;Edit12.Clear;
      Edit13.Clear;Edit14.Clear;Edit15.Clear;Edit16.Clear;
      Edit17.Clear;Edit18.Clear;Edit19.Clear;Edit20.Clear;
      Edit1.SetFocus;
end;

```

Двоклик на тастер *Одреди*:

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,h1,h2,h3,p,o:real;
    a1,a2,a3,b1,b2,b3,c1,c2,c3:integer;
begin h1:=StrToFloatDef(Edit1.Text,5);Edit1.Text:=FloatToStr(h1);Edit7.Text:=Format('%7.3f',[h1]);
      h2:=StrToFloatDef(Edit2.Text,5);Edit2.Text:=FloatToStr(h2);Edit8.Text:=Format('%7.3f',[h2]);
      h3:=StrToFloatDef(Edit3.Text,5);Edit3.Text:=FloatToStr(h3);Edit9.Text:=Format('%7.3f',[h3]);
      a:=1/h1;b:=1/h2;c:=1/h3; // странице троугла су обрнуто пропорционалне висинама

```



```

If (a+b>c)and(b+c>a)and(c+a>b) // збир сваке две мора бити већи од треће странице
then begin p:=Sqrt((a+b+c)*(-a+b+c)*(a-b+c)*(a+b-c))/2;
a:=a/p;Edit4.Text:=Format('%7.3f',[a]);
b:=b/p;Edit5.Text:=Format('%7.3f',[b]);
c:=c/p;Edit6.Text:=Format('%7.3f',[c]);
o:=a+b+c; Edit10.Text:=Format('%7.3f',[o]);
p:=a*h1/2;Edit11.Text:=Format('%7.3f',[p]);
p:=(b*b+c*c-a*a)/(2*b*c);
If p=0 then a3:=90*3600 // b*b+c*c=a*a, угао α је прав
else If p<0 // троугао је тупоугли са теменом тупог угла у В
then a3:=Round((pi+ArcTan(Sqrt(1-p*p)/p))/pi*180*3600)
else a3:=Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600);
p:=(c*c+a*a-b*b)/(2*a*c);
If p=0 then b3:=90*3600 // c*c+a*a=b*b, угао β је прав
else If p<0 // троугао је тупоугли са теменом тупог угла у В
then b3:=Round((pi+ArcTan(Sqrt(1-p*p)/p))/pi*180*3600)
else b3:=Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600);
c3:=180*3600-b3-a3;
a1:=a3 div 3600; Edit12.Text:=IntToStr(a1);
a2:=a3 div 60 mod 60;Edit13.Text:=IntToStr(a2);
a3:=a3 mod 60; Edit14.Text:=IntToStr(a3);
b1:=b3 div 3600; Edit15.Text:=IntToStr(b1);
b2:=b3 div 60 mod 60;Edit16.Text:=IntToStr(b2);
b3:=b3 mod 60; Edit17.Text:=IntToStr(b3);
c1:=c3 div 3600; Edit18.Text:=IntToStr(c1);
c2:=c3 div 60 mod 60;Edit19.Text:=IntToStr(c2);
c3:=c3 mod 60; Edit20.Text:=IntToStr(c3);
Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
Button2.SetFocus;
end
else begin Application.MessageBox('Не постоји такав троугао!', 'ГРЕШКА У УНОСУ',0);
Button2.Click;
end;
end;
end;
Или овако:
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,h1,h2,h3,p,o,x,y:real;
a1,a2,a3,b1,b2,b3,c1,c2,c3:integer;
begin h1:=StrToFloatDef(Edit1.Text,5);Edit1.Text:=FloatToStr(h1);Edit7.Text:=Format('%7.3f',[h1]);
h2:=StrToFloatDef(Edit2.Text,5);Edit2.Text:=FloatToStr(h2);Edit8.Text:=Format('%7.3f',[h2]);
h3:=StrToFloatDef(Edit3.Text,5);Edit3.Text:=FloatToStr(h3);Edit9.Text:=Format('%7.3f',[h3]);
If (1/h1+1/h2>1/h3)and(1/h2+1/h3>1/h1)and(1/h3+1/h1>1/h2)
{ ако изразимо странице троугла помоћу површине и одговарајућих висина и сетимо се да је збир
сваке две странице већи од треће, заменом страница и проширењем неједнакости са 2Р добијамо ове
неједнакости које морају да важе за висине сваког троугла.
како је  $ah_1=bh_2=ch_3/: h_1h_2 \Rightarrow \frac{a}{h_2}=\frac{b}{h_1}=\frac{c}{h_3h_2}$ , па је  $\cos\alpha=(b^2+c^2-a^2)/2bc=(h_1^2+(\frac{h_1h_2}{h_3})^2-h_2^2)/2h_1\frac{h_1h_2}{h_3}$  }
then begin y:=h1*h2/h3; x:=(h1*h1+y*y-h2*h2)/(2*h1*y); // x је косинус угла α
b:=h3/Sqrt(1-x*x);Edit5.Text:=Format('%7.3f',[b]);
c:=h2/Sqrt(1-x*x);Edit6.Text:=Format('%7.3f',[c]);
a:=b*h2/h1;Edit4.Text:=Format('%7.3f',[a]);
p:=a*h1/2; Edit10.Text:=Format('%7.3f',[p]);
o:=a+b+c; Edit11.Text:=Format('%7.3f',[o]);
If x=0
then a3:=90*3600
else If x<0
then a3:=Round((pi+ArcTan(Sqrt(1-x*x)/x))*180/pi*3600)
else a3:=Round(ArcTan(Sqrt(1-x*x)/x)*180/pi*3600);
p:=(h2*h2+y*y-h1*h1)/(2*h2*y); // косинус угла β
If p=0
then b3:=90*3600
else If p<0
then b3:=Round((pi+ArcTan(Sqrt(1-p*p)/p))*180/pi*3600)
else b3:=Round(ArcTan(Sqrt(1-p*p)/p)*180/pi*3600);
c3:=180*3600-a3-b3;
a1:=a3 div 3600; Edit12.Text:=IntToStr(a1);
a2:=a3 div 60 mod 60;Edit13.Text:=IntToStr(a2);
a3:=a3 mod 60; Edit14.Text:=IntToStr(a3);
b1:=b3 div 3600; Edit15.Text:=IntToStr(b1);
b2:=b3 div 60 mod 60;Edit16.Text:=IntToStr(b2);
b3:=b3 mod 60; Edit17.Text:=IntToStr(b3);
c1:=c3 div 3600; Edit18.Text:=IntToStr(c1);
c2:=c3 div 60 mod 60;Edit19.Text:=IntToStr(c2);
c3:=c3 mod 60; Edit20.Text:=IntToStr(c3);
Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
Button2.SetFocus;
end
else begin Application.MessageBox('Не постоји такав троугао!', 'ГРЕШКА У УНОСУ',0);
Button2.Click;
end;
end;
end;

```



• **Саставити програм који за унети редни број дана у недељи одређује његов назив.**

Користићемо делфи функцију *LowerCase* да бисмо поједноставили тестирање уписаног стринга.

Подесићемо карактеристике форме на следећи начин:

```
BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ
ГРАНАЊА; ClientHeight = 250; ClientWidth = 350; Color = Teal; Font
= Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop;
Position = poScreenCenter.
```

На форму ћемо ставити лателу за наслов, лателу за комуникацију са корисником, едит за унос, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

```
Align = alTop; Alignment = taCenter; AutoSize = False; Caption =
Дани у недељи; Font = 22, Lime; Height = 50; Layout = tlCenter.
```

Едит за унос:

```
Font = Bold, Black; Left = 140; MaxLength = 1; Top = 90; Width = 50.
```

Латела за испис резултата:

```
Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 50; Layout = tlCenter.
```

Латела за комуникацију са корисником:

```
AutoSize = False; Caption = Редни број; Height = 36; Layout = tlCenter; Left = 20; Top = 90; Width = 120.
```

Тастери за управљање:

```
Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (210, 70, 210); Top = (90, 150, 150);
Width = 120.
```

Сачуваћемо дизајнирану форму у креирани фолдер *ImeDana*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Label3.Caption:='';
      Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди* (варијанта са **If**-наредбом):

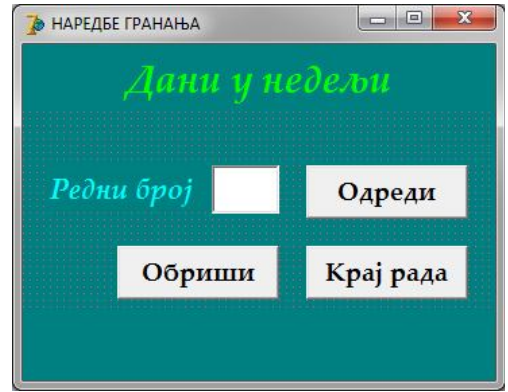
```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    b:integer;
begin b:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(b);
      If b=1 then s:='Први дан у недељи је понедељак.'
      else If b=2 then s:='Други дан у недељи је уторак.'
      else If b=3 then s:='Трећи дан у недељи је среда.'
      else If b=4 then s:='Четврти дан у недељи је четвртак.'
      else If b=5 then s:='Пети дан у недељи је петак.'
      else If b=6 then s:='Шести дан у недељи је субота.'
      else If b=7 then s:='Седми дан у недељи је недеља.'
      else s:=IntToStr(b)+' . није редни број дана у недељи.';
      Label3.Caption:=s;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Двоклик на тастер *Одреди* (варијанта са **Case**-наредбом):

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    b:integer;
begin b:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(b);
      Case b of
        1 : s:='Први дан у недељи је понедељак.'
        2 : s:='Други дан у недељи је уторак.'
        3 : s:='Трећи дан у недељи је среда.'
        4 : s:='Четврти дан у недељи је четвртак.'
        5 : s:='Пети дан у недељи је петак.'
        6 : s:='Шести дан у недељи је субота.'
        7 : s:='Седми дан у недељи је недеља.'
      else s:=Edit1.Text+' . није редни број дана у недељи.';
      Label3.Caption:=s;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Двоклик на тастер *Одреди* (варијанта са скупом и низом константи):

```
procedure TForm1.Button1Click(Sender: TObject);
const d:array[1..7]of string=('понедељак', 'уторак', 'среда', 'четвртак', 'петак', 'субота', 'недеља');
var s:string;
    b:integer;
begin b:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(b);
      If b in [1..7]
      then s:=Edit1.Text+' . дан у недељи је '+d[b]+'.'
end;
```





```

    else s:=Edit1.Text+'. није редни број дана у недељи.';
    Label3.Caption:=s;
    Edit1.ReadOnly:=true;
    Button2.SetFocus;
end;

```

Решење је комплетније ако се уведе контрола уноса. Кориснику нећемо дозволити да унесе број који није редни број неког дан у недељи. Контролу уноса ћемо увести као процедуру која се извршава када се напусти едит за унос података. Селектујемо едит за унос, у прозору *ObjectInspector* на листићу *Events* двоклик на поље вредности догађаја *OnExit* декларише одговарајућу процедуру коју дефинишемо:

```

procedure TForm1.Edit1Exit(Sender: TObject);
begin If not (StrToIntDef(Edit1.Text,0) in [1..7])
    then begin MessageBox(0,'Морате унети број од 1 до 7.','ГРЕШКА',MB_ICONERROR);
        Edit1.Clear;Edit1.SetFocus;
    end;
end;

```

• **Саставити програм који за унет назив дана у недељи одређује његов редни број.**

Подесићемо карактеристике форме на следећи начин:

```

BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ
ГРАНАЊА; ClientHeight = 250; ClientWidth = 410; Color = Teal; Font
= Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop;
Position = poScreenCenter.

```

На форму ћемо ставити лателу за наслов, лателу за комуникацију са корисником, едит за унос, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

```

Align = alTop; Alignment = taCenter; AutoSize = False; Caption =
Дани у недељи; Font = 22, Lime; Height = 50; Layout = tlCenter.

```

Едит за унос:

```

Font = Bold, Black; Left = 140; CharCase = ecLowerCase; Top = 90;
Width = 120.

```

Латела за испис резултата:

```

Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 50; Layout = tlCenter.

```

Латела за комуникацију са корисником:

```

AutoSize = False; Caption = Назив дана; Height = 36; Layout = tlCenter; Left = 20; Top = 90; Width = 120.

```

Тастери за управљање:

```

Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (210, 140, 210); Top = (90, 150, 150);
Width = 120.

```

Сачуваћемо дизајнирану форму у креирани фолдер *RBDana*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```

procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
    Label3.Caption:='';
    Edit1.SetFocus;
end;

```

Двоклик на тастер *Одреди* (варијанта са *If*-наредбом):

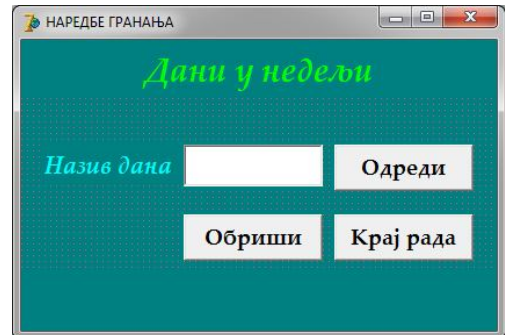
```

procedure TForm1.Button1Click(Sender: TObject);
var a,s:string;
begin s:=Edit1.Text;a:=LowerCase(s);
    If a='понедељак' then s:=s+' је први дан у недељи.'
    else If a='уторак' then s:=s+' је други дан у недељи.'
    else If a='среда' then s:=s+' је трећи дан у недељи.'
    else If a='четвртак' then s:=s+' је четврти дан у недељи.'
    else If a='петак' then s:=s+' је пети дан у недељи.'
    else If a='субота' then s:=s+' је шести дан у недељи.'
    else If a='недеља' then s:=s+' је седми дан у недељи.'
    else s:=s+' није дан у недељи.';
    Label3.Caption:=s;
    Edit1.ReadOnly:=true;
    Button2.SetFocus;
end;

```

**напомена:**

Функција *LowerCase* не ради са ћириличним словима, па овај програм неће радити ако дане у недељи исписујемо малим и великим словима иако је алгоритамски коректан, зато смо поставили *CharCase = ecLowerCase* што би требало да помогне да се називи дана могу да уписују само малим словима (зависи од инсталираног оперативног система). Ако ово не ради, а желимо да направимо програм који ће коректно тестирати низ унетих ћириличних слова да би се утврдио дан у недељи, морамо направити сами одговарајућу функцију која ће сва велика ћирилична слова претворити у мала. Са знањем које у овом тренутку поседујемо, то нисмо у могућности. Делимично решење проблема је да се уместо едита, за унос података користи комбиновани оквир (*ComboBox*) из кога би се бирао назив дана. У том случају



задатак би се могао решити и помоћу *Case* наредбе. Приказаћемо и такво решење. У дизајнирању форме измене су незнатне.

Додајемо комбиновани оквир за унос података:

**AutoCloseUp** = true; **AutoDrop Down** = true; **Items** = (понедељак, уторак, среда, четвртак, петак, субота, недеља); **Left** = 130; **Style** = csOwner DrawVariable; **Top** = 90; **Width** = 125.

Испред оквира је иста лабела за комуникацију. Делимично се мењају и процедуре за тастере *Обриши* и *Одреди*.

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  ComboBox1.ItemIndex:=-1;
  Label3.Caption:='';
  ComboBox1.Enabled:=true;
  ComboBox1.SetFocus;
end;

procedure TForm1.Button1Click(Sender: TObject);
var s:string;
begin
  s:=ComboBox1.Text;
  If s='понедељак'
  then s:=s+' је први дан у недељи.'
  else If s='уторак' then s:=s+' је други дан у недељи.'
  else If s='среда' then s:=s+' је трећи дан у недељи.'
  else If s='четвртак' then s:=s+' је четврти дан у недељи.'
  else If s='петак' then s:=s+' је пети дан у недељи.'
  else If s='субота' then s:=s+' је шести дан у недељи.'
  else s:=s+' је седми дан у недељи.';
  Label3.Caption:=s;
  ComboBox1.Enabled:=false;
  Button2.SetFocus;
end;
```

Ова процедура записана са *Case* наредбом је једноставнија:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
begin
  s:=ComboBox1.Text;
  Case ComboBox1.ItemIndex of
    0 : s:=s+' је први дан у недељи.';
    1 : s:=s+' је други дан у недељи.';
    2 : s:=s+' је трећи дан у недељи.';
    3 : s:=s+' је четврти дан у недељи.';
    4 : s:=s+' је пети дан у недељи.';
    5 : s:=s+' је шести дан у недељи.';
    6 : s:=s+' је седми дан у недељи.';
  end;
  Label3.Caption:=s;
  ComboBox1.Enabled:=false;
  Button2.SetFocus;
end;
```

**AutoCloseUp** = true аутоматски затвара контејнер када се изабере нека од понуђених ставки;

**AutoDropDown** = true аутоматски отвара контејнер када објекат постане активан;

**Items** = (понедељак, уторак, среда, четвртак, петак, субота, недеља) је садржај контејнера, ако желимо да садржај буде ћириличан, ставке се уписују у *Notepad*, а затим копирају у *StringListEditor* који се отвара када се кликне на карактеристику *Items*;

**Style** = csOwnerDrawVariable забрањује да се у оквир упише било шта осим ставки из контејнера (зато нема поруке - није дан у недељи).

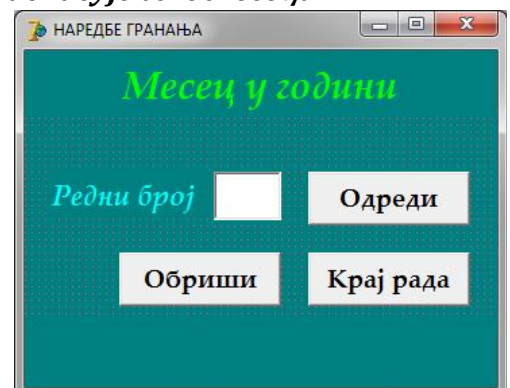
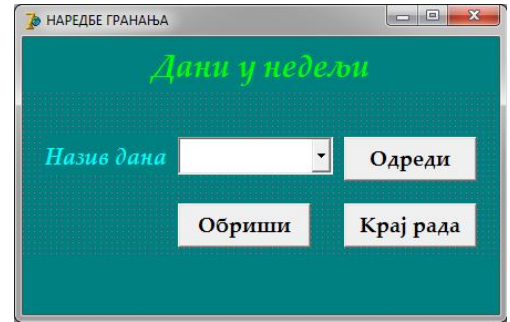
У наредним задацима алгоритме ћемо описивати и са *If* и са *Case* када је то могуће, а у компликованијим задацима комбиноваћемо ове наредбе тако да програм буде што једноставнији за читање.

Следећа два задатка су готово истоветна са претходна два, зато ћемо, без велике буке само написати све варијанте процедуре за тастер *Одреди*.

### • Саставити програм који за унети редни број месеца испишује име месеца.

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Label3.Caption:='';
  Edit1.SetFocus;
end;

Ca If
procedure TForm1.Button1Click(Sender: TObject);
var s:string;b:integer;
begin
  b:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(b);
  s:=Edit1.Text+'. месец у години је ';
  If b=1 then s:=s+'јануар'
  else If b=2 then s:=s+'фебруар'
  else If b=3 then s:=s+'март'
  else If b=4 then s:=s+'април'
  else If b=5 then s:=s+'мај'
```



```

else If b=6 then s:=s+'јун'
else If b=7 then s:=s+'јул'
else If b=8 then s:=s+'август'
else If b=9 then s:=s+'септембар'
else If b=10 then s:=s+'октобар'
else If b=11 then s:=s+'новембар'
else If b=12 then s:=s+'децембар'
else s:=Edit1.Text+'. није редни број'+#13+'месеца у години';
Label3.Caption:=s;
Edit1.ReadOnly:=true;
Button2.SetFocus;
end;

```

### Ca Case

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string;b:integer;
begin b:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(b);
s:=Edit1.Text+'. месец у години је ';
Case b of
1 : s:=s+'јануар';
2 : s:=s+'фебруар';
3 : s:=s+'март';
4 : s:=s+'април';
5 : s:=s+'мај';
6 : s:=s+'јун';
7 : s:=s+'јул';
8 : s:=s+'август';
9 : s:=s+'септембар';
10: s:=s+'октобар';
11: s:=s+'новембар';
12: s:=s+'децембар';
else s:=Edit1.Text+'. није редни број'+#13+'месеца у години';
end;
Label3.Caption:=s;
Edit1.ReadOnly:=true;
Button2.SetFocus;
end;

```

### Са скуповним типом и контролом уноса

```

procedure TForm1.Edit1Exit(Sender: TObject);
begin If not (StrToIntDef(Edit1.Text,0) in [1..12])
then begin MessageBox(0,'Морате унети број од 1 до 12.','ГРЕШКА',MB_ICONERROR);
Edit1.Clear;
Edit1.SetFocus;
end;
end;

procedure TForm1.Button1Click(Sender: TObject);
const d:array[1..12] of string=('јануар','фебруар','март','април','мај','јун','јул',
'август','септембар','октобар','новембар','децембар');
var s:string;
b:integer;
begin b:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(b);
s:=Edit1.Text+'. месец у години је '+d[b]+'.'
Label3.Caption:=s;
Edit1.ReadOnly:=true;
Button2.SetFocus;
end;
end;

```

### • Саставити програм који за унети редни број месеца и годину исписује број дана у месецу.

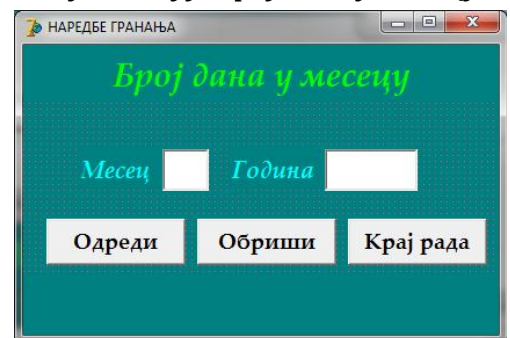
Овде само треба водити рачуна о преступним годинама. За сада важи, година је преступна ако је дељива са 4 и није последња година века, односно, није дељива са 100 или ако је дељива са 400.

```

procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
Edit2.Clear;Edit2.ReadOnly:=false;
Label3.Caption:='';
Edit1.SetFocus;
end;

procedure TForm1.Button1Click(Sender: TObject);
var s:string;
m,g,p,b:integer;
begin m:=StrToIntDef(Edit1.Text,10);
Edit1.Text:=IntToStr(m);
g:=StrToIntDef(Edit2.Text,2000);
Edit2.Text:=IntToStr(g);
If (m<1) or (m>12)
then s:='Погрешан унос месеца у години.'
else begin If (g mod 4=0)and(g mod 100<>0)or(g mod 400=0)
then p:=1
else p:=0;

```



```

Case m of
  2 : b:=28+p;
  4,6,9,11 : b:=30;
  else b:=31;
end;
s:=IntToStr(b);
Case m of
  1 : s:='Јануар има '+s+' дан.';
  2 : s:='Фебруар има '+s+' дана.';
  3 : s:='Март има '+s+' дан.';
  4 : s:='Април има '+s+' дана.';
  5 : s:='Мај има '+s+' дан.';
  6 : s:='Јун има '+s+' дана.';
  7 : s:='Јул има '+s+' дан.';
  8 : s:='Август има '+s+' дан.';
  9 : s:='Септембар има '+s+' дана.';
  10: s:='Октобар има '+s+' дан.';
  11: s:='Новембар има '+s+' дана.';
  12: s:='Децембар има '+s+' дан.';
end;
end;
Label3.Caption:=s;
Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
Button2.SetFocus;
end;

```

Обе **Case** наредбе у претходној процедури могу се заменити групом **If** наредби.

#### Са скуповним типом и контролом уноса

```

procedure TForm1.Edit1Exit(Sender: TObject);
begin
  if not (StrToIntDef(Edit1.Text,0) in [1..12])
    then begin
      MessageBox(0, 'Морате унети број од 1 до 12.', 'ГРЕШКА', MB_ICONERROR);
      Edit1.Clear;Edit1.SetFocus;
    end;
end;

procedure TForm1.Button1Click(Sender: TObject);
const d:array[1..12] of string = ('јануар', 'фебруар', 'март', 'април', 'мај', 'јун', 'јул',
                                'август', 'септембар', 'октобар', 'новембар', 'децембар');
var s:string;
    m,g,p,b:integer;
begin
  m:=StrToIntDef(Edit1.Text,10); Edit1.Text:=IntToStr(m);
  g:=StrToIntDef(Edit2.Text,1960);Edit2.Text:=IntToStr(g);
  if (g mod 4=0)and(g mod 100<>0)or(g mod 400=0)
    then p:=1
    else p:=0;
  Case m of
    2 : b:=28+p;
    4,6,9,11 : b:=30;
    else b:=31;
  end;
  s:=d[m]+'има '+IntToStr(b)+' дана.';
  Label3.Caption:=s;
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
  Button2.SetFocus;
end;

```

#### • Саставити програм који за унети датум одређује сутрашњи датум.

Променљива  $p$  биће преступни дан (0 ако година није преступна, 1 ако јесте). Ради лакше контроле датума, увешћемо још једну променљиву  $bd$  која ће представљати број дана у месецу (28, 29, 30 или 31). Након уноса датума прво ћемо извршити контролу унетог, а затим, ако је датум коректан, одредити сутрашњи датум.

Подоцићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 250; ClientWidth = 400; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лателу за наслов, лателу за комуникацију са корисником, три едита за унос, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Сутрашњи датум; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едита за унос:

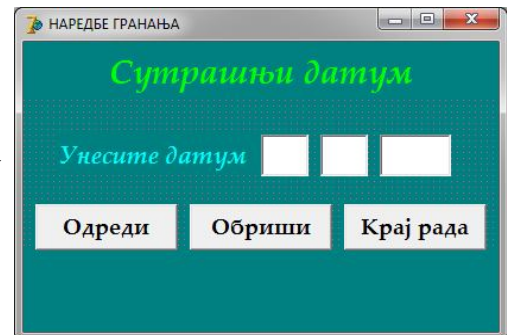
**Font = Bold, Black; Left = (200, 250, 300); MaxLength = (2, 2, 4); Top = 80; Width = (40, 40, 60).**

Латела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.**

Латела за комуникацију са корисником:

**AutoSize = False; Caption = Унесите датум; Height = 36; Layout = tlCenter; Left = 30; Top = 80; Width = 170.**





Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 10; Top = (10, 140, 270); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Sutra*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit3.ReadOnly:=false;
      Label3.Caption:='';
      Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    d,m,g,p,bd:integer;
begin d:=StrToIntDef(Edit1.Text,31); Edit1.Text:=IntToStr(d);
      m:=StrToIntDef(Edit2.Text,12); Edit2.Text:=IntToStr(m);
      g:=StrToIntDef(Edit3.Text,2013);Edit3.Text:=IntToStr(g);
      If (g mod 4=0)and(g mod 100<>0)or(g mod 400=0)
        then p:=1
         else p:=0;
      Case m of
        2 : bd:=28+p;
        4,6,9,11 : bd:=30;
        else bd:=31;
      end;
      If (d>bd)or(d<1)or(m>12)or(m<1)
        then begin MessageBox(0,'Погрешан унос датума.','ГРЕШКА',0);
                Button2.Click;
            end
        else begin If d<bd
                  then d:=d+1
                  else begin d:=1;
                          If m<12
                            then m:=m+1
                             else begin m:=1; g:=g+1;
                                    end;
                                end;
                        s:=Format('Сутрашњи датум је: %1d.%1d.%4d.',[d,m,g]);
                        Label3.Caption:=s;
                        Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
                        Button2.SetFocus;
                    end;
            end;
end;
```

Case наредба за одређивање броја дана у месецу може се једноставно заменити двема *If* наредбама.

### • Саставити програм који за унети датум одређује јучерашњи датум.

Променљива *p* биће преступни дан (0 ако година није преступна, 1 ако јесте). Ради лакше контроле датума, увешћемо још једну променљиву *bd* која ће представљати број дана у месецу (28, 29, 30 или 31). Након уноса датума прво ћемо извршити контролу унетог, а затим, ако је датум коректан, одредити јучерашњи датум.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 250; ClientWidth = 400; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, лабелу за комуникацију са корисником, три едита за унос, лабелу за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Јучерашњи датум; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едити за унос:

**Font = Bold, Black; Left = (200, 250, 300); MaxLength = (2, 2, 4); Top = 80; Width = (40, 40, 60).**

Лабела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.**

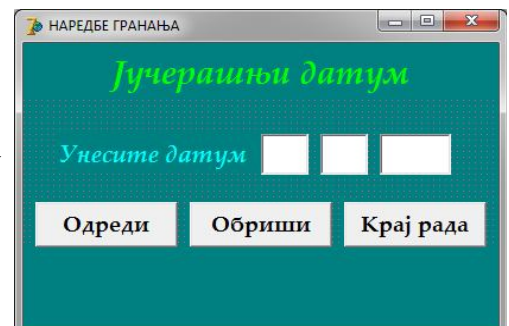
Лабела за комуникацију са корисником:

**AutoSize = False; Caption = Унесите датум; Height = 36; Layout = tlCenter; Left = 30; Top = 80; Width = 170.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 10; Top = (10, 140, 270); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Juce*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.



Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Label3.Caption:='';
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    d,m,g,p,b:integer;
begin
  d:=StrToIntDef(Edit1.Text,1);   Edit1.Text:=IntToStr(d);
  m:=StrToIntDef(Edit2.Text,1);   Edit2.Text:=IntToStr(m);
  g:=StrToIntDef(Edit3.Text,2014);Edit3.Text:=IntToStr(g);
  If (g mod 4=0)and(g mod 100<>0)or(g mod 400=0)
    then p:=1
    else p:=0;
  If m=2
    then bd:=28+p
    else If m in [4,6,9,11]
      then bd:=30
      else bd:=31;
  If (d>bd)or(d<1)or(m>12)or(m<1)
    then s:='Погрешан унос датума.'
    else begin If d>1
      then d:=d-1
      else begin m:=m-1;
        If m=0
          then begin m:=12; g:=g-1;
            end;
          Case m of
            2 : d:=28+p;
            4,6,9,11 : d:=30;
            else d:=31;
          end;
        end;
      s:=Format('Јучерашњи датум је: %1d.%1d.%4d.',[d,m,g]);
    end;
  Label3.Caption:=s;
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
  Button2.SetFocus;
end;
```

Две *If* наредбе за одређивање броја дана у месецу може се једноставно заменити *Case* наредбом.

### • Саставити програм који за унети датум одређује редни број дана у години.

Подесићемо карактеристике форме на следећи начин:

*BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 250; ClientWidth = 400; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.*

На форму ћемо ставити лателу за наслов, лателу за комуникацију са корисником, три едита за унос, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Редни број дана у години; Font = 22, Lime; Height = 50; Layout = tlCenter.*

Едита за унос:

*Font = Bold, Black; Left = (200, 250, 300); MaxLength = (2, 2, 4); Top = 80; Width = (40, 40, 60).*

Латела за испис резултата:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.*

Латела за комуникацију са корисником:

*AutoSize = False; Caption = Унесите датум; Height = 36; Layout = tlCenter; Left = 30; Top = 80; Width = 170.*

Тастери за управљање:

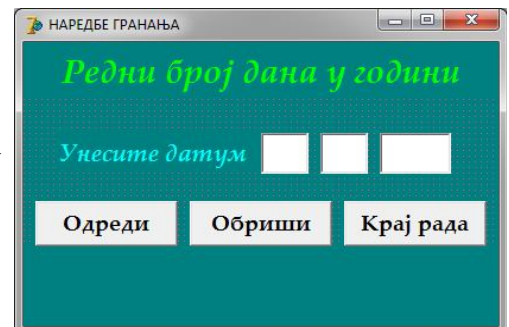
*Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 10; Top = (10, 140, 270); Width = 120.*

Сачуваћемо дизајнирану форму у креирани фолдер *RBDanaUG*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Label3.Caption:='';
end;
```



```

Edit1.SetFocus;
end;
Двоклик на тастер Одреди:
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    d,m,g,p,b:integer;
begin d:=StrToIntDef(Edit1.Text,20); Edit1.Text:=IntToStr(d);
      m:=StrToIntDef(Edit2.Text,10); Edit2.Text:=IntToStr(m);
      g:=StrToIntDef(Edit3.Text,2010);Edit3.Text:=IntToStr(g);
      p:=0;
      If (g mod 4=0)and(g mod 100<>0)or(g mod 400=0) then p:=1
      If m=2
        then b:=28+p
        else If m in [4,6,9,11]
          then b:=30
          else b:=31;
      If (d>b)or(d<1)or(m>12)or(m<1)
        then s:='Погрешан унос датума.'
        else begin If m>1 then d:=d+31;
                  If m>2 then d:=d+28+p;
                  If m>3 then d:=d+31;
                  If m>4 then d:=d+30;
                  If m>5 then d:=d+31;
                  If m>6 then d:=d+30;
                  If m>7 then d:=d+31;
                  If m>8 then d:=d+31;
                  If m>9 then d:=d+30;
                  If m>10 then d:=d+31;
                  If m>11 then d:=d+30;
                  s:='Уписани датум је '+IntToStr(d)+' дан у години.';
                end;
      Label3.Caption:=s;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
      Button2.SetFocus;
end;

```

Процедура за тастер *Одреди* може се написати и овако:

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    d,m,g,p,bd:integer;
begin d:=StrToIntDef(Edit1.Text,20); Edit1.Text:=IntToStr(d);
      m:=StrToIntDef(Edit2.Text,10); Edit2.Text:=IntToStr(m);
      g:=StrToIntDef(Edit3.Text,2010);Edit3.Text:=IntToStr(g);
      If (g mod 4=0)and(g mod 100<>0)or(g mod 400=0)
        then p:=1
        else p:=0;
      Case m of
        2 : bd:=28+p;
        4,6,9,11 : bd:=30;
        else bd:=31;
      end;
      If (d>bd)or(d<1)or(m>12)or(m<1)
        then s:='Погрешан унос датума.'
        else begin Case m of
                  2 : d:=d+31; // прошао је јануар, додајемо 31 дан
                  3 : d:=d+31+28+p; // прошли су јануар и фебруар
                  4 : d:=d+31+28+p+31; // прошли су јануар, фебруар и март ...
                  5 : d:=d+31+28+p+31+30;
                  6 : d:=d+31+28+p+31+30+31;
                  7 : d:=d+31+28+p+31+30+31+30;
                  8 : d:=d+31+28+p+31+30+31+30+31;
                  9 : d:=d+31+28+p+31+30+31+30+31+31;
                  10 : d:=d+31+28+p+31+30+31+30+31+31+30;
                  11 : d:=d+31+28+p+31+30+31+30+31+31+30+31;
                  12 : d:=d+31+28+p+31+30+31+30+31+31+30+31+30;
                end;
                s:='Уписани датум је '+IntToStr(d)+' дан у години.';
              end;
      Label3.Caption:=s;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
      Button2.SetFocus;
end;

```

- **Саставити програм који за унети редни број дана и годину одређује датум.**

Подесићемо карактеристике форме на следећи начин:

***BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 250; ClientWidth = 400; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.***

На форму ћемо ставити лателу за наслов, лателу за комуникацију са корисником, два едита за унос, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Датум према редном броју; Font = 22, Lime; Height = 50; Layout = tlCenter.*

Едити за унос:

*Font = Bold, Black; Left = (270, 320); MaxLength = (3, 4); Top = 80; Width = (40, 60).*

Лабела за испис резултата:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.*

Лабела за комуникацију са корисником:

*AutoSize = False; Caption = Редни број дана и година; Height = 36; Layout = tlCenter; Left = 20; Top = 80; Width = 250.*

Тастери за управљање:

*Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 10; Top = (10, 140, 270); Width = 120.*

Сачуваћемо дизајнирану форму у креирани фолдер *DatumRB*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

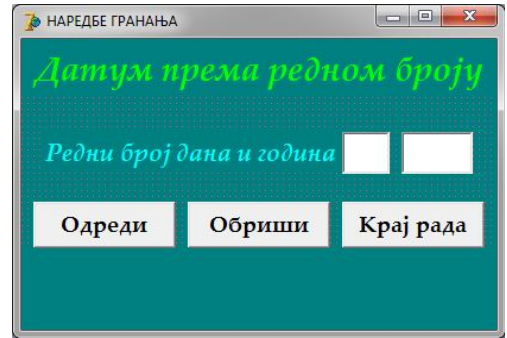
```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Label3.Caption:='';
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    d,m,g,p:integer;
begin
  d:=StrToIntDef(Edit1.Text,111); Edit1.Text:=IntToStr(d);
  g:=StrToIntDef(Edit2.Text,2011); Edit2.Text:=IntToStr(g);
  If (g mod 4=0)and(g mod 100<>0)or(g mod 400=0)
  then p:=1
  else p:=0;
  If (d>365+p)or(d<1)
  then s:='Погрешан унос редног броја дана.'
  else begin If (p=1)and(d=60)
            then begin m:=2;d:=29;
                  end
            else begin If d>59 then d:=d-p;
                      Case d of
                        1..31 : m:= 1;
                        32..59 : begin d:=d-31;m:=2;end;
                        60..90 : begin d:=d-59;m:=3;end;
                        91..120: begin d:=d-90;m:=4;end;
                        121..151: begin d:=d-120;m:=5;end;
                        152..181: begin d:=d-151;m:=6;end;
                        182..212: begin d:=d-181;m:=7;end;
                        213..243: begin d:=d-212;m:=8;end;
                        244..273: begin d:=d-243;m:=9;end;
                        274..304: begin d:=d-273;m:=10;end;
                        305..334: begin d:=d-304;m:=11;end;
                        else begin d:=d-334;m:=12;end;
                      end;
            end;
            s:=Format('Тражени датум је: %1d.%1d.%4d.', [d,m,g]);
  end;
  Label3.Caption:=s;
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
  Button2.SetFocus;
end;
```

Процедура за тастер *Одреди* може се написати, можда јасније и једноставније, овако:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    d,m,g,p:integer;
begin
  d:=StrToIntDef(Edit1.Text,111); Edit1.Text:=IntToStr(d);
  g:=StrToIntDef(Edit2.Text,2011); Edit2.Text:=IntToStr(g);
  If (g mod 4=0)and(g mod 100<>0)or(g mod 400=0)
  then p:=1
  else p:=0;
  If (d>365+p)or(d<1)
  then s:='Погрешан унос редног броја дана.'
  else begin m:= 1;
        If d>31 then begin m:=2;d:=d-31; end;
        If d>28+p then begin m:=3;d:=d-28-p; end;
        If d>31 then begin m:=4;d:=d-31; end;
        If d>30 then begin m:=5;d:=d-30; end;
        If d>31 then begin m:=6;d:=d-31; end;
        end;
```





```

    If d>30 then begin m:=7;d:=d-30; end;
    If d>31 then begin m:=8;d:=d-31; end;
    If d>31 then begin m:=9;d:=d-31; end;
    If d>30 then begin m:=10;d:=d-30; end;
    If d>31 then begin m:=11;d:=d-31; end;
    If d>30 then begin m:=12;d:=d-30; end;
    s:=Format('Тражени датум је: %1d.%1d.%4d.', [d,m,g]);
  end;
  Label3.Caption:=s;
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
  Button2.SetFocus;
end;

```

Сачуваћемо програм и извршити га.

• **Саставити програм који одређује разлику између два датума у истој години.**

Година нам је потребна само због преступног дана у фебруару, иначе је довољно само унети дан и месец у оба датума. Зато ћемо уносити годину само код првог датума, а она ће се преписивати у други. Задатак ћемо решити тако што ћемо одредити редне бројеве дана у години првог и другог датума, а затим апсолутну вредност њихове разлике.

Подесићемо карактеристике форме на следећи начин:

```

  BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ
  ГРАНАЊА; ClientHeight = 300; ClientWidth = 400; Font = Palatino
  Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position =
  poScreenCenter.

```

На форму ћемо ставити лателу за наслов, шест едита за унос, две лателе за комуникацију са корисником, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

```

  Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Одузимање датума; Font = 22, Lime; Height = 50; Layout = tlCenter.

```

Едити за унос:

```

  Font = Bold, Black; Left = (190, 240, 290); MaxLength = (2, 2, 4); Top = (80, 130); Width = (40, 40, 60).

```

Латела за испис резултата:

```

  Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.

```

Латела за комуникацију са корисником:

```

  AutoSize = False; Caption = Редни број дана и година; Height = 36; Layout = tlCenter; Left = 50; Top = (80, 130); Width = 140.

```

Тастери за управљање:

```

  Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 10; Top = (10, 140, 270); Width = 120.

```

Сачуваћемо дизајнирану форму у креирани фолдер *RazlikaDD*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```

procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit3.ReadOnly:=false;
      Edit4.Clear;Edit4.ReadOnly:=false;
      Edit5.Clear;Edit5.ReadOnly:=false;
      Edit6.Clear;Edit6.ReadOnly:=false;
      Label3.Caption:='';
      Edit1.SetFocus;
end;

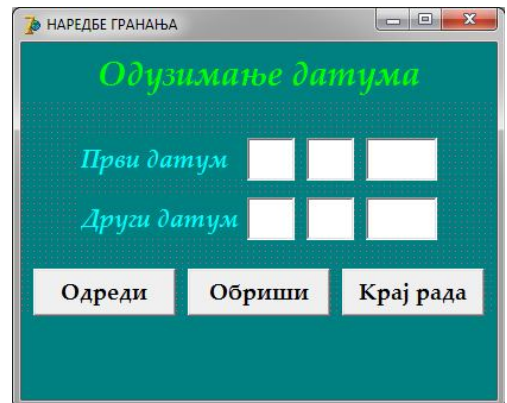
```

Двоклик на тастер *Одреди*:

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    d1,m1,g,d2,m2,p,b:integer;
begin d1:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(d1);
      m1:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(m1);
      g:=StrToIntDef(Edit3.Text,2013);Edit3.Text:=IntToStr(g);Edit6.Text:=IntToStr(g);
      d2:=StrToIntDef(Edit4.Text,31);Edit4.Text:=IntToStr(d2);
      m2:=StrToIntDef(Edit5.Text,12);Edit5.Text:=IntToStr(m2);
      g:=StrToIntDef(Edit6.Text,2013);Edit3.Text:=IntToStr(g);Edit6.Text:=IntToStr(g);
      If (g mod 4=0)and(g mod 100<>0)or(g mod 400=0)
        then p:=1
            else p:=0;
      Case m1 of
        2 : b:=28+p;
        4,6,9,11 : b:=30;
        else b:=31;
      end;
      If (d1>b)or(d1<1)or(m1>12)or(m1<1)

```



```

then s:='Погрешно унет први датум.'
else begin Case m2 of
    2 : b:=28+p;
    4,6,9,11 : b:=30;
    else b:=31;
end;
If (d2>b) or (d2<1) or (m2>12) or (m2<1)
then s:='Погрешно унет други датум.'
else begin Case m1 of
    2 : d1:=d1+31;
    3 : d1:=d1+31+28+p;
    4 : d1:=d1+31+28+p+31;
    5 : d1:=d1+31+28+p+31+30;
    6 : d1:=d1+31+28+p+31+30+31;
    7 : d1:=d1+31+28+p+31+30+31+30;
    8 : d1:=d1+31+28+p+31+30+31+30+31;
    9 : d1:=d1+31+28+p+31+30+31+30+31+31;
    10 : d1:=d1+31+28+p+31+30+31+30+31+31+30;
    11 : d1:=d1+31+28+p+31+30+31+30+31+31+30+31;
    12 : d1:=d1+31+28+p+31+30+31+30+31+31+30+31+30;
end;

If m2> 1 then d2:=d2+31;
If m2> 2 then d2:=d2+28+p;
If m2> 3 then d2:=d2+31;
If m2> 4 then d2:=d2+30;
If m2> 5 then d2:=d2+31;
If m2> 6 then d2:=d2+30;
If m2> 7 then d2:=d2+31;
If m2> 8 then d2:=d2+31;
If m2> 9 then d2:=d2+30;
If m2>10 then d2:=d2+31;
If m2>11 then d2:=d2+30;
s:='Разлика у данима је '+IntToStr(Abs(d2-d1));

end;
end;
Label3.Caption:=s;
Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
Edit4.ReadOnly:=true;Edit5.ReadOnly:=true;Edit6.ReadOnly:=true;
Button2.SetFocus;
end;

```

end;

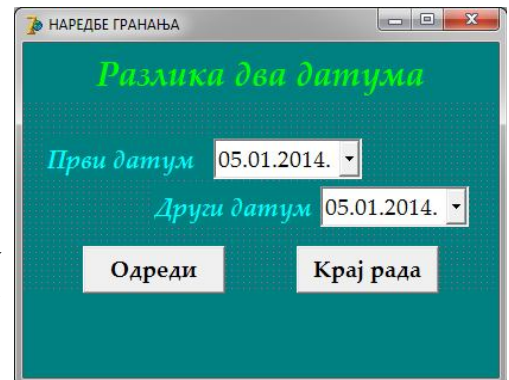
Ако се погрешно унесе један датум морају се поново уписати оба датума, ово је могло бити боље решено. Редни бројеви дана код првог и другог датума одређивани су различитим начинима само да би се приказале обе могућности. Уместо уноса датума помоћу три едита, може се искористити компонента **DateTimePicker** (десета компонента на листићу **Win32**). У том случају нема потребе вршити контролу уноса датума, јер ће датум бити сигурно исправно унет из календара, већ треба само проверити да ли су године у оба датума исте. Функције **DayOf**, **MonthOf** и **YearOf** за издвајање дана, месеца и године из унетог датума, нису стандардне, већ се налазе у посебној библиотеци **DataUtils**, па ову библиотеку треба прикључити програму у одељку **uses** да би се могле користити. Такође, у овој библиотеци постоји и функција **DaysBetween(dat1, dat2)** која одређује разлику између два датума, али ми је нећемо користити јер овај задатак користимо за увежбавање наредби гранања, а овом функцијом би оне биле потпуно непотребне.

Приказаћемо само процедуру за тастер *Одреди*:

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    d1,m1,g1,d2,m2,g2,p:integer;
begin
    d1:=DayOf(DateTimePicker1.Date);
    m1:=MonthOf(DateTimePicker1.Date);
    g1:=YearOf(DateTimePicker1.Date);
    d2:=DayOf(DateTimePicker2.Date);
    m2:=MonthOf(DateTimePicker2.Date);
    g2:=YearOf(DateTimePicker2.Date);
    If g1<>g2
    then s:='Погрешно унети датуми.'
    else begin If (g1 mod 4=0) and (g1 mod 100<>0) or (g1 mod 400=0)
        then p:=1
        else p:=0;
        If m1> 1 then d1:=d1+31;
        If m1> 2 then d1:=d1+28+p;
        If m1> 3 then d1:=d1+31;
        If m1> 4 then d1:=d1+30;
        If m1> 5 then d1:=d1+31;
        If m1> 6 then d1:=d1+30;
        If m1> 7 then d1:=d1+31;

```



```

If m1> 8 then d1:=d1+31;
If m1> 9 then d1:=d1+30;
If m1>10 then d1:=d1+31;
If m1>11 then d1:=d1+30;
If m2> 1 then d2:=d2+31;
If m2> 2 then d2:=d2+28+p;
If m2> 3 then d2:=d2+31;
If m2> 4 then d2:=d2+30;
If m2> 5 then d2:=d2+31;
If m2> 6 then d2:=d2+30;
If m2> 7 then d2:=d2+31;
If m2> 8 then d2:=d2+31;
If m2> 9 then d2:=d2+30;
If m2>10 then d2:=d2+31;
If m2>11 then d2:=d2+30;
s:='Разлика у данима је '+IntToStr(Abs(d2-d1));
end;
Label3.Caption:=s;
end;

```

• **Саставити програм који за унета два броја и рачунску операцију одређује резултат.**

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 400; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, три лабеле за комуникацију са корисником, три едита за унос, лабелу за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Калкулатор; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едити за унос:

**Font = Bold, Black; Left = (60, 140, 170); MaxLength = (0, 1, 0); Top = (80, 120, 160); Width = (120, 30, 120).**

Лабела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.**

Лабела за комуникацију са корисником:

**AutoSize = False; Caption = (a =, Операција, b =); Height = 36; Layout = tlCenter; Left = (20, 30, 130); Top = (80, 120, 160); Width = (20, 120, 20).**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 270; Top = (80, 130, 190); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Kalkulator*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

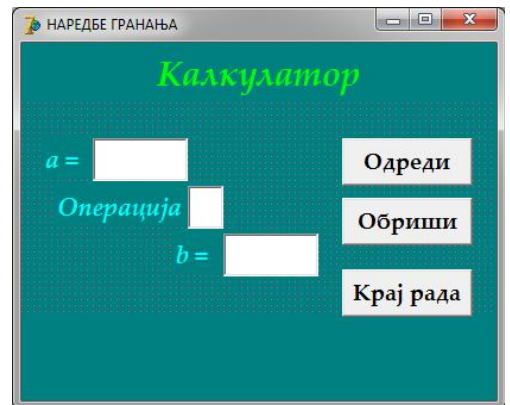
```

procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Label3.Caption:='';
  Edit1.SetFocus;
end;

Двоклик на тастер Одреди:
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    a,b:real;
    o:char;
begin
  a:=StrToFloatDef(Edit1.Text,1); Edit1.Text:=FloatToStr(a);
  o:=Edit2.Text[1];
  b:=StrToFloatDef(Edit3.Text,3); Edit3.Text:=FloatToStr(b);
  s:=Edit1.Text+o+Edit3.Text+'=';
  Case o of
    '+':s:=s+FloatToStr(a+b);
    '-':s:=s+FloatToStr(a-b);
    '*':s:=s+FloatToStr(a*b);
    '/':If b<>0
      then s:=s+FloatToStr(a/b)
      else s:='није дефинисано';
    else s:='непозната операција';
  end;
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
  Button2.SetFocus;
end;

```

Следеће три процедуре кориснику олакшавају употребу програма анализом притиснутог тастера у одговарајући едит. У првом и трећем дозвољено је користити тастере са цифрама, *BackSpace*, *Delete* и



Enter што је дефинисано скупом. У другом едиту дозвољено је користити тастере +, -, \*, /, BackSpace и Enter што је такође, дефинисано скупом. Ако се употреби било који други тастер оквири ће се понашати као да је притиснут тастер Esc, односно, неће се десити ништа.

```
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If key=#13
      then Edit2.SetFocus
      else If not (key in ['0'..'9',#8,#128])
            then key:=#27;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If key=#13
      then Edit3.SetFocus
      else If not (key in ['+', '-', '*', '/', #8])
            then key:=#27;
end;
procedure TForm1.Edit3KeyPress(Sender: TObject; var Key: Char);
begin If key=#13
      then Button1.Click
      else If not (key in ['0'..'9',#8,#128])
            then key:=#27;
end;
```

Задатак се може решити и коришћењем комбобокса за унос операција (на овај начин можемо додати и још неке једноставне двочлане операције). Форма се незнатно мења, а као и програм, па ћемо приказати само процедуру за тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    a,b:real;
begin a:=StrToFloatDef(Edit1.Text,1);
      Edit1.Text:=FloatToStr(a);
      b:=StrToFloatDef(Edit2.Text,3);
      Edit2.Text:=FloatToStr(b);
      s:=Edit1.Text+ComboBox1.Text+Edit2.Text+'=';
      Case ComboBox1.ItemIndex of
        1:s:=s+FloatToStr(a+b);
        2:s:=s+FloatToStr(a-b);
        3:s:=s+FloatToStr(a*b);
        4:If b<>0
           then s:=s+FloatToStr(a/b)
           else s:='није дефинисано';
           else s:='непозната операција';
      end;
      Label3.Caption:=a;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      ComboBox1.Enabled:=false;
      Button2.SetFocus;
end;
```

Било би занимљивије направити верзију овог програма где би се и цифре бројева и ознака рачунске операције уносили кликом на одговарајуће тастере - мало реалнија симулација рада са калкулатором. Форма би могла бити дизајнирана као на слици. Сви улазни и излазни подаци приказују се у едиту коме је кориснику забрањен приступ (*ReadOnly := true*). Али, пошто је ово мало компликованије и алгоритамски и за описивање, нека то буде задатак за оне који желе, могу и знају више.

- **Бројна права је трима тачкама подељена на четири дела. Саставити програм који одређује у ком делу се налази уписани број.**

Подесићемо карактеристике форме на следећи начин:

*BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 400; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.*

На форму ћемо ставити лабелу за наслов, лабелу за комуникацију са корисником, едит за унос, лабелу за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Бројна права и тачка; Font = 22, Lime; Height = 50; Layout = tlCenter.*

Едит за унос:

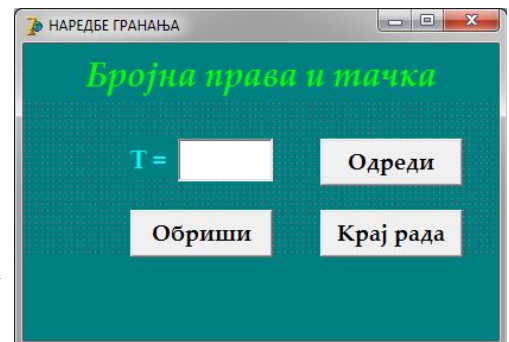
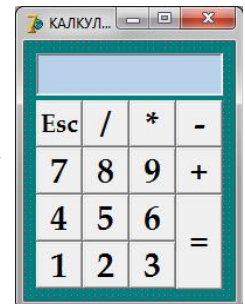
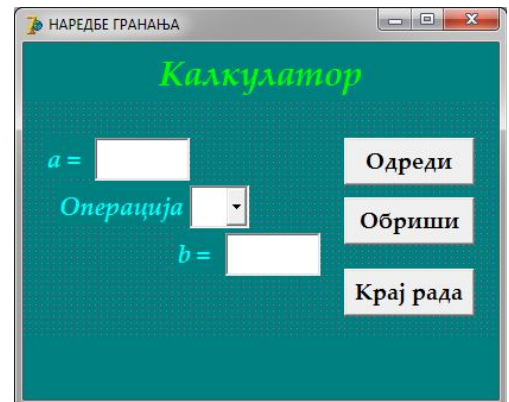
*Font = Bold, Black; Left = 130; Top = 80; Width = 80.*

Лабела за испис резултата:

*Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 70; Layout = tlCenter.*

Лабела за комуникацију са корисником:

*AutoSize = False; Caption = T =; Height = 36; Layout = tlCenter; Left = 90; Top = 80; Width = 40.*





Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 270; Top = (80, 130, 190); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *BPrava*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Label3.Caption:='';
      Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

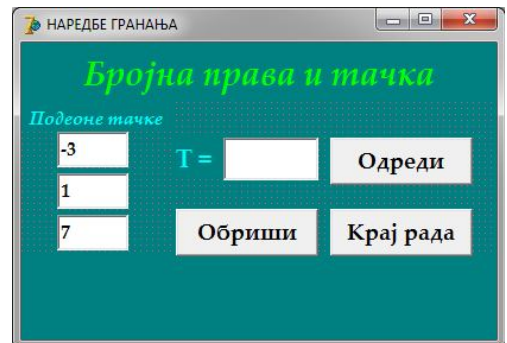
```
procedure TForm1.Button1Click(Sender: TObject);
const a=-3; b=1; c=7;
var s:string;
      d:integer;
begin d:=StrToIntDef(Edit1.Text,4);Edit1.Text:=IntToStr(d);
      s:='Тачка '+Edit1.Text+' ';
      Case d of
        -maxint..a-1:s:=s+'је испред тачке '+IntToStr(a);
        a:           s:=s+'се поклапа са тачком '+IntToStr(a);
        a+1..b-1:   s:=s+'је између тачака '+IntToStr(a)+' и '+IntToStr(b);
        b:           s:=s+'се поклапа са тачком '+IntToStr(b);
        b+1..c-1:   s:=s+'је између тачака '+IntToStr(b)+' и '+IntToStr(c);
        c:           s:=s+'се поклапа са тачком '+IntToStr(c);
        c+1..maxint:s:=s+'је иза тачке '+IntToStr(c);
      end;
      Label3.Caption:=s;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Уместо *Case* наредбе могли смо употребити и комбинацију од неколико *If* наредби, али је овако решење читљивије.

Подеоне тачке морају бити претходно дефинисане константе да бисмо смели да употребимо **Case** наредбу. Ако желимо да се подеоне тачке мењају у програму, онда се овај задатак може решити само помоћу **If** наредбе. У том случају, најпре се подеоне тачке морају уредити у растући поредак, а затим се одређује положај унете тачке на бројној правој.

Приказаћемо само процедуру за тастер *Одреди* у оваквом случају.

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
      a,b,c,d,p:integer;
begin a:=StrToIntDef(Edit2.Text,-3);Edit2.Text:=IntToStr(a);
      b:=StrToIntDef(Edit3.Text,1);Edit3.Text:=IntToStr(b);
      c:=StrToIntDef(Edit4.Text,7);Edit4.Text:=IntToStr(c);
      If a>b
        then begin p:=a;a:=b;b:=p;
              end;
      If a>c
        then begin p:=a;a:=c;c:=p;
              end;
      If b>c
        then begin p:=b;b:=c;c:=p;
              end;
      d:=StrToIntDef(Edit1.Text,4); Edit1.Text:=IntToStr(d);
      s:='Тачка '+Edit1.Text+' ';
      If d=a
        then s:=s+' се поклапа са тачком '+IntToStr(a)
      else If d=b
        then s:=s+' се поклапа са тачком '+IntToStr(b)
      else If d=c
        then s:=s+' се поклапа са тачком '+IntToStr(c)
      else If d<a
        then s:=s+' је испред тачке '+IntToStr(a)
      else If d<b
        then s:=s+' је између тачака '+IntToStr(a)+' и '+IntToStr(b)
      else If d<c
        then s:=s+' је између тачака '+IntToStr(b)+' и '+IntToStr(c)
        else s:=s+' је иза тачке '+IntToStr(c);
      Label3.Caption:=s;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```



- Саставити програм који одређује површину и запремину правилне тростране пирамиде ако су дате висина пирамиде и висина бочне стране.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 380; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лателу за наслов, четири лателе за комуникацију са корисником, два едита за унос, два едита за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Тространа пирамида; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едита за унос:

**Font = Bold, Black; Left = 60; Top = (80, 130); Width = 70.**

Едита за испис резултата:

**Font = Bold, Black; Left = 140; Top = (180, 230); Width = 100.**

Лателе за комуникацију са корисником:

**AutoSize = False; Caption = (H =, h =, P =, V =); Height = 36; Layout = tlCenter; Left = (20, 100); Top = (80, 130, 180, 230); Width = 40.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (140, 140, 250); Top = (80, 130, 250); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *TPiramida*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit4.Clear;Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var h,h1,a,p,v:real;
begin h:=StrToFloatDef(Edit1.Text,1); Edit1.Text:=FloatToStr(h); // висина пирамиде
      h1:=StrToFloatDef(Edit2.Text,3);Edit2.Text:=FloatToStr(h1); // висина бочне стране
      a:=2*Sqrt(3*(h1*h1-h*h)); // основица
      p:=a*a*Sqrt(3)/4+3*a*h1/2;Edit3.Text:=Format('%8.2f',[p]);
      v:=a*a*Sqrt(3)/4*h/3; Edit4.Text:=Format('%8.2f',[v]);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Да би све било коректно, морамо обезбедити да подаци буду исправно унети. То ћемо урадити помоћу следеће две процедуре за контролу уноса података у прва два едита:

```
procedure TForm1.Edit1Exit(Sender: TObject);
begin If StrToFloatDef(Edit1.Text,0)=0
      then begin MessageBox(0,'Висина пирамиде мора бити'+#13+
                           'већа од нуле','ГРЕШКА У УНОСУ',0);
            Edit1.Clear;Edit1.SetFocus;
          end;
end;

procedure TForm1.Edit2Exit(Sender: TObject);
begin If StrToFloatDef(Edit2.Text,0)<=StrToFloatDef(Edit1.Text,0)
      then begin MessageBox(0,'Висина бочне стране мора бити'+#13+
                           'већа од висине пирамиде','ГРЕШКА У УНОСУ',0);
            Edit2.Clear;Edit2.SetFocus;
          end;
end;
```

- Саставити програм који одређује површину и запремину правилне четворостране пирамиде ако је дате висина пирамиде и бочна ивица.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 380; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

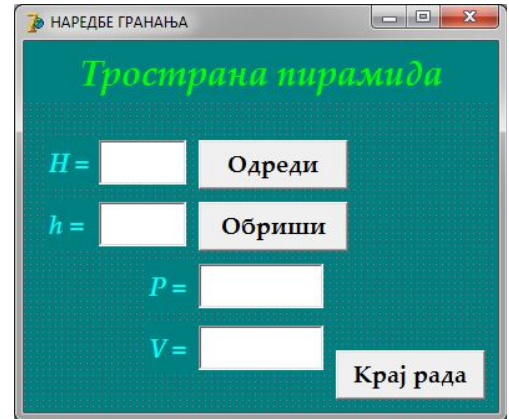
На форму ћемо ставити лателу за наслов, четири лателе за комуникацију са корисником, два едита за унос, два едита за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Четворострана пирамида; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едита за унос:

**Font = Bold, Black; Left = 60; Top = (80, 130); Width = 70.**



Едити за испис резултата:

**Font = Bold, Black; Left = 140; Top = (180, 230); Width = 100.**

Лабеле за комуникацију са корисником:

**AutoSize = False; Caption = (H =, s =, P =, V =); Height = 36; Layout = tlCenter; Left = (20, 100); Top = (80, 130, 180, 230); Width = 40.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (140, 140, 250); Top = (80, 130, 250); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *SPiramida*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit4.Clear;Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var h,s,a,p,v:real;
begin h:=StrToFloatDef(Edit1.Text,1);Edit1.Text:=FloatToStr(h); // висина пирамиде
      s:=StrToFloatDef(Edit2.Text,3);Edit2.Text:=FloatToStr(s); // бочна ивица
      a:=Sqrt(2*(s*s-h*h)); // основица
      p:=a*a+4*Sqrt((s+a/2)*(s-a/2)*a*a/4);Edit3.Text:=Format('%1.2f',[p]);
      v:=a*a*h/3; Edit4.Text:=Format('%1.2f',[v]);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

end;

Да би све било коректно, морамо обезбедити да подаци буду исправно унети. То ћемо урадити помоћу следеће две процедуре за контролу уноса података у прва два едита:

```
procedure TForm1.Edit1Exit(Sender: TObject);
begin If StrToFloatDef(Edit1.Text,0)=0
      then begin MessageBox(0,'Висина пирамиде мора бити'+#13+
                           'већа од нуле','ГРЕШКА У УНОСУ',0);
            Edit1.Clear;Edit1.SetFocus;
            end;
end;
```

end;

```
procedure TForm1.Edit2Exit(Sender: TObject);
begin If StrToFloatDef(Edit2.Text,0)<=StrToFloatDef(Edit1.Text,0)
      then begin MessageBox(0,'Бочна ивица пирамиде мора бити'+#13+
                           'већа од висине пирамиде','ГРЕШКА У УНОСУ',0);
            Edit2.Clear;Edit2.SetFocus;
            end;
end;
```

end;

- **Саставити програм који одређује површину и запремину правилне шестостране пирамиде ако су дате бочна ивица и висина.**

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 380; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, четири лабеле за комуникацију са корисником, два едита за унос, два едита за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Шестострана пирамида; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едити за унос:

**Font = Bold, Black; Left = 60; Top = (80, 130); Width = 70.**

Едити за испис резултата:

**Font = Bold, Black; Left = 140; Top = (180, 230); Width = 100.**

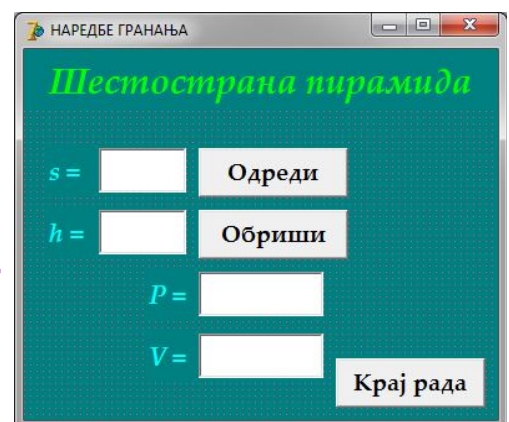
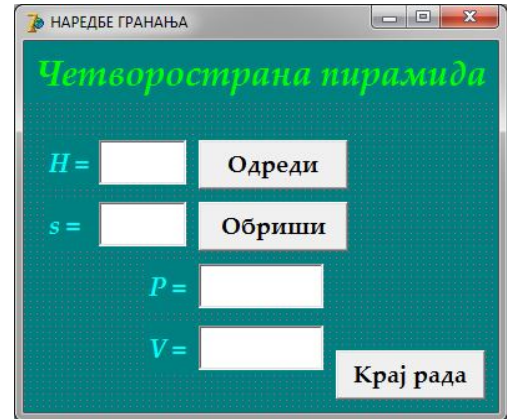
Лабеле за комуникацију са корисником:

**AutoSize = False; Caption = (s =, h =, P =, V =); Height = 36; Layout = tlCenter; Left = (20, 100); Top = (80, 130, 180, 230); Width = 40.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (140, 140, 250); Top = (80, 130, 250); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *SPiramida*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.





Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit4.Clear;Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var h,s,a,p,v:real;
begin
  s:=StrToFloatDef(Edit1.Text,1);Edit1.Text:=FloatToStr(s); // бочна ивица
  h:=StrToFloatDef(Edit2.Text,3);Edit2.Text:=FloatToStr(h); // бочна висина
  a:=2*Sqrt(s*s-h*h); // основица
  p:=3*a*a*Sqrt(3)/2+3*a*h; Edit3.Text:=Format('%1.2f',[p]);
  v:=3*a*a*Sqrt(3)/2*Sqrt(s*s-a*a)/3;Edit4.Text:=Format('%1.2f',[v]);
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
  Button2.SetFocus;
end;
```

Да би све било коректно, морамо обезбедити да подаци буду исправно унети. То ћемо урадити помоћу следеће две процедуре за контролу уноса података у прва два едита:

```
procedure TForm1.Edit1Exit(Sender: TObject);
begin
  If StrToFloatDef(Edit1.Text,0)=0
  then begin
    MessageBox(0,'Висина пирамиде мора бити'+#13+
      'већа од нуле','ГРЕШКА У УНОСУ',0);
    Edit1.Clear;Edit1.SetFocus;
  end;
end;

procedure TForm1.Edit2Exit(Sender: TObject);
var a,s,h:real;
begin
  s:=StrToFloatDef(Edit1.Text,0);
  h:=StrToFloatDef(Edit2.Text,0);
  a:=4*h*h-3*s*s;
  If StrToFloatDef(Edit2.Text,0)<=0
  then begin
    MessageBox(0,'Бочна висина мора бити'+#13+
      'већа од нуле','ГРЕШКА У УНОСУ',0);
    Edit2.Clear;Edit2.SetFocus;
  end
  else If s<=h
  then begin
    MessageBox(0,'Бочна висина мора бити'+#13+
      'мања од бочне ивице','ГРЕШКА У УНОСУ',0);
    Edit2.Clear;Edit2.SetFocus;
  end
  else If a<=0
  then begin
    MessageBox(0,'Бочна висина мора бити већа од уписане',
      'ГРЕШКА У УНОСУ',0);
    Edit2.Clear;Edit2.SetFocus;
  end;
end;
end;
```

- **Саставити програм који одређује површину и запремину ваљка уписаног у купу ако је дата изводница и полупречник основе купе, а висина ваљка и висина купе су у односу  $n:m$ .**

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 380; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лателу за наслов, шест латела за комуникацију са корисником, четири едита за унос, два едита за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Ваљак у купу; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едита за унос:

**Font = Bold, Black; Left = (60, 180); Top = (80, 130); Width = (70, 60).**

Едита за испис резултата:

**Font = Bold, Black; Left = 140; Top = (180, 230); Width = 100.**

Лателе за комуникацију са корисником:

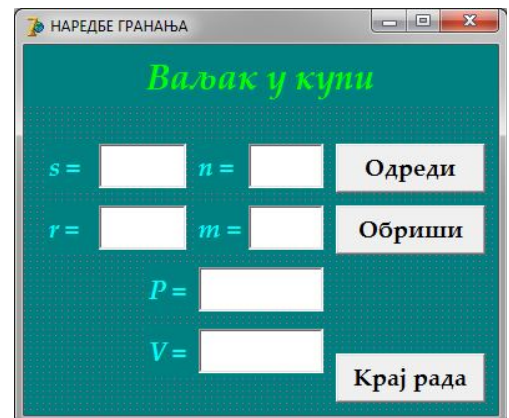
**AutoSize = False; Caption = (s =, r =, n =, m =, P =, V =); Height = 36; Layout = tlCenter; Left = (20, 20, 140, 140, 100, 100); Top = (80, 130, 80, 130, 180, 230); Width = 40.**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 250; Top = (80, 130, 250); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Valjak*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.





Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  Edit3.Clear;Edit3.ReadOnly:=false;
  Edit4.Clear;Edit4.ReadOnly:=false;
  Edit5.Clear;Edit6.Clear;Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var h,r,s,m,n,p,v:real;
begin
  s:=StrToFloatDef(Edit1.Text,4);Edit1.Text:=FloatToStr(s); // изводница
  r:=StrToFloatDef(Edit2.Text,3);Edit2.Text:=FloatToStr(r); // полупречник основе купе
  n:=StrToFloatDef(Edit3.Text,3);Edit3.Text:=FloatToStr(n);
  m:=StrToFloatDef(Edit4.Text,4);Edit4.Text:=FloatToStr(m);
  h:=Sqrt(s*s-r*r); // висина купе
  h:=h*n/m; // висина ваљка
  r:=r*(m-n)/m; // полупречник основе ваљка
  p:=2*r*pi*(r+h);Edit5.Text:=Format('%1.2f',[p]);
  v:=r*r*pi*h; Edit6.Text:=Format('%1.2f',[v]);
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
  Edit3.ReadOnly:=true;Edit4.ReadOnly:=true;
  Button2.SetFocus;
end;
```

Да би све било коректно, морамо обезбедити да подаци буду исправно унети. То ћемо урадити помоћу следеће четири процедуре за контролу уноса података у прва четири едита:

```
procedure TForm1.Edit1Exit(Sender: TObject);
begin
  If StrToFloatDef(Edit1.Text,0)=0
  then begin MessageBox(0,'Изводница купе мора бити'+#13+
    'већа од нуле','ГРЕШКА У УНОСУ',0);
    Edit1.Clear;Edit1.SetFocus;
  end;
end;

procedure TForm1.Edit2Exit(Sender: TObject);
begin
  If StrToFloatDef(Edit2.Text,0)<=StrToFloatDef(Edit1.Text,0)
  then begin MessageBox(0,'Полупречник основе купе мора бити'+#13+
    'мањи од изводнице','ГРЕШКА У УНОСУ',0);
    Edit2.Clear;Edit2.SetFocus;
  end;
end;

procedure TForm1.Edit3Exit(Sender: TObject);
begin
  If StrToFloatDef(Edit3.Text,0)<=0
  then begin MessageBox(0,'n мора бити веће од нуле','ГРЕШКА У УНОСУ',0);
    Edit3.Clear;Edit3.SetFocus;
  end;
end;

procedure TForm1.Edit4Exit(Sender: TObject);
begin
  If StrToFloatDef(Edit4.Text,0)<=StrToFloatDef(Edit3.Text,0)
  then begin MessageBox(0,'m мора бити веће од n','ГРЕШКА У УНОСУ',0);
    Edit4.Clear;Edit4.SetFocus;
  end;
end;
end;
```

- **Саставити програм који одређује да ли су три тачке у равни колинеарне, па ако јесу одређује тачку у средини, а ако нису израчунава висине троугла који оне формирају.**

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 380; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лателу за наслов, шест латела за комуникацију са корисником, шест едита за унос, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Колинеарне; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едита за унос:

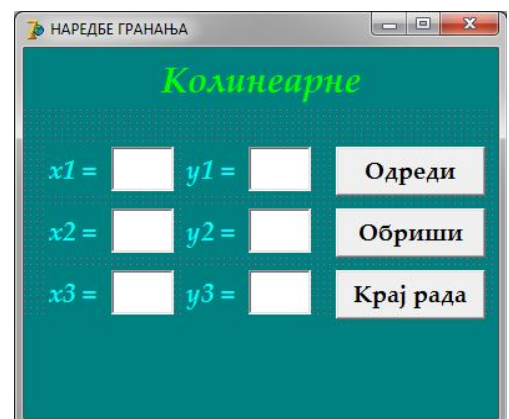
**Font = Bold, Black; Left = (70, 180); Top = (80, 130, 180); Width = 50.**

Латела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 80; Layout = tlCenter.**

Лателе за комуникацију са корисником:

**AutoSize = False; Caption = (x1 =, y1 =, x2 =, y2 =, x3 =, y3 =); Height = 36; Layout = tlCenter; Left = (20, 130); Top = (80, 130, 180); Width = 50.**



Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 250; Top = (80, 130, 180); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Kolinearne*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit3.ReadOnly:=false;
      Edit4.Clear;Edit4.ReadOnly:=false;
      Edit5.Clear;Edit5.ReadOnly:=false;
      Edit6.Clear;Edit6.ReadOnly:=false;
      Label8.Caption:=' ';Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var x1,x2,x3,y1,y2,y3,p,h1,h2,h3:real;
    a,b,c:string;
begin x1:=StrToFloatDef(Edit1.Text,1);Edit1.Text:=FloatToStr(x1);
      y1:=StrToFloatDef(Edit2.Text,3);Edit2.Text:=FloatToStr(y1);
      x2:=StrToFloatDef(Edit3.Text,5);Edit3.Text:=FloatToStr(x2);
      y2:=StrToFloatDef(Edit4.Text,2);Edit4.Text:=FloatToStr(y2);
      x3:=StrToFloatDef(Edit5.Text,4);Edit5.Text:=FloatToStr(x3);
      y3:=StrToFloatDef(Edit6.Text,6);Edit6.Text:=FloatToStr(y3);
{ тачке су на истој правој ако су апсцисе или ординате све три тачке једнаке или ако су на
  истом краку угла са теменом у једној од тачака. Услов у If обухвата све те случајеве јер:
  ако су им једнаке апсцисе  $x1-x3=0=x1-x2$ ; ако су им једнаке ординате  $y1-y2=0=y1-y3$ ;
  ако су на истом краку тангенс угла (теме у првој тачки) је  $(y1-y2)/(x1-x2)=(y1-y3)/(x1-x3)$  }
  If Abs((y1-y2)*(x1-x3))=Abs((y1-y3)*(x1-x2))
    then begin If (x1=x2)and(x2=x3) // уређујемо тачке према растојању од коо-почетка
               then begin If y1>y2 then begin p:=y1;y1:=y2;y2:=p;end;
                           If y1>y3 then begin p:=y1;y1:=y3;y3:=p;end;
                           If y2>y3 then begin p:=y3;y3:=y2;y2:=p;end;
                        end
               else begin If x1>x2 then begin p:=x1;x1:=x2;x2:=p;end;
                           If x1>x3 then begin p:=x1;x1:=x3;x3:=p;end;
                           If x2>x3 then begin p:=x3;x3:=x2;x2:=p;end;
                        end;
               a:='('+FloatToStr(x1)+','+FloatToStr(y1)+')'; // координате прве тачке
               b:='('+FloatToStr(x2)+','+FloatToStr(y2)+')'; // координате друге тачке
               c:='('+FloatToStr(x3)+','+FloatToStr(y3)+')'; // координате треће тачке
               Label8.Caption:='Тачка '+b+' је између'+#13+'тачка 'a+' и '+c+'.'
            end
    else begin h1:=Sqrt(Sqr(x1-x2)+Sqr(y1-y2)); // дужина прве странице троугла
              h2:=Sqrt(Sqr(x2-x3)+Sqr(y2-y3)); // дужина друге странице троугла
              h3:=Sqrt(Sqr(x3-x1)+Sqr(y3-y1)); // дужина треће странице троугла
              p:=Sqrt((h1+h2+h3)*(h1+h2-h3)*(h2+h3-h1)*(h3+h1-h2))/4; // површина - Херон
              a:=Format('%1.3f',[2*p/h1]); // висина на прву страницу
              b:=Format('%1.3f',[2*p/h2]); // висина на другу страницу
              c:=Format('%1.3f',[2*p/h3]); // висина на трећу страницу
              Label8.Caption:='Висине троугла су '+#13+a+'; '+b+' и '+c+'.'
            end;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
      Edit4.ReadOnly:=true;Edit5.ReadOnly:=true;Edit6.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- **Саставити програм који одређује да ли су три тачке у равни колинеарне, па ако нису одређује тачку која са ове три чини темена паралелограма.**

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 380; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лателу за наслов, шест латела за комуникацију са корисником, шест едита за унос, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

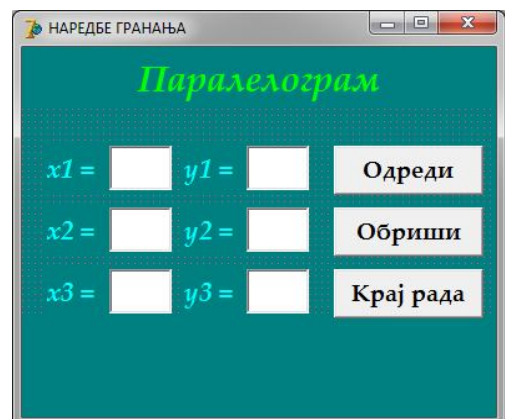
**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Паралелограм; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едити за унос:

**Font = Bold, Black; Left = (70, 180); Top = (80, 130, 180); Width = 50.**

Латела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = Lime; Height = 80; Layout = tlCenter.**



Лабеле за комуникацију са корисником:

```
AutoSize = False; Caption = (x1 =, y1 =, x2 =, y2 =, x3 =, y3 =); Height = 36; Layout = tlCenter; Left = (20, 130);
Top = (80, 130, 180); Width = 50.
```

Тастери за управљање:

```
Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 250; Top = (80, 130, 180); Width = 120.
```

Сачуваћемо дизајнирану форму у креирани фолдер *Paralelogram*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit3.ReadOnly:=false;
      Edit4.Clear;Edit4.ReadOnly:=false;
      Edit5.Clear;Edit5.ReadOnly:=false;
      Edit6.Clear;Edit6.ReadOnly:=false;
      Label8.Caption:='';
      Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var x1,x2,x3,x4,x5,x6,y1,y2,y3,y4,y5,y6,k1,k2,k3,n1,n2,n3:real;
    a,b,c:string;
begin x1:=StrToFloatDef(Edit1.Text,1);Edit1.Text:=FloatToStr(x1);
      y1:=StrToFloatDef(Edit2.Text,3);Edit2.Text:=FloatToStr(y1);
      x2:=StrToFloatDef(Edit3.Text,5);Edit3.Text:=FloatToStr(x2);
      y2:=StrToFloatDef(Edit4.Text,2);Edit4.Text:=FloatToStr(y2);
      x3:=StrToFloatDef(Edit5.Text,4);Edit5.Text:=FloatToStr(x3);
      y3:=StrToFloatDef(Edit6.Text,6);Edit6.Text:=FloatToStr(y3);
      If Abs((y1-y2)*(x1-x3))=Abs((y1-y3)*(x1-x2))
      then Label8.Caption:='Тачке се налазе на једној правој.'
      else begin If x1=x2
                then begin x4:=x3;      y4:=y3+(y1-y2); // y3 померено за наспрамну страницу
                           x5:=x3;      y5:=y3-(y1-y2); // горе или доле, два темена троугла
                           x6:=2*x1-x3;y6:=y1+y2-y3; // треће теме троугла
                end
                else If x2=x3
                then begin x4:=x1;      y4:=y1+(y2-y3);
                           x5:=x1;      y5:=y1-(y2-y3);
                           x6:=2*x2-x1;y6:=y2+y3-y1;
                end
                else If x3=x1
                then begin x4:=x2;      y4:=y2+(y3-y1);
                           x5:=x2;      y5:=y2-(y3-y1);
                           x6:=2*x3-x2;y6:=y3+y1-y2;
                end
                else If y1=y2
                then begin y4:=y3;      x4:=x3+(x1-x2); // x3 померено за наспрамну страницу
                           y5:=y3;      x5:=x3-(x1-x2); // горе или доле, два темена троугла
                           y6:=2*y1-y3;x6:=x1+x2-x3; // треће теме троугла
                end
                else If y2=y3
                then begin y4:=y1;      x4:=x1+(x2-x3);
                           y5:=y3;      x5:=x1-(x2-x3);
                           y6:=2*y2-y1;x6:=x2+x3-x1;
                end
                else If y3=y1
                then begin y4:=y2;      x4:=x2+(x3-x1);
                           y5:=y3;      x5:=x2-(x3-x1);
                           y6:=2*y3-y2;x6:=x3+x1-x2;
                end
                else begin k1:=(y2-y1)/(x2-x1);n1:=y3-k1*x3; // права паралелна првој дужи
                           k2:=(y3-y1)/(x3-x1);n2:=y2-k2*x2; // права паралелна другој дужи
                           k3:=(y2-y3)/(x2-x3);n3:=y1-k3*x1; // права паралелна трећој дужи
                           x4:=(n1-n2)/(k2-k1);y4:=k1*x4+n1; // пресек прве и друге праве
                           x5:=(n1-n3)/(k3-k1);y5:=k1*x5+n1; // пресек прве и треће праве
                           x6:=(n2-n3)/(k3-k2);y6:=k3*x6+n3; // пресек друге и треће праве
                end;
                a:='('+FloatToStr(x4)+','+FloatToStr(y4)+')';
                b:='('+FloatToStr(x5)+','+FloatToStr(y5)+')';
                c:='('+FloatToStr(x6)+','+FloatToStr(y6)+')';
                Label8.Caption:='Тачке '+a+' ; '+b+' и '+c+' могу бити'+#13+
                               'четврто теме паралелограма.'
                end;
                Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
                Edit3.ReadOnly:=true;Edit4.ReadOnly:=true;
                Edit5.ReadOnly:=true;Edit6.ReadOnly:=true;
                Button2.SetFocus;
end;
```

Ако неке две тачке имају исту апсцису или ординату врло једноставно се одређују координате четврте тачке. Представимо тачке у координатном систему и јасно видимо два решења, треће решење захтева допуњавање слике до великог троугла, али није тешко открити га. Значи, постоје три тачке које задовољавају услов задатка. Ако, међутим, никоје две тачке немају исту ни апсцису ни ординату онда морамо применити мало аналитичке геометрије у равни. Нацртаћемо све три дужи које су одређене са ове три тачке, а затим поставити праве паралелне са тим дужима које пролазе кроз трећу тачку (која није на дужи). Таквих правих има три, секу се у три тачке које су решење задатка. Наравно, задатак је изузетно тежак, ако немамо довољно знања из области аналитичке геометрије.

- **Саставити програм који прихвата цену и количину производа и начин плаћања помоћу радио групе, а затим одређује износ за плаћање. Ако се плаћа готовином одобрава се попуст 10%, плаћање чековима је без попушта, а за куповину на кредит рачуна се камата 15%. Предвидети да се проценти могу мењати.**

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 300; ClientWidth = 420; Font = Palatino Linotype, BoldItalic, 16, Aqua; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лателу за наслов, две лателе за комуникацију са корисником, четири едита за унос, радио групу за избор врсте плаћања, лателу за испис резултата и три тастера за управљање апликацијом.

Латела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Обрачун плаћања; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едита за унос:

**Font = Bold, Black; Left = (160, 290, 115, 115); Top = (80, 80, 130, 184); Width = (50, 120, 35, 35).**

Радио група:

**Caption = Плаћање; Columns = 1; Font = 12; Height = 110; Items = [готовина, чекови, кредит]; Left = 5; Top = 110; Width = 105.**

Латела за испис резултата:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = ; Font = 22, Lime; Height = 80; Layout = tlCenter.**

Лателе за комуникацију са корисником:

**AutoSize = False; Caption = (Количина, Цена); Height = 36; Layout = tlCenter; Left = (50, 230); Top = 80; Width = (110, 60).**

Тастери за управљање:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = (160, 290, 290); Top = (130, 130, 180); Width = 120.**

Сачуваћемо дизајнирану форму у креирани фолдер *Naplata*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

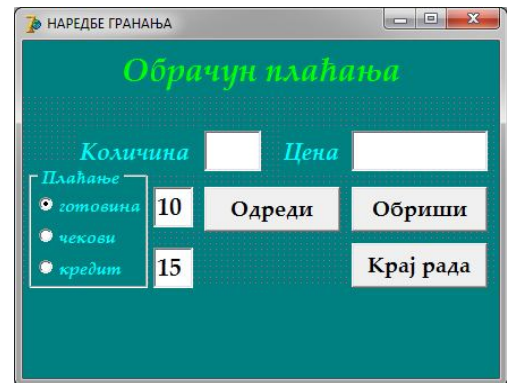
Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=false;
  Edit2.Clear;Edit2.ReadOnly:=false;
  RadioGroup1.ItemIndex:=0;
  Label4.Caption:='';
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure TForm1.Button1Click(Sender: TObject);
var kol,cena,pl:real;
    p,k:integer;
begin
  kol:= StrToFloatDef(Edit1.Text,1);Edit1.Text:=FloatToStr(kol);
  cena:=StrToFloatDef(Edit2.Text,3);Edit2.Text:=FloatToStr(cena);
  p:=StrToIntDef(Edit3.Text,5);Edit3.Text:=IntToStr(p);
  k:=StrToIntDef(Edit4.Text,2);Edit4.Text:=IntToStr(k);
  { ставке у радио групи броје се од 0, па је: 0 - готовина, 1 - чекови, 2 - кредит }
  Case RadioGroup1.ItemIndex of
    0 : pl:=kol*cena*(1-p/100); // 10% значи 10/100, а 1 је 100/100, па 1-10% даје 0,9
    1 : pl:=kol*cena;
    else pl:=kol*cena*(1+k/100); // 15% значи 15/100, а 1 је 100/100, па 1+15% даје 1,15
  end;
  Label4.Caption:='За плаћање '+Format('%1.0m',[pl]);
  { %1.0m (m - money) у формату дефинише испис реалног броја без децимала (.0) са поделом на хиљаде и додатом ознаком валуте која је предефинисана у оперативном систему. Ако не желимо ознаку валуте у испису резултата, онда уместо m треба ставити ознаку n (n - number) да бисмо задржали поделу на хиљаде или f (f - float) ако желимо да у испису буде само реалан број }
  Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
  Button2.SetFocus;
end;
```





- **Саставити програм који за унуту брзину и угао одређује време лета, највећу висину и удаљеност пада пројектила код косог хица. Брзину задавати у  $m/s$  или  $km/h$ , а угао у степенима или у радијанима,**

Да би се задатак што једноставније решио увешћемо радио групе којима ће се дефинисати јединице за брзину и угао. Затим ћемо, када је то потребно, конвертовати брзину у  $m/s$  и/или угао у радијанима да не бисмо имали више формула него што нам је потребно. Улазни подаци ће се, најпре, уносити у едите, а касније ћемо показати и коришћење неких других компоненти којима се олакшава унос података.

Подесићемо карактеристике форме на следећи начин:

**BorderIcons = []; BorderStyle = bsSingle; Caption = НАРЕДБЕ ГРАНАЊА; ClientHeight = 350; ClientWidth = 400; Font = Palatino Linotype, BoldItalic, 13; FormStyle = fsStayOnTop; Position = poScreenCenter.**

На форму ћемо ставити лабелу за наслов, три лабеле за комуникацију са корисником, два едита за унос, две радио групе за избор типа података, три едита за испис резултата и три тастера за управљање апликацијом.

Лабела за наслов:

**Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Коси хицац; Font = 22, Lime; Height = 50; Layout = tlCenter.**

Едита за унос:

**Font = Bold, Black; Left = 270; Top = (75, 145); Width = 110;**

Едита за исписивање резултата:

**Font = Bold, Black; Left = 150; Text = ; ReadOnly = True; Top = (210, 255, 300); Width = 110.**

Радио групе:

**Caption = (Почетна брзина, Угао); Columns = 3; Height = 65; ItemIndex = 1; Items = ([km/h, m/s], [Степени, Радијани]); Left = 5; Top = (55, 125); Width = 385.**

Лабеле за комуникацију са корисником:

**AutoSize = False; Caption = (Време лета, Даљина, Највећа висина); Height = 31; Layout = tlCenter. Left = 15; Top = (210, 255, 300); Width = 135.**

Додаћемо три тастера са следећим карактеристикама:

**Caption = (Одреди, Обриши, Крај рада); Font = Bold; Height = 40; Left = 270; Top = (200, 250, 300); Width = 120.**

Резултати су уоквирени бевелом:

**Height = 140; Left = 5; Shape = bsFrame; Style = bsRaised; Top = 200; Width = 260.**

Сачуваћемо дизајнирану форму у креирани фолдер *KosiHitac*, а затим, двокликом мишем на тастере редом дефинисаћемо све неопходне процедуре.

Двокликом на тастер *Крај рада* дефинисаћемо процедуру којом се прекида извршење програма.

Двоклик на тастер *Обриши*:

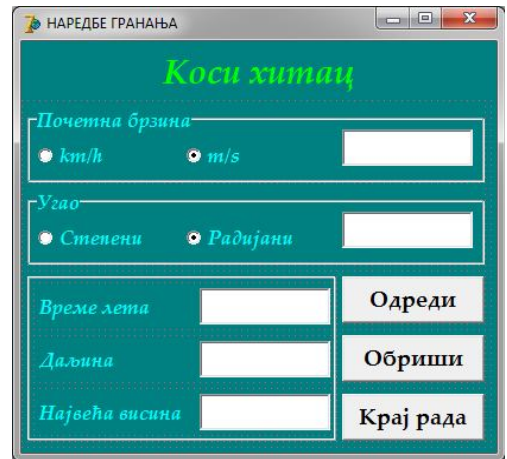
```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Clear;Edit1.ReadOnly:=true;
  Edit2.Clear;Edit2.ReadOnly:=true;
  Edit3.Clear;Edit4.Clear;Edit5.Clear;
  RadioGroup1.Enabled:=true;
  RadioGroup2.Enabled:=true;
  Edit1.SetFocus;
end;
```

Двоклик на тастер *Одреди*:

```
procedure Button1Click(Sender: TObject);
const g = 9.8062;
var V, Fi, T: Real;
begin
  V := StrToFloatDef(Edit1.Text, 0);Edit1.Text:=FloatToStr(V);
  Fi:= StrToFloatDef(Edit2.Text, 0);Edit2.Text:=FloatToStr(Fi);
  If RadioGroup1.ItemIndex=0
    then V:=V/3.6;
  If RadioGroup2.ItemIndex=0
    then Fi:=Fi*Pi/180;
  T:=2*V/g*Sin(Fi);Edit3.Text:=FloatToStr(T)+' s';
  Edit4.Text:=FloatToStr(T*V*Cos(Fi))+' m';
  Edit4.Text:=FloatToStr(T*V/4*Sin(Fi))+' m';
  Edit1.ReadOnly:=true;RadioGroup1.Enabled:=false;
  Edit2.ReadOnly:=true;RadioGroup2.Enabled:=false;
  Button2.SetFocus;
end;
```

Избором јединица из радио група мора се извршити и конверзија података који су претходно унети у одговарајуће едите. За то ћемо написати две процедуре које ће реаговати на клик мишем на радио групу:

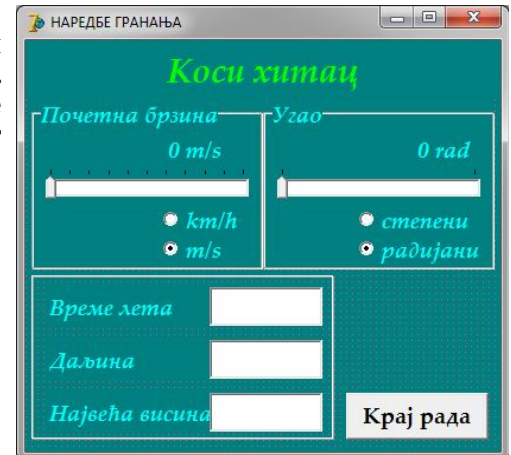
```
procedure TForm1.RadioGroup1Click(Sender: TObject);
var V: real;
begin
  V:=StrToFloatDef(Edit1.Text, 0);
```



```

    If (V<>0) and(RadioGroup1.ItemIndex=1)
        then Edit1.Text:=FloatToStr(V/3.6)
        else Edit1.Text:=FloatToStr(V*3.6);
end;
procedure TForm1.RadioGroup2Click(Sender: TObject);
var Fi:real;
begin Fi:=StrToFloatDef(Edit2.Text,0);
    If (Fi<>0) and(RadioGroup2.ItemIndex=1)
        then Edit2.Text:=FloatToStr(Fi*Pi/180)
        else Edit2.Text:=FloatToStr(Fi/Pi*180);
end;
За уношење почетне брзине и угла можемо користити
компоненту TrackBar (пета компонента на листићу Win32).
Компонента је мало сложенија, па ћемо навести само решење
проблема помоћи ње, а оставићемо за додатну активност
детаљније упознавање и употребу компоненте.
procedure TForm1.Odredi(Sender: TObject);
const g = 9.8062;
var V, Fi, T: Real;
begin V:=TrackBar1.Position;
    If RadioButton1.Checked then V:=V/3.6;
    If RadioButton4.Checked
        then Fi:=TrackBar2.Position/100000
        else Fi:=TrackBar2.Position*Pi/180;T:=2*V/g*Sin(Fi);
    Edit3.Text:=Format('%5.2f s', [T]);
    Edit4.Text:=Format('%5.2f m', [T*V*Cos(Fi)]);
    Edit5.Text:=Format('%5.2f m', [T*V/4*Sin(Fi)]);
end;
procedure TForm1.TrackBar1Change(Sender: TObject);
begin Label2.Caption:=IntToStr(TrackBar1.Position);
    Odredi(Sender);
end;
procedure TForm1.TrackBar2Change(Sender: TObject);
begin if RadioButton4.Checked
    then Label3.Caption:=FloatToStr(TrackBar2.Position/100000)
    else Label3.Caption:=IntToStr(TrackBar2.Position);
    Odredi(Sender);
end;
procedure TForm1.RadioButton1Click(Sender: TObject);
var P: Integer;
begin if Label4.Caption='m/s' then
    begin Label4.Caption:='km/h';
        P:=Round(TrackBar1.Position*3.6);
        TrackBar1.Max :=360;TrackBar1.Frequency:=50;
        TrackBar1.Position:=P; TrackBar1Change(Sender);
    end;
end;
procedure TForm1.RadioButton2Click(Sender: TObject);
var P: Integer;
begin if Label4.Caption = 'km/h' then
    begin Label4.Caption := 'm/s';
        P:=Round(TrackBar1.Position/3.6);
        TrackBar1.Max :=100;TrackBar1.Frequency:=10;
        TrackBar1.Position:=P; TrackBar1Change(Sender);
    end;
end;
procedure TForm1.RadioButton4Click(Sender: TObject);
var P: Integer;
begin if Label5.Caption='st' then
    begin Label5.Caption:='rad';
        P:=Round(TrackBar2.Position*Pi/0.00180);
        TrackBar2.Max :=157079;TrackBar2.Frequency:=10000;
        TrackBar2.PageSize:=10000; TrackBar2.LineSize :=1000;
        TrackBar2.Position:=P; TrackBar2Change(Sender)
    end;
end;
procedure TForm1.RadioButton3Click(Sender: TObject);
var P: Integer;
begin if Label5.Caption='rad' then
    begin Label5.Caption:='st';
        P:=Round(TrackBar2.Position/Pi*0.00180);
        TrackBar2.Max :=90;TrackBar2.Frequency:=10;
        TrackBar2.PageSize:=5; TrackBar2.LineSize :=1;
        TrackBar2.Position:=P; TrackBar2Change(Sender);
    end;
end;
end;

```



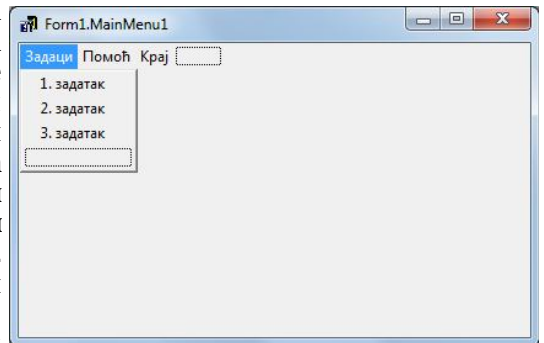
## Неке занимљиве компоненте

Написаћемо програм који користи компоненте које могу бити јако занимљиве, али се код решавања школских задатака ретко користе јер никада нема довољно времена за играње са програмима. Искористићемо три већ урађена задатка, али сада упакована у занимљивије окружење.

**Отворићемо нов пројекат и подесити параметре форме на следећи начин:**

**`BorderIcons = []`; `BorderStyle = bsSingle`; `Caption = ВИЗУЕЛНО ПРОГРАМИРАЊЕ`; `ClientHeight = 800`; `ClientWidth = 600`; `Font = Palatino Linotype, 14`; `FormStyle = fsStayOnTop`; `Position = poScreenCenter`.**

На форму ћемо ставити компоненту **MainMenu**, то је друга компонента на листићу **Standard**. Компонента се користи за креирање главног менија апликације и невидљива је када се програм покрене, зато није битно где смо је поставили, али нека буде у левом горњем углу форме где ће најмање сметати даљем уређивању форме. Ставке менија се додају двокликом на ову компоненту. Отвара се мени едитор у који ћемо уписати систем менија и подменија селектовањем ставке и уписивањем одговарајућег текста у њену **Caption** карактеристику. Дефинисаћемо три ставке менија: *Задаци*, *Помоћ* и *Крај*. Мени *Задаци* имаће подмени са ставкама: *1. задатак*, *2. задатак* и *3. задатак*. Мени *Помоћ* имаће подмени са две ставке: *Упутство* и *Аутор*. Мени *Крај* нема подменије. Свака ставка има своје име помоћу кога ће бити идентификована приликом извршавања апликације. Име се може променити уносом карактеристике **Name**. Када смо креирали жељени систем менија затварамо овај едитор. На врху радне површине прозора појавила се трака са менијем. Сада треба ставке менија повезати са одговарајућим процедурама и функцијама. Кликом на *Крај* дефинисаћемо процедуру којом се прекида рад апликације:



```
procedure TForm1.N3Click(Sender: TObject);
begin Application.Terminate;
end;
```

Додаћемо статусну линију (**StatusBar**, петнаеста компонента на листићу **Win32**) и двокликом на њу отворити едитор у коме ћемо додати три панела кликом на прво дугме у тулбар траци са карактеристикама:

**`Alignment = taCenter`; `Width = (220, 360, 220)`, `Text = ( , ДЕМОНСТРАЦИЈА НЕКИХ ВИЗУЕЛНИХ КОМПОНЕНТИ, )`.**

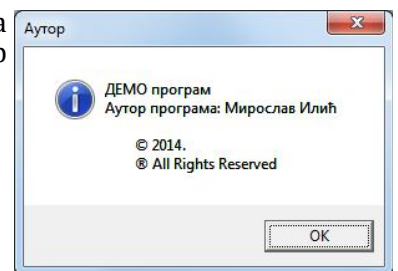
Први и трећи панел имају празну текст карактеристику зато што ће се она попуњавати програмски. На првом панелу ће писати шта се тренутно дешава у апликацији, а трећи панел ће приказивати датум и време.

Додаћемо и компоненту тајмер (**Timer**, прва компонента на листићу **System**), дефинисаћемо промене на сваких 100 милесекунди (`Interval = 100`) и двокликом декларисати и дефинисати процедуру која испишује датум и време на трећем панелу статусне линије:

```
procedure TForm1.Timer1Timer(Sender: TObject);
var d,t:string;
begin t:=TimeToStr(time);
      d:=FormatDateTime('dddd, DD.MM.YYYY. ',date);
      StatusBar1.Panels[2].Text:=d+' '+t;
end;
```

Постављањем миша на ставку *Помоћ* отвара се подмени. Кликом на ставку *Аутор* дефинисаћемо процедуру којом се представља аутор апликације:

```
procedure TForm1.N8Click(Sender: TObject);
begin MessageBox(0,'ДЕМО програм'+#13+
                  'Аутор програма: Мирослав Илић'+#13+#13+
                  '© 2014.'+#13+
                  '© All Rights Reserved',
                  'Аутор',MB_TASKMODAL+MB_ICONASTERISK+MB_OK);
end;
```



Команда **MessageBox(0,<порука>, <наслов>, <врста>)** има различите облике, а ово је један од њих.

- први параметар одређује власника прозора са поруком (0 - припада форми са које смо га отворили),
- <порука> је текст који се испишује у прозору, при чему #13 означава прелазак у нови ред, чиме је омогућено да се порука испишује у више редова,
- <наслов> је текст који се испишује у траци са називом прозора
- <врста> одређује карактеристике поруке (могуће вредности су: **MB\_APPLMODAL**, **MB\_SYSTEMMODAL**, **MB\_TASKMODAL** - ниво зависности дијалога; **MB\_ABORTRETRYIGNORE**, **MB\_OK**, **MB\_OKCANCEL**, **MB\_RETRYCANCEL**, **MB\_YESNO**, **MB\_YESNOCANCEL** - тастери који ће бити приказани у дијалошком прозору; **MB\_ICONEXCLAMATION**, **MB\_ICONWARNING**, **MB\_ICONINFORMATION**, **MB\_ICONASTERISK**, **MB\_ICONQUESTION**, **MB\_ICONSTOP**, **MB\_ICONERROR**, **MB\_ICONHAND** сличица која ће пратити текст у поруци. Дозвољено је комбиновати по једну вредност из сваке групе, вредности се раздвајају +, а редослед није битан. Вредност коју даје затварање овог дијалога, у зависности од тога који су тастери видљиви, може бити: **IDOK**, **IDCANCEL**, **IDABORT**, **IDRETRY**, **IDIGNORE**, **IDYES**, **IDNO** - овде нисмо користили јер је дијалог само информативан).

Кликом на ставку *Упутство* дефинисаћемо процедуру којом се испишује упутство за употребу апликације (овде је приказана процедура која отвара панел са упутством, а упутство је текст датотека коју треба уписати помоћу неког екстерног програма и уградити у мемо поље на панелу):

```
procedure TForm1.N7Click(Sender: TObject);
begin Panel1.Visible:=true;
      Panel1.BringToFront;
end;
```

Да би ова процедура имала смисла, претходно ћемо додати неколико компоненти на форму:

- панел (**Panel**, петнаеста компонента на листићу **Standard**):

```
Align = alClient; BevelInner = bvLowered; BevelOuter = bvLowered;
BevelWidth = 4; BorderStyle = bsSingle; BorderWidth = 2, Caption = ;
Color = clTeal; Visible = false.
```

- лабелу за наслов:

```
Align = alTop; Alignment = taCenter; AutoSize = False; Caption = Упутство за употребу програма; Font =
BoldItalic, 22; Height = 50; Layout = tlCenter
```

- битбатон (**BitBtn**, прва компонента на листићу **Additional**):

```
Caption = ; Height = 40; Glyph = (двокликом се отвара дијалог за избор сличице); Left = 741; Top = 15;
Width = 40
```

- мемо поље (**Memo**, шеста компонента на листићу **Standard**):

```
Align = alClient; BevelInner = bvLowered; BevelKind = bkSoft; BevelOuter = bvLowered; Color =
clMoneyGreen; Font = Italic, 14; Lines = (двокликом се отвара едитор у који треба уписати упутство);
ReadOnly = true.
```

Lines карактеристику можемо попунити садржајем екстерне текстуалне датотеке *Упутство* командом на почетку, приликом креирања апликације или у тренутку отварања панела са упутством):

```
Mem1.Lines.LoadFromFile('Uputstvo.txt');
```

Двокликом на битбатон дефинисаћемо процедуру којом се напушта упутство за употребу:

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin Panel1.Visible:=false;
end;
```

Сада смо учинили све припремне радње којима је апликација динамизирана. Додаћемо сада три панела, за сваки задатак по један, и копираћемо све објекте и процедуре које су везане за те задатке. Наравно, мораћемо да начинимо неке ситне преправке у карактеристикама копираних објеката, пре свега у положају и величини, али и у називима, а тиме и у именима процедура које извршавају неки задатак.

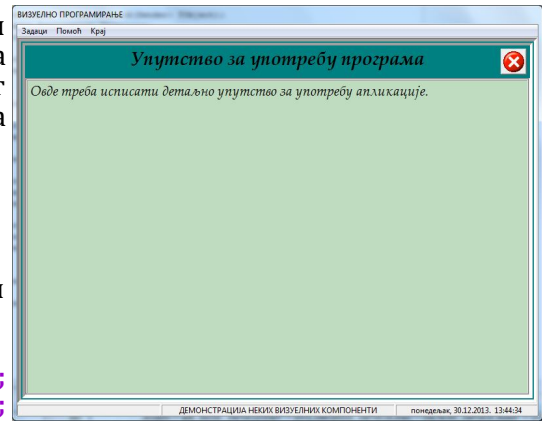
Копираћемо већ постојећи панел тако што ћемо поставити показивач миша на руб панела, кликнути десним тастером и изабрати команду *Edit -> Copy*. Затим ћемо поставити показивач миша на тајмер компоненту, кликнути десним тастером и изабрати команду *Edit -> Paste* (на овај начин се избегава да се нови панел постави на претходни панел, већ се поставља на форму). Обрисаћемо мемо поље са новог панела и дефиницију процедуре на клик на битбатон (у објект инспектору на листићу догађаја обрисаћемо *BitBtn1Click*), а двокликом дефинисати *BitBtn2Click* и променити име процедуре у *Povratak*, а затим ћемо копирати овај панел још два пута, на исти начин. Да бисмо, осим именом, разликовали ова три панела један од другог поставићемо им различите боје: *clMaroon*, *clNavy*, *clPurple*. Одмах можемо дефинисати процедуре које приказују и уклањају ове панеле тако што ћемо из менија *Задаци* бирати једну по једну ставку:

```
procedure TForm1.N4Click(Sender: TObject);
begin Panel2.Visible:=true;
      Panel3.Visible:=false;
      Panel4.Visible:=false;
      StatusBar1.Panels[0].Text:='Први задатак';
end;

procedure TForm1.N5Click(Sender: TObject);
begin Panel3.Visible:=true;
      Panel4.Visible:=false;
      Panel2.Visible:=false;
      StatusBar1.Panels[0].Text:='Други задатак';
end;

procedure TForm1.N6Click(Sender: TObject);
begin Panel4.Visible:=true;
      Panel2.Visible:=false;
      Panel3.Visible:=false;
      StatusBar1.Panels[0].Text:='Трећи задатак';
end;

procedure TForm1.Povratak(Sender: TObject);
begin Panel4.Visible:=false;
      Panel2.Visible:=false;
      Panel3.Visible:=false;
      StatusBar1.Panels[0].Text:='';
end;
```





Овако изгледају панели са задацима:

The figure shows four screenshots of a software application interface for solving math problems. The top-left screenshot is titled "Збир природних бројева у низу" (Sum of natural numbers in a sequence) and shows a user input of 100. Below the input, there are several text boxes displaying results: "Збир првих n природних је 5050", "Збир првих n парних је 10100", "Збир првих n непарних је 10000", "Збир парних до n је 2550", and "Збир непарних до n је 2500". The top-right screenshot is titled "Истиносна вредност формуле" (Truth value of the formula) and shows the logical formula  $\tau = (p \vee q) \wedge r$ . Below the formula, there are dropdown menus for variables p, q, and r, and a text box for the result  $\tau$ . The bottom-left screenshot is titled "Решавање произвољног троугла" (Solving an arbitrary triangle) and shows three diagrams of a triangle with various parameters labeled. The bottom-right screenshot is also titled "Решавање произвољног троугла" and shows a grid of input fields for sides (a, b, c), heights (ha, hb, hc), and angles (alpha, beta, gamma).

Оквир за резултате у првом и трећем задатку је компонента **Bevel** (осма компонента на листићу **Additional**), а листићи са варијантама трећег задатка су делови компоненте **PageControl** (друга компонента на листићу **Win32**). Листићи на **PageControl** компоненти додају се десним кликом на мишу и избором команде **New Page** из отвореног менија, а називи листића се уписују у **Caption** карактеристику. На насловној страни колекције трећег задатка су три сличице троугла у три компоненте **Image** (шеста компонента на листићу **Additional**), а сличице се додају кликом на карактеристику **Picture**, отвара се едитор на коме треба кликнути на дугме **Load**, а затим, на уобичајен начин, изабрати одговарајућу сличицу из отвореног дијалогског прозора. За прекривање основне боје листића искоришћена је лабела у белој боји која је постављена иза сличица и раширена по целом листићу (**Align = alClient**). За унос логичких података у другом задатку користићен је комбиновани оквир (**ComboBox**, једнаеста компонента са листића **Standard**) да би се избегла потреба за коришћењем услова за проверу исправности уноса. Логичка формула у наслову истог задатка исписана је са две лабеле, основна је са нормалним фонтом, а преко ње је провидна лабела са фонтом **Symbol**. Код решавања троугла, да би задатак био коректно решен, коришћена је **If** наредба код израчунавања углова да би се избегло дељење нулом, ако је угао прав. Задатак би радио и без ове наредбе, осим ако је један од углова  $\beta$  или  $\gamma$  прав. Следи комплетан радни део програма.

```
// Збир природних бројева у низу
procedure TForm1.Button1Click(Sender: TObject); // Тастер одреди
var n,s:integer;
begin n:=StrToIntDef(Edit1.Text,9); Edit1.Text:=IntToStr(n);
      s:=n*(n+1) div 2; Edit2.Text:=IntToStr(s);
      s:=n*(n+1); Edit3.Text:=IntToStr(s);
      s:=n*n; Edit4.Text:=IntToStr(s);
      s:=Sqr(n div 2)+n div 2; Edit5.Text:=IntToStr(s);
      s:=Sqr((n+1) div 2); Edit6.Text:=IntToStr(s);
      Edit1.ReadOnly:=true; Button2.SetFocus;
end;
procedure TForm1.Button2Click(Sender: TObject); // Тастер обриши
begin Edit1.Clear; Edit2.Clear; Edit3.Clear;
      Edit4.Clear; Edit5.Clear; Edit6.Clear;
      Edit1.ReadOnly:=false; Edit1.SetFocus;
end;
```

```

// Истиносна вредност логичке формуле
procedure TForm1.Button3Click(Sender: TObject); // Тастер одреди
var p,q,r,t:boolean;
begin p:=StrToBoolDef(ComboBox1.Text,false); ComboBox1.Text:=BoolToStr(p,true);
      q:=StrToBoolDef(ComboBox2.Text,false); ComboBox2.Text:=BoolToStr(q,true);
      r:=StrToBoolDef(ComboBox3.Text,false); ComboBox3.Text:=BoolToStr(r,true);
      t:=(p or q) and r; Edit7.Text:=BoolToStr(t,true);
      Button4.SetFocus;
end;
procedure TForm1.Button4Click(Sender: TObject); // Тастер обриши
begin ComboBox1.Text:=''; ComboBox2.Text:='';
      ComboBox3.Text:=''; Edit1.Clear;
      ComboBox1.SetFocus;
end;
// Решавање произвољног троугла
procedure TForm1.Button5Click(Sender: TObject); // Тастер одреди a,b,c
var a,b,c,h1,h2,h3,t1,t2,t3,p,o:real;
    al,a2,a3,b1,b2,b3,c1,c2,c3:integer;
begin a:=StrToFloatDef(Edit8.Text,5); Edit8.Text:=FloatToStr(a);
      b:=StrToFloatDef(Edit9.Text,3); Edit9.Text:=FloatToStr(b);
      c:=StrToFloatDef(Edit10.Text,4); Edit10.Text:=FloatToStr(c);
      p:=Sqrt((a+b+c)*(-a+b+c)*(a-b+c)*(a+b-c))/4;
      h1:=2*p/a; Edit11.Text:=Format('%7.3f',[a]);
      h2:=2*p/b; Edit12.Text:=Format('%7.3f',[b]);
      h3:=2*p/c; Edit13.Text:=Format('%7.3f',[c]);
      Edit14.Text:=Format('%7.3f',[h1]);
      Edit15.Text:=Format('%7.3f',[h2]);
      Edit16.Text:=Format('%7.3f',[h3]);
      Edit29.Text:=Format('%7.3f',[p]);
      o:=a+b+c; Edit30.Text:=Format('%7.3f',[o]);
      p:=(a*a+b*b-c*c)/(2*a*b);
      t1:=Sqrt(b*b+a*a/4-a*b*p); Edit17.Text:=Format('%7.3f',[t1]);
      t2:=Sqrt(b*b/4+a*a-a*b*p); Edit18.Text:=Format('%7.3f',[t2]);
      If p=0
      then c3:=90*3600
      else c3:=Abs(Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600));
      p:=(a*a-b*b+c*c)/(2*a*c);
      t3:=Sqrt(a*a+c*c/4-a*c*p); Edit19.Text:=Format('%7.3f',[t3]);
      If p=0
      then b3:=90*3600
      else b3:=Abs(Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600));
      a3:=180*3600-b3-c3;
      a1:=a3 div 3600; Edit20.Text:=IntToStr(a1);
      a2:=a3 mod 3600 div 60; Edit21.Text:=IntToStr(a2);
      a3:=a3 mod 60; Edit22.Text:=IntToStr(a3);
      b1:=b3 div 3600; Edit23.Text:=IntToStr(b1);
      b2:=b3 mod 3600 div 60; Edit24.Text:=IntToStr(b2);
      b3:=b3 mod 60; Edit25.Text:=IntToStr(b3);
      c1:=c3 div 3600; Edit26.Text:=IntToStr(c1);
      c2:=c3 mod 3600 div 60; Edit27.Text:=IntToStr(c2);
      c3:=c3 mod 60; Edit28.Text:=IntToStr(c3);
      Edit8.ReadOnly:=true; Edit10.ReadOnly:=true;
      Edit9.ReadOnly:=true;
end;
procedure TForm1.Button6Click(Sender: TObject); // Тастер обриши a,b,c
begin Edit8.Clear; Edit9.Clear; Edit10.Clear;
      Edit11.Clear; Edit12.Clear; Edit13.Clear;
      Edit14.Clear; Edit15.Clear; Edit16.Clear;
      Edit17.Clear; Edit18.Clear; Edit19.Clear;
      Edit20.Clear; Edit21.Clear; Edit22.Clear;
      Edit23.Clear; Edit24.Clear; Edit25.Clear;
      Edit26.Clear; Edit27.Clear; Edit28.Clear;
      Edit29.Clear; Edit30.Clear;
      Edit8.ReadOnly:=false; Edit10.ReadOnly:=false;
      Edit9.ReadOnly:=false; Edit8.SetFocus;
end;
procedure TForm1.Button7Click(Sender: TObject); // Тастер одреди a,b,ha
var a,b,c,h1,h2,h3,t1,t2,t3,p,o:real;
    al,a2,a3,b1,b2,b3,c1,c2,c3:integer;
begin a:=StrToFloatDef(Edit31.Text,4); Edit31.Text:=FloatToStr(a);
      b:=StrToFloatDef(Edit32.Text,3); Edit32.Text:=FloatToStr(b);
      h1:=StrToFloatDef(Edit33.Text,3); Edit33.Text:=FloatToStr(h1);
      p:=a*h1/2; h2:=2*p/b;
      c:=b-Sqrt(a*a-h2*h2); c:=Sqrt(c*c+h2*h2);
      Edit34.Text:=Format('%7.3f',[a]);
      Edit35.Text:=Format('%7.3f',[b]);
      h3:=2*p/c; Edit36.Text:=Format('%7.3f',[c]);
      Edit37.Text:=Format('%7.3f',[h1]);
      Edit38.Text:=Format('%7.3f',[h2]);
      Edit39.Text:=Format('%7.3f',[h3]);
      Edit52.Text:=Format('%7.3f',[p]);
      o:=a+b+c; Edit53.Text:=Format('%7.3f',[o]);

```

```

p:=(a*a+b*b-c*c)/(2*a*b);
t1:=Sqrt(b*b+a*a/4-a*b*p); Edit40.Text:=Format('%7.3f',[t1]);
t2:=Sqrt(b*b/4+a*a-a*b*p); Edit41.Text:=Format('%7.3f',[t2]);
If p=0
  then c3:=90*3600
  else c3:=Abs(Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600));
p:=(a*a-b*b+c*c)/(2*a*c);
t3:=Sqrt(a*a+c*c/4-a*c*p); Edit42.Text:=Format('%7.3f',[t3]);
If p=0
  then b3:=90*3600
  else b3:=Abs(Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600));
a3:=180*3600-b3-c3;
a1:=a3 div 3600; Edit43.Text:=IntToStr(a1);
a2:=a3 mod 3600 div 60; Edit44.Text:=IntToStr(a2);
a3:=a3 mod 60; Edit45.Text:=IntToStr(a3);
b1:=b3 div 3600; Edit46.Text:=IntToStr(b1);
b2:=b3 mod 3600 div 60; Edit47.Text:=IntToStr(b2);
b3:=b3 mod 60; Edit48.Text:=IntToStr(b3);
c1:=c3 div 3600; Edit49.Text:=IntToStr(c1);
c2:=c3 mod 3600 div 60; Edit50.Text:=IntToStr(c2);
c3:=c3 mod 60; Edit51.Text:=IntToStr(c3);
Edit31.ReadOnly:=true; Edit32.ReadOnly:=true;
Edit33.ReadOnly:=true;
end;
procedure TForm1.Button8Click(Sender: TObject); // Тастер обриши a,b,ha
begin Edit31.Clear; Edit32.Clear; Edit33.Clear;
Edit34.Clear; Edit35.Clear; Edit36.Clear;
Edit37.Clear; Edit36.Clear; Edit39.Clear;
Edit40.Clear; Edit41.Clear; Edit42.Clear;
Edit43.Clear; Edit44.Clear; Edit45.Clear;
Edit46.Clear; Edit47.Clear; Edit48.Clear;
Edit49.Clear; Edit50.Clear; Edit51.Clear;
Edit52.Clear; Edit53.Clear;
Edit31.ReadOnly:=false; Edit32.ReadOnly:=false;
Edit33.ReadOnly:=false; Edit31.SetFocus;
end;
procedure TForm1.Button9Click(Sender: TObject); // Тастер одреди a,ha,hb
var a,b,c,h1,h2,h3,t1,t2,t3,p,o:real;
a1,a2,a3,b1,b2,b3,c1,c2,c3:integer;
begin a:=StrToFloatDef(Edit54.Text,4); Edit54.Text:=FloatToStr(a);
h1:=StrToFloatDef(Edit55.Text,3); Edit55.Text:=FloatToStr(h1);
h2:=StrToFloatDef(Edit56.Text,4); Edit56.Text:=FloatToStr(h2);
p:=a*h1/2; b:=2*p/h2;
c:=b-Sqrt(a*a-h2*h2); c:=Sqrt(c*c+h2*h2);
Edit57.Text:=Format('%7.3f',[a]);
Edit58.Text:=Format('%7.3f',[b]);
h3:=2*p/c; Edit59.Text:=Format('%7.3f',[c]);
Edit60.Text:=Format('%7.3f',[h1]);
Edit61.Text:=Format('%7.3f',[h2]);
Edit62.Text:=Format('%7.3f',[h3]);
Edit75.Text:=Format('%7.3f',[p]);
o:=a+b+c; Edit76.Text:=Format('%7.3f',[o]);
p:=(a*a+b*b-c*c)/(2*a*b);
t1:=Sqrt(b*b+a*a/4-a*b*p); Edit63.Text:=Format('%7.3f',[t1]);
t2:=Sqrt(b*b/4+a*a-a*b*p); Edit64.Text:=Format('%7.3f',[t2]);
If p=0
  then c3:=90*3600
  else c3:=Abs(Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600));
p:=(a*a-b*b+c*c)/(2*a*c);
t3:=Sqrt(a*a+c*c/4-a*c*p); Edit65.Text:=Format('%7.3f',[t3]);
If p=0
  then b3:=90*3600
  else b3:=Abs(Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600));
a3:=180*3600-b3-c3;
a1:=a3 div 3600; Edit66.Text:=IntToStr(a1);
a2:=a3 mod 3600 div 60; Edit67.Text:=IntToStr(a2);
a3:=a3 mod 60; Edit68.Text:=IntToStr(a3);
b1:=b3 div 3600; Edit69.Text:=IntToStr(b1);
b2:=b3 mod 3600 div 60; Edit70.Text:=IntToStr(b2);
b3:=b3 mod 60; Edit71.Text:=IntToStr(b3);
c1:=c3 div 3600; Edit72.Text:=IntToStr(c1);
c2:=c3 mod 3600 div 60; Edit73.Text:=IntToStr(c2);
c3:=c3 mod 60; Edit74.Text:=IntToStr(c3);
Edit54.ReadOnly:=true; Edit55.ReadOnly:=true;
Edit56.ReadOnly:=true;
end;
procedure TForm1.Button10Click(Sender: TObject); // Тастер обриши a,ha,hb
begin Edit54.Clear; Edit55.Clear; Edit56.Clear;
Edit57.Clear; Edit58.Clear; Edit59.Clear;
Edit60.Clear; Edit61.Clear; Edit62.Clear;
Edit63.Clear; Edit64.Clear; Edit65.Clear;
Edit66.Clear; Edit67.Clear; Edit68.Clear;

```

```

Edit69.Clear; Edit70.Clear; Edit71.Clear;
Edit72.Clear; Edit73.Clear; Edit74.Clear;
Edit75.Clear; Edit76.Clear;
Edit54.ReadOnly:=false; Edit55.ReadOnly:=false;
Edit56.ReadOnly:=false; Edit54.SetFocus;
end;
procedure TForm1.Button11Click(Sender: TObject); // Тастер одреди ha,hb,hc
var a,b,c,h1,h2,h3,t1,t2,t3,p,o:real;
    a1,a2,a3,b1,b2,b3,c1,c2,c3:integer;
begin h1:=StrToFloatDef(Edit77.Text,4); Edit77.Text:=FloatToStr(h1);
      h2:=StrToFloatDef(Edit78.Text,3); Edit78.Text:=FloatToStr(h2);
      h3:=StrToFloatDef(Edit79.Text,2,4); Edit79.Text:=FloatToStr(h3);
      a:=1/h1; b:=1/h2; c:=1/h3; p:=Sqrt((a+b+c)*(-a+b+c)*(a-b+c)*(a+b-c))/4;
      a:=a/p/2; Edit80.Text:=Format('%7.3f',[a]);
      b:=b/p/2; Edit81.Text:=Format('%7.3f',[b]);
      c:=c/p/2; Edit82.Text:=Format('%7.3f',[c]);
      Edit83.Text:=Format('%7.3f',[h1]);
      Edit84.Text:=Format('%7.3f',[h2]);
      Edit85.Text:=Format('%7.3f',[h3]);
      p:=a*h1/2; Edit99.Text:=Format('%7.3f',[p]);
      o:=a+b+c; Edit98.Text:=Format('%7.3f',[o]);
      p:=(a*a+b*b-c*c)/(2*a*b);
      t1:=Sqrt(b*b+a*a/4-a*b*p); Edit86.Text:=Format('%7.3f',[t1]);
      t2:=Sqrt(b*b/4+a*a-a*b*p); Edit87.Text:=Format('%7.3f',[t2]);
      If p=0
      then c3:=90*3600
      else c3:=Abs(Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600));
      p:=(a*a-b*b+c*c)/(2*a*c);
      t3:=Sqrt(a*a+c*c/4-a*c*p); Edit88.Text:=Format('%7.3f',[t3]);
      If p=0
      then b3:=90*3600
      else b3:=Abs(Round(ArcTan(Sqrt(1-p*p)/p)/pi*180*3600));
      a3:=180*3600-b3-c3;
      a1:=a3 div 3600; Edit89.Text :=IntToStr(a1);
      a2:=a3 mod 3600 div 60; Edit90.Text:=IntToStr(a2);
      a3:=a3 mod 60; Edit91.Text:=IntToStr(a3);
      b1:=b3 div 3600; Edit92.Text:=IntToStr(b1);
      b2:=b3 mod 3600 div 60; Edit93.Text:=IntToStr(b2);
      b3:=b3 mod 60; Edit94.Text:=IntToStr(b3);
      c1:=c3 div 3600; Edit95.Text:=IntToStr(c1);
      c2:=c3 mod 3600 div 60; Edit96.Text:=IntToStr(c2);
      c3:=c3 mod 60; Edit97.Text:=IntToStr(c3);
      Edit77.ReadOnly:=true; Edit78.ReadOnly:=true;
      Edit79.ReadOnly:=true;
end;
procedure TForm1.Button12Click(Sender: TObject); // Тастер обриши ha,hb,hc
begin Edit77.Clear; Edit78.Clear; Edit79.Clear;
      Edit80.Clear; Edit81.Clear; Edit82.Clear;
      Edit83.Clear; Edit84.Clear; Edit85.Clear;
      Edit86.Clear; Edit87.Clear; Edit88.Clear;
      Edit89.Clear; Edit90.Clear; Edit91.Clear;
      Edit92.Clear; Edit93.Clear; Edit94.Clear;
      Edit95.Clear; Edit96.Clear; Edit97.Clear;
      Edit98.Clear; Edit99.Clear;
      Edit77.ReadOnly:=false; Edit78.ReadOnly:=false;
      Edit79.ReadOnly:=false; Edit77.SetFocus;
end;

```

У понуди програмског окружења са којим радимо има још јако пуно занимљивих, а једноставних компоненти, али њихова примена захтева познавање многих наредби које још нисмо научили и много жеље и осећаја, али и времена проведеног уз рачунар на остварењу идеја. Неке од њих ћемо приказати касније, када још нешто научимо и када се ослободимо страха и неверице или на додатној настави. Следе задаци различите тежине које треба урадити да бисмо схватили где смо, на ком нивоу су наше могућности, знање, интересовање.



### Задаци за самосталан рад

- Саставити програм који од два уписана броја исписује мањи и за колико је он мањи.
- Саставити програм који одређује да ли је уписани број дељив са 13.
- Саставити програм који за два уписана цела броја одређује да ли је њихова разлика делилац њиховог збира.
- Саставити програм који одређује да ли је уписани троцифрени број дељив збиром својих цифара.
- Саставити програм који одређује да ли је уписани цео број квадрат неког целог броја и ако јесте исписати тај број, а ако није исписати њему најближи квадрат целог броја.
- Саставити програм који за уписани цео број исписује најближи куб неког целог броја.
- Саставити програм који одређује да ли се уписани број  $a$  налази између уписаних бројева  $b$  и  $c$ .
- Саставити програм који одређује количник већег од бројева  $a$  и  $b$  са мањим од бројева  $c$  и  $d$ .
- Саставити програм који два уписана реална броја  $a$  и  $b$  прво помножи, а затим подели, па исписује мањи резултат.
- Саставити програм који за три уписана броја одређује аритметичку средину и који је од та три броја њој најближи. Ако су два броја једнако близу исписати мањи.
- Саставити програм који од уписаног четвороцифреног броја формира нови тако што цифре које су парне дели са два, а непарне умањује за један.
- Дате су странице  $a$  и  $b$  и дијагонала  $d_1$  паралелограма. Саставити програм који одређује који је то паралелограм (квадрат, правоугаоник, ромб, ромбоид).
- Саставити програм који ће за унети цели број  $a$  испитати да ли је већи, мањи или једнак броју  $b$  и да ли је дељив бројем  $c$ .
- Саставити програм који за унети број од 3 до 6 исцртава одговарајући правилан многоугао.
- Саставити програм који одређује обим, површину и углове троугла ако су дате страница и висина која јој одговара и још једна висина.
- Саставити програм који одређује обим, површину и углове троугла ако су дате страница, висине која јој одговара и један од налеглих углова.
- Саставити програм који одређује обим, површину и углове троугла ако су дате страница и висине и тежишна линија које одговарају тој страници.
- Саставити програм који одређује обим, површину и висине троугла ако су дата два угла и страница насупрот једног од њих.
- Саставити програм који одређује тежишне линије троугла ако су дате све странице.
- Саставити програм који одређује обим и површину троугла ако су дате страница и висина и тежишна линија које јој одговарају.
- Саставити програм који одређује обим и површину троугла ако су дати обими уписаног и описаног круга и једна страница му је двапута већа од разлике друге две.
- Саставити програм који одређује обим и површину круга описаног око троугла са страницама  $a$ ,  $b$ ,  $c$ .
- Саставити програм који одређује обим и површину круга уписаног у троугао са страницама  $a$ ,  $b$ ,  $c$ .
- Саставити програм који на основу унетих странице и дијагонале паралелограма одређује да ли се ради о ромбу, ромбоиду, квадрату или правоугаонику.
- Уноси се четвороцифрени број, а затим се од његових цифара формирају најмањи и највећи могући број. Саставити програм који одређује да ли је већи број целобројни умножак мањег.
- Уноси се четвороцифрени број, а затим се од његових цифара формирају најмањи и највећи могући број. Саставити програм који одређује да ли мањи број може бити целобројни количник већег.
- Уноси се троцифрени број. Саставити програм који од његових цифара формира све могуће двоцифрене бројеве код којих прва цифра није мања од друге и уређује их у неоппадајући поредак.

- Уносе се три цела броја  $a$ ,  $b$  и  $c$ . Саставити програм који одређује највећи и да ли је најмањи делилац бар једног од остала два.
- Уносе се три цела броја  $a$ ,  $b$  и  $c$ . Саставити програм који их уређује у неоппадајући поредак, а затим одређује да ли је двоструки средњи мањи, већи или једнак збиру крајњих бројева.
- Уносе се три цела броја  $a$ ,  $b$  и  $c$ . Саставити програм који одређује најмањи и да ли је највећи дељив бар једним од остала два.
- Уносе се три цела броја  $a$ ,  $b$  и  $c$ . Саставити програм који их уређује у нерастући поредак, а затим одређује да ли је аритметичка средина крајњих мања, већа или једнака средњем броју.
- Саставити програм који одређује која је од две тачке у координатној равни ближа координатном почетку.
- Саставити програм који одређује која је од две тачке у координатном простору ближа координатном почетку.
- Саставити програм који проверава да ли три унете дужине могу бити странице троугла, па ако могу одређује најдужу, површину или обим троугла. Избор резултата се врши помоћу радио групе.
- Саставити програм који за дати угао у степенима одређује квадрант у коме се налази.
- Саставити програм који за сваки унети број од 0 до 9 исписује име тог броја.
- Саставити програм који за унету нумеричку исписује описну оцену.
- Направите програм који за унету вредност времена (0-24) одређује доба дана (0-поноћ, 1-12 - пре подне, 12 - подне, 13 - 20 после подне, 21-0 вече)
- Хороскопски знаци слажу се са следећим датумима : Водолија (20.1 - 18.2); Рибe (19.3 - 20.3); Ован (21.3 - 19.4); Бик (20.4 - 20.5); Близанци (21.5-21.6) ; Рак (22.6-22.7); Лав (23.7-22.8) ; Девица (23.8-22.9); Вага (23.9-22.10); Шкорпија (23.10-22.11); Стрелац (23.11-21.12) ; Јарац (22.12-19.1). Саставити програм који на основу унетог датума рођења (дан и месец) исписује ком хороском знаку особа припада.
- Уносе се бројеви све док се не унесе 0. Саставити програм који исписује број унетих бројева и колико је било позитивних.
- Уносе се бројеви све док се не унесе 0. Саставити програм који одређује број унетих парних бројева и њихову аритметичку средину
- Саставити програм који прихвата дужину дужи, а затим, избором помоћу радио групе, одређује обим и површину: круга, једнакостраничног троугла, квадрата или шестоугла.
- Саставити програм који који уз помоћ два чекбокса, који дозвољавају или забрањују упис дужина страница  $a$  и  $b$ , одређује обим и површину квадрата или правоугаоника.
- Саставити програм који помоћу чекбокса одређује која два од четири податка ( $a$ ,  $h_a$ ,  $H$ ,  $s$ ) се уписује да би се одредиле површина и запремина праве правилне тростране пирамиде.