

Delphi

Uwaga odnośnie korzystania z instrukcji

W instrukcji w treści zadań są zapisane odnośniki np. **(1)** **(2)** **(3)** itp.

Oznacz to, że musisz przeczytać z instrukcji treść tego odnośnika jak wykonać polecenie. Wy tłumaczenia znajdują się na **stronie 36**.

Treść przykładu 1

Wykonaj:

- 1)Przepisz temat.
- 2)Rozwiąż przykład na podstawie tekstu jak to zrobić zamieszczonego poniżej.
- 3)Zapisz w zeszycie wytłumaczenie:
 - suma.Caption:=IntToStr(s)
 - Funkcja StrToInt()
 - Close
 - Funkcja IntToStr()
- 4)Zapisz w zeszycie jakie plik utworzył delphi oraz ich znaczenie.

Temat: Obliczanie sumy ciągu po wczytaniu kolejnych wyrazów tego ciągu.

Uwagi na temat działania(treści zadania) programu

Widoczne będzie okienko do wczytywania danych (kolejnych wyrazów tego ciągu). Po wpisaniu kolejnego elementu ciągu wciskamy przycisk Wprowadź. Wczytywanie danych (elementów ciągu) następować będzie, aż naciśnięty będzie przycisk Koniec. W specjalnym okienku wyświetlana będzie aktualna suma ciągu (za każdym wczytanym elemencie ciągu).

Rozwiązanie:

1.)Załącz na dysku twardym C w katalogu głównym folder do, którego będziesz zapisywać efekty swojej pracy. O nazwie Twoje nazwisko np. kowalski. Każde zadanie oraz przykład zapisuj w osobnym folderze np. P1 dla przykładu 1 oraz np. Z1 dla zadania 1.

Gdy chcesz zabrać efekty pracy do domu to musisz dokonać kopiowania całego folderu (Delphi tworzy kilka rodzajów plików i wszystkie są potrzebne).

2.Wgraj Delphi (ikona startowa programu znajduje się w menu Start w górnej jego części) a nowy projekt utwórz się sam Projekt **(1)**

3.Zobacz w oknie edycji programów jak wygląda program nowego Projektu (okno to znajduje się pod oknem Projektu, więc przesun okno projektu i zobaczysz okno edycji programu)

4.Zmień nazwę okna Formularza (projektu) nadaj *nazwę nazwisko_numer_przykladu/zadania (2) np. kowalski_p1*

5.Aby zachować zadanie(przykład) wybierz Save Project AS. Zachowaj do Twojego folderu Unit po nazwą *nr_numer_z_dziennika_u* np. nr_4_u . (unit) **(3)**

Zachowaj do Twojego folderu Projekt po nazwą *nr_numer_z_dziennika_p* np. nr_4_p (projekt) **(3)**

6.Przeczytaj w jaki sposób wgrywa się program **(16)**

7.Wstaw pole wprowadzania danych do Projektu nadaj nazwę zmiennej **zm1** oraz **brak tekstu** w okienku do wczytywania danych podczas uruchamiania programu (czyli nic nie powinno być w komórce Text) **(4)**

8.Wstaw dwa pola wyświetlające tekst **(5)**

pierwsze pole z komunikatem SUMA CIĄGU czyli:

- Name etykieta1
 - Caption SUMA CIĄGU
 - Font jak poniżej
- | | |
|------------|-----------|
| krój pisma | Arial |
| wielkość | 22 |
| kolor | granatowy |

drugie pole z komunikatem bez komunikatu z wartością 0 (zero)

- Name suma
- Caption 0 (zero)
- Font jak poniżej

krój pisma	Arial
wielkość	18
kolor	czerwony

Wstaw pola jedno za drugim tak aby otrzymać napis np.

SUMA CIĄGU 0

9. Wstaw przycisk **Wprowadź** i **Koniec** (6)

pierwsze przycisk z napisem Wprowadź czyli:

- Name przycisk1
 - Caption Wprowadź
 - Font jak poniżej
- | | |
|------------|-------|
| krój pisma | Arial |
| wielkość | 20 |

drugie przycisk z napisem Koniec czyli:

- Name przycisk2
 - Caption Koniec
 - Font jak poniżej
- | | |
|------------|-------|
| krój pisma | Arial |
| wielkość | 20 |

10. Wprowadź akcje przypisaną do przycisku Wprowadź (6)

Kliknij dwa razy szybko przycisk Wprowadź i wpisz brakujące linie procedury tak aby otrzymać

```
procedure TForm1.przycisk1Click(Sender: TObject);
begin
    s:=s+StrToInt(zm1.text);
    suma.Caption:= IntToStr(s)
end;
```

wytlmaczenie:

suma.Caption:=IntToStr(s)	Wprowadza do pola suma wartości ze zmiennej s
Funkcja StrToInt()	zamienia tekst na liczbę.
Close	zamyka program
Funkcja IntToStr()	zamienia liczbę na tekst

11. Wprowadź akcje przypisaną do przycisku Koniec (6)

Kliknij dwa razy szybko przycisk Koniec i wpisz brakujące linie procedury tak aby otrzymać

```
procedure TForm1.przycisk2Click(Sender: TObject);
begin
    Close
end;
```

12. Ustawianie wartości początkowych

Ustaw wartość początkową dla zmiennej s na zero. Tak aby otrzymać metodę.

Poprzez : wybierz TForm1 w okienku Object Inspector, następnie zakładkę Events a w niej kliknij dwa razy szybko białe okienko obok zdarzenia OnCreate i dopisz wartości początkowe s:=0; a otrzymasz procedurę jak poniżej

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    s:=0;
end;
```

13. Zadeklaruj zmienną s w następujący sposób

```
var
    Form1: TForm1;
    s:integer;
```

14. Dokonaj wykonania programu (8)

15. Dokonaj kompilacji na dysk programu (9)

Treść zadania 1

Napisz program, który będzie obliczał sumę odwrotności kwadratów dwóch wielkości X i Y wczytanych w okienkach. Użyj przycisków **Wprowadź** oraz **Koniec**.

- Oblicz i zapisz w zeszycie sumę odwrotności kwadratów
Użyj kalkulatora systemowego
dla licz:

X=liczba_liter_imienia
Y=liczba_liter_nazwiska

$$SUMA = \frac{1}{X * X} + \frac{1}{Y * Y}$$

- Wygląd formularza (Twój powinien być zbliżony):

Obliczanie sumy odwrotności kwadratów			
	X=		Y=
sumy odwrotności kwadratów=			
Wprowadź			Koniec.

- Nazwa Unitu i projektu z trzema literami nazwiska z1_kow_u oraz z1_kow_p
- Nazwa okienka z1_nazwisko ucznia np. z1_kowalski
- Nazwy zmiennych z trzema literami np. z1_kow
- Do zamiany zmiennych Real na tekstowe użyj np. s1:=CurrToStr(s);
- Do zamiany zmiennych tekstowe na Real użyj np. s1:=StrToCurr(s);
- Opisz w zeszycie:
Funkcja StrToCurr()
Funkcja CurrToStr()
- Zapisz w zeszycie tę część programu, która wykonuje obliczenia czyli:
Od:

```
procedure TForm1.przycisk1Click(Sender: TObject);
begin
    .....
    .....
    .....
end;
```

Treść zadania 2

Wykonaj:

- Nazwa Unitu i projektu z trzema literami nazwiska z2_kow_u oraz z2_kow_p
- Nazwa okienka (formularza) z1_nazwisko ucznia np. z2_kowalski
- Nazwy zmiennych z trzema literami np. a_kow, b_kow, c_kow

Napisz program, który będzie rozwiązywał równanie kwadratowe w postaci ogólnej. Współczynniki A B C wczytywane w okienkach. Użyj przycisków Wprowadź oraz Koniec.

- Zapisz w zeszycie tę część programu, która wykonuje obliczenia czyli:
Od:

```
procedure TForm1.przycisk1Click(Sender: TObject);
begin
    .....
    .....
    .....
end;
```

Zadanie 1 indywidualne --> zapytaj nauczyciela

Treść przykładu 2

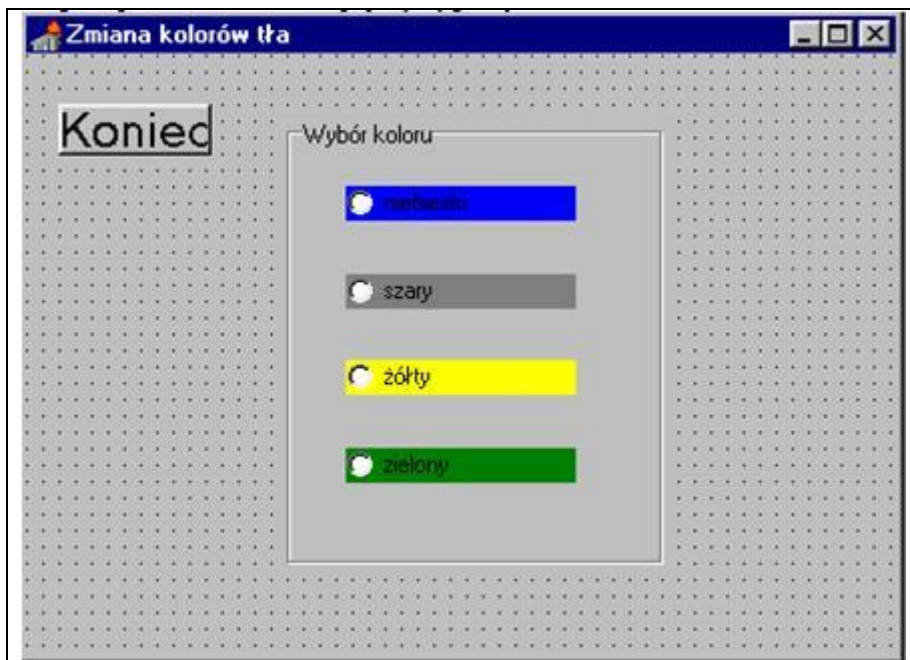
Wykonaj:

- Nazwa Unitu i projektu z trzema literami nazwiska p2_kow_u oraz p2_kow_p
- Nazwa okienka (formularza) *zmiana kolorów tła_nazwisko_ucznia*
- Zapisz temat.
- Przerysuj odręczny szkic ikon oraz zapisz co one oznaczają:
Align ikonę
Align left edges
Space equally vertically
Center horizontally

Temat:

Program z wykorzystaniem okrągłych przycisków wyboru ujętych w grupę do sterowania wyborem koloru tła formularza.

Program powinien mieć następujący wygląd:



Rozwiązanie:

1. Utwórz nowy Projekt (1)
2. Zmień nazwę okna Formularza (projektu) nadaj nazwę *zmiana kolorów tła_nazwisko_ucznia* (2)
3. Nazwa Unitu i projektu z trzema literami nazwiska p2_kow_u oraz p2_kow_p. Nagranie do nowego folderu.
4. Wstaw przycisk **Koniec** (6)

czyli:

- Name przycisk1
- Caption Koniec
- Font jak poniżej
 - krój pisma Arial
 - wielkość 17

5. Wprowadź akcje przypisaną do przycisku Koniec (6)

Kliknij dwa razy szybko przycisk Koniec i wpisz brakujące linie procedury tak aby otrzymać

```
procedure TForm1.przycisk1Click(Sender: TObject);
begin
  Close;
end;
```

6. Wstaw w formularzu cztery okrągłe przyciski wyboru (`RadioButton`) (10) w losowo wybranych miejscach formularza.

Własności ustawiaj w okienku Object Inspector.

Ustawiaj następujące własności:

Pierwszy przycisk:

- Name przycisk1_o_kow --> trzy pierwsze litery nazwiska
- Caption żółty
- Color clYellow (gdy wpisujesz nazwę koloru to kliknij wyraz Color we właściwość Color, uzyskasz listę kolorów czyli klikij strzałkę lisy i wybierz kolor)

drugi przycisk:

- Name przycisk2_o_kow --> trzy pierwsze litery nazwiska
- Caption szary
- Color clGray

trzeci przycisk:

- Name przycisk3_o_kow --> trzy pierwsze litery nazwiska
- Caption niebieski
- Color clBlue

czwarty przycisk:

- Name przycisk4_o_kow --> trzy pierwsze litery nazwiska
- Caption zielony
- Color clGreen

7. Wyrównaj przyciski (13) czterech przycisków z pytania poprzedniego w tym celu :

- zaznacz cztery przyciski jako blok
- wybierz z okienka Align ikonę Align left edges (wyrównanie w pionie). Wytlumaczenia znaczenia ikony uzyskasz poprzez najechanie na ikonę)
- wybierz ikonę Space equally vertically (wyrównanie odstępu pomiędzy elementami)
- wybierz Center horizontally in window oraz Center horizontally in window (wyrównanie w oknie formularza w poziomie oraz pionie)

8. Zaprogramuj działanie (akcje) przypisane do czterech przycisków. Działaniem tym jest nadanie koloru dla tła okna programu (formularza). Nadanie koloru polega na przypisaniu do zmiennej `Form1.Color` określonego koloru tła.

dwa razy szybko kliknij przycisk i wpisz działanie (akcje) aby otrzymać:

```
procedure TForm1.przycisk1_oClick(Sender: TObject);
```

```
begin
```

```
  Form1.Color:=ClYellow;
```

```
end;
```

Dla pozostałych przycisków podobnie lecz inne kolory

.

9. Ujmij okrągłe przyciski (`RadioButton`) z poprzednich pytań w grupę (`RadioGroup`) w tym celu :

- zaznacz zdefiniowane cztery przyciski jako blok i wytnij je czyli `Ctrl+X` (konieczne jest wycięcie ponieważ w pierwszej kolejności należy wstawić `RadioGroup` a dopiero później `RadioButton`)
 - wstaw grupę `RadioGroup1` poprzez wybór odpowiedniej ikony z menu
 - zapisz Caption dla `RadioGroup` jako Wybór koloru
 - wstaw `RadioButton` przechowywane w skrytce poprzez `Ctrl+V`
- dokonaj odpowiedniego umiejscowienia `RadioGroup` (patrz wygląd zadania powyżej)

10. Dokonaj wykonania programu (8)

11. Dokonaj kompilacji na dysk programu (9)

Treść przykładu 2a

Temat: Przykład demonstruje użycie dwóch współdziałających ze sobą RadioGroup.

Treść zadania.

Poniżej znajduje się formularz, który będziesz wykonywał. Przykład wykonuj etapami wg opisu.

Rozwiązanie:

1. Utwórz nowy Projekt (1)
2. Zmień nazwę okna Formularza (projektu) nadaj nazwę Przewidywacz_nazwisko_ucznia (2)
3. Nazwa Unitu i projektu z dwoma literami nazwiska p2a_ko_u oraz p2a_ko_p. Nagranie do nowego folderu.
4. Wstaw przycisk **Koniec programu** (6)

czyli:

- Name przy_nazwisko_ucznia np. przy_kowalski
- Caption Koniec
- Font jak poniżej
 - krój pisma Arial
 - wielkość 14

5. Wprowadź akcje przypisaną do przycisku Koniec (6)

Kliknij dwa razy szybko przycisk Koniec i wpisz brakujące linie procedury tak aby otrzymać

```
procedure TForm1.przycisk1Click(Sender: TObject);
```

```
begin
```

```
  Close
```

```
end;
```

6. Wstaw dwa Labely z następującymi napisami:

Label1	Przepowiem Tobie długość twojego życia
Label2	tutaj będzie przepowiednia o długości Twojego życia

Ustaw wielkości liter oraz ich kolor.

7. Wstaw dwa RadioGroup. Zmień im kolor tła na podobne jak na formularzu w treści zadania, zmiana koloru poprzez zmianę właściwości obiektu Color, zmiana opisu RadioGroup poprzez wpis do właściwości obiektu Caption .

8. Dodaj opcje do obu RadioGroup poprzez zmianę właściwości **Items**. Kliknij po prawie stronie Items, pojawia się trzy kropki i teraz kliknij te kropki a otrzymasz okno. W tym okienku wpisz opcje dla RadioGroup czyli:

kobieta

mężczyzna

Opcje wpisz jedna pod drugą.

9. Wykonaj obsługę wyboru opcji Radio poprzez kliknięcie pierwszego RadioGroup, uzyskasz okienko do wpisywania kodu programu i wpisz aby uzyskać:

```
procedure TPrzewidywacz.RadioGroup1Click(Sender: TObject);
begin
  if RadioGroup1.ItemIndex=0 then
    begin
      label1.caption:='Niepełne dane wybierz opcję po prawej stronie';
    end;
  if RadioGroup1.ItemIndex=1 then
    begin
      label1.caption:='Niepełne dane wybierz opcję po prawej stronie';
    end;
  if (RadioGroup1.ItemIndex=0)and(RadioGroup2.ItemIndex=0) then
    begin
      label1.caption:='Bedziesz żyła krócej niż 40 lat';
    end;
  if (RadioGroup1.ItemIndex=1)and(RadioGroup2.ItemIndex=0) then
    begin
      label1.caption:='Bedziesz żył krócej niż 30 lat';
    end;
  if (RadioGroup1.ItemIndex=0)and(RadioGroup2.ItemIndex=1) then
    begin
      label1.caption:='Bedziesz żyła dłużej niż 80 lat';
    end;
  if (RadioGroup1.ItemIndex=1)and(RadioGroup2.ItemIndex=1) then
    begin
      label1.caption:='Bedziesz żył dłużej niż 70 lat';
    end;
end;
```

kliknij drugiego RadioGroup i wpisz aby uzyskać:

```
procedure TPrzewidywacz.RadioGroup2Click(Sender: TObject);
begin
  if RadioGroup2.ItemIndex=0 then
    begin
      label1.caption:='Niepełne dane wybierz opcję po lewej stronie';
    end;
  if RadioGroup2.ItemIndex=1 then
    begin
      label1.caption:='Niepełne dane wybierz opcję po lewej stronie';
    end;
  if (RadioGroup1.ItemIndex=0)and(RadioGroup2.ItemIndex=0) then
    begin
      label1.caption:='Bedziesz żyła krócej niż 40 lat';
    end;
  if (RadioGroup1.ItemIndex=1)and(RadioGroup2.ItemIndex=0) then
    begin
      label1.caption:='Bedziesz żył krócej niż 30 lat';
    end;
  if (RadioGroup1.ItemIndex=0)and(RadioGroup2.ItemIndex=1) then
    begin
      label1.caption:='Bedziesz żyła dłużej niż 80 lat';
    end;
  if (RadioGroup1.ItemIndex=1)and(RadioGroup2.ItemIndex=1) then
    begin
      label1.caption:='Bedziesz żył dłużej niż 70 lat';
    end;
end;
```


end;

10. Dopisz trzecią RadioGroup z dwoma opcjami pasującymi do dwóch poprzednich RadioGroup. Wykonaj obsługę dla trzech RadioGroup z sensownymi prognozami długości życia.

Uwaga 1:

Do obsługi stosuj:

RadioGroup1.ItemIndex

pamiętaj że pierwsza pozycja z RadioGroup ma wartość 0, druga 1 itp.

Uwaga 2:

Gdy chcesz aby jedno Radio było aktywne po uruchomieniu programu to we właściwościach ItemIndex zamiast liczby (-1) wpisz numer opcji, która ma być aktywna po uruchomieniu programu.

11. Poproś nauczyciela o sprawdzenie.

Treść zadania 3

Wykonaj:

- 1) przepisz temat
- 2) Przerysuj w zeszycie oraz opisz w zeszycie okno MessageDlg(, ,) → co oznaczają poszczególne parametry (są trzy).
- 3) zmienne z trzema literami (będę sprawdzał)
- 4) wykonaj zadanie
- 5) Nazwa Unitu i projektu z trzema literami nazwiska z3_kow_u oraz z3_kow_p
- 6) Nazwa okienka (formularza) *trojkat_nazwisko_ucznia*
- 7) wpisz do zeszytu listing części obliczającej zadanie

Temat: Wybór rodzaju obliczeń przy użyciu przycisku Radiobutton oraz wyprowadzanie danych przy użyciu okienka dialogowego MessageDlg(, ,).

Treść zadania.

Napisz program, który będzie obliczał pole, promień okręgu wpisane oraz opisane na trójkącie po wczytaniu trzech boków trójkąta. Użyj przycisków Wprowadź dane oraz Koniec. Do wyprowadzania danych użyj okienka dialogowego. Wygląd programu (formularza) patrz poniżej. Program będzie sprawdzał wczytywane dane i będzie informował o błędach (ujemne boki, nie zamykający się trójkąt). Informacja w okienku dialogowego MessageDlg(, ,).

Potrzebne wzory:

$$p = \frac{a + b + c}{2}$$

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

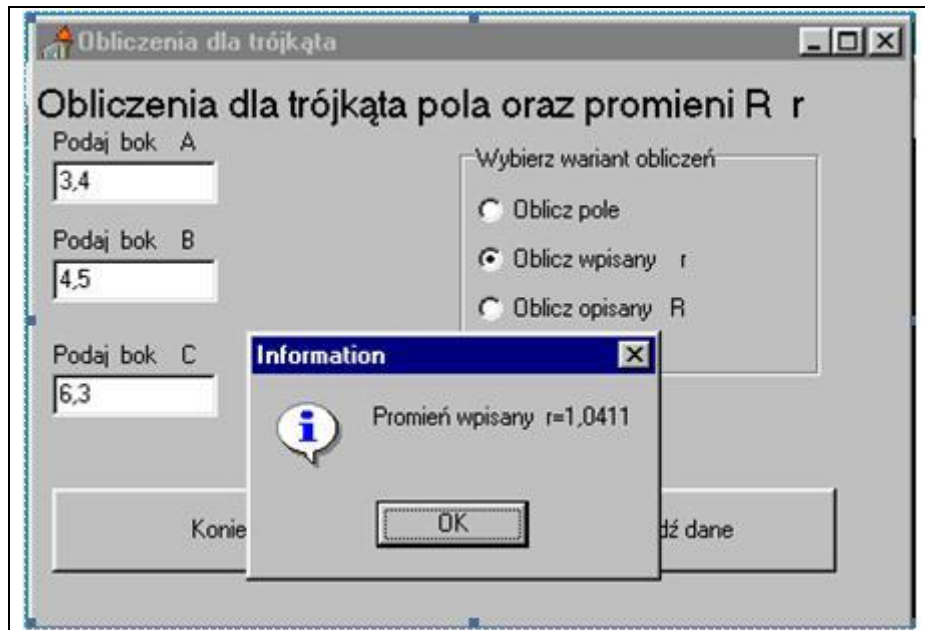
$$r = \frac{S}{p}$$

$$R = \frac{abc}{4S}$$

S - pole trójkąta
a,b,c - boki trójkąta
r - promień okręgu wpisanego
R - promień okręgu opisanego

Uwagi:

- Do zamiany zmiennych tekstowych na zmienne typu Real użyj
a := StrToCurr(Edit1.text);
- Do zamiany zmiennych Real na tekstowe użyj
s1 := CurrToStr(s);
- Do wyprowadzania danych użyj
MessageDlg('Pole S= '+s1, mtInformation, [mbOK], 0);
- Nadaj wartości początkowe zmiennym a,b,c
- podczas wpisywania danych dla liczb z miejscami po przecinku używaj przecinka



Zadanie 2 indywidualne --> zapytaj nauczyciela

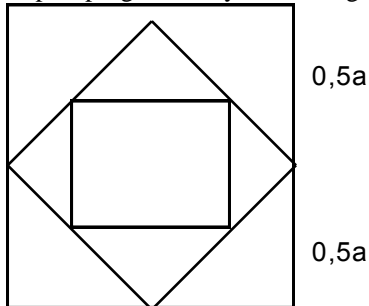
Treść przykładu 3

Wykonaj:

- 1) przepisz temat (łącznie z rysunkiem i nazwą procedury oraz jej parametrami)
- 2) zmienne z trzema literami (będę sprawdzał)
- 3) Zapisz w zeszycie znaczenie instrukcji
 - MoveTo(X,Y :integer) → zapisz znaczenie
 - LineTo(X,Y :integer) → zapisz znaczenie
- 4) Zapisz jaki obiekt należy wstawić do formularza aby można było rysować.
- 5) Nazwa Unitu i projektu z trzema literami nazwiska p3_kow_u oraz p3_kow_p
- 6) Nazwa okienka (formularza) kwadrat_nazwisko_ucznia

Temat:

Napisz program do rysowania figury:



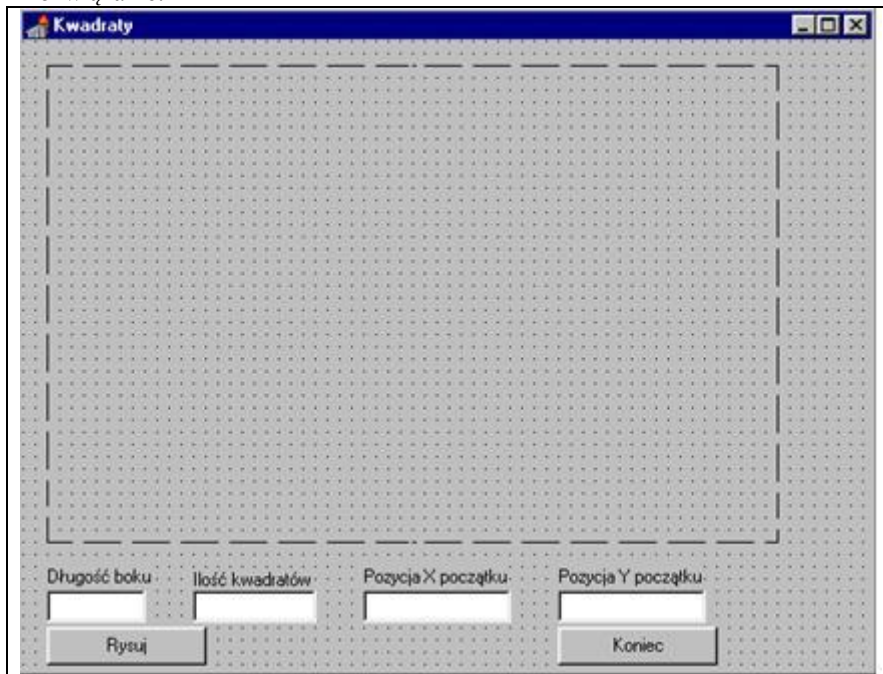
Użyj procedury KWAD(x,y,a,n: integer)

(x,y)- dolne współrzędne początku rysowania kwadratu

a- długość boku największego kwadratu;

n- ilość kwadratów

Rozwiązanie:



1. Zmień zawartość pola Caption w Object Inspector z Form1 na Kwadraty. (2)

2. Utwórz cztery pola typu Edit o następujących właściwościach: (4)

- Pole1: Name- dlug;
- Pole2: Name- ilosc;
- Pole3: Name- pozx;
- Pole4: Name- pozy;

Dla każdego z tych pól wyczyść zawartość Text. Ustawienie i wielkość po Edit → patrz poniżej

3. Nad każdym polem Edit umieść jedno pole typu Label o poniższych właściwościach: (5)

- Name-tekst1; Caption-Długość boku;
- Name-tekst2; Caption-Ilość kwadratów;
- Name-tekst3; Caption-Pozycja X początku;
- Name-tekst4; Caption-Pozycja Y początku;

4. Teraz umieść dwa przyciski sterujące: (6)

- Name-Rysuj; Caption-Rysuj;
- Name-Koniec; Caption-Koniec;

5. Kliknij teraz na zakładkę System na pasku pod menu głównym. Wybierz obiekt PaintBox. Umieść go na obszarze okna tworzonego programu tak, by zajmował jak najwięcej miejsca i nie zasłaniał innych elementów. Zmień jego Name z PaintBox1 na Obraz.

6. Teraz można już przystąpić do tworzenia właściwej części programu. Najpierw trzeba zapewnić zamykanie programu. W tym celu kliknij szybko dwa razy na przycisku Koniec. Przejdiesz do okna w którym w miejscu gdzie mruga kursor należy wpisać:

Close;

Teraz kliknij szybko dwa razy na przycisku Rysuj. Przed słowem “begin” umieść poniższą deklarację zmiennych lokalnych:

var

xp,yp,dl,ile:integer;

Pod słowem “begin” umieść następujące instrukcje:

xp:=StrToInt(pozx.Text);

yp:=StrToInt(pozy.Text);

dl:=StrToInt(Dlug.Text);

ile:=StrToInt(Ilosc.Text);

Kwad(xp,yp,dl,ile);

7. Teraz należy dokonać deklaracji procedur. Procedura kwadrat będzie korzystała z dwóch procedur:

- Kwadrat_norm: kwadrat “normalny”
- Kwadrat_obroc: kwadrat “obrócony” o 45°

Znajdź w tekście programu fragment:

private

{ Private declarations }

public

{ Public declarations }

end;

i pod nim umieść deklarację:

procedure kwad (x,y,a,n:word);

procedure kwad_norm(xa,ya,dl:word);

procedure kwad_obroc(xa,ya,dl:word);

Teraz zjedź na sam dół programu i przed słowem “end.” umieść kilka pustych linii w które wstawisz treść procedur. A więc najpierw główna procedura Kwad:

procedure kwad(x,y,a,n:word);

var i:word;

begin

for i:=1 to n do

begin

if (i mod 2)=1 then kwad_norm(x,y,a);

if (i mod 2)=0 then

begin

kwad_obroc(x,y,a);

a:=round(a/2);

x:=x+round(a/2);

y:=y+round(a/2);

end;

end;

end;

8. Do rysowania służy komponent TCanvas. W programie będziesz musiał użyć dwóch jego metod:

- MoveTo(X,Y :integer) – przemieszcza kursor graficzny na pozycję (x,y);
- LineTo(X,Y :integer) – rysuje linię od aktualnej pozycji kursora graficznego do punktu (x,y)

Procedury będą więc wyglądały następująco:

```
procedure kwad_norm(xa,ya,dl:word);
begin
  ya:=form1.obraz.height-ya;
  form1.obraz.canvas.MoveTo(xa,ya);
  form1.obraz.canvas.LineTo(xa+dl,ya);
  form1.obraz.canvas.LineTo(xa+dl,ya-dl);
  form1.obraz.canvas.LineTo(xa,ya-dl);
  form1.obraz.canvas.LineTo(xa,ya);
end;
```

```
procedure kwad_obroc(xa,ya,dl:word);
begin
  ya:=form1.obraz.height-ya;
  dl:=round(dl/2);
  xa:=xa+dl;
  form1.obraz.canvas.MoveTo(xa,ya);
  form1.obraz.canvas.LineTo(xa+dl,ya-dl);
  form1.obraz.canvas.LineTo(xa,ya-2*dl);
  form1.obraz.canvas.LineTo(xa-dl,ya-dl);
  form1.obraz.canvas.LineTo(xa,ya);
end;
```

9. Teraz nagraj i uruchom projekt (program).

10. Jeżeli wpiszesz kilka razy różne wartości, to zauważysz, że kolejne rysunki rysowane są na starych.

Należałoby więc wyczyścić pole Obraz przed rysowaniem kwadratów od nowa. W tym celu wejdź do procedury

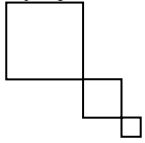
RysujClick. Zmodyfikuj ją, aby wyglądała następująco:

```
procedure TForm1.RysujClick(Sender: TObject);
var xp,yp,dl,ile:integer;
wspol:TRect;
wsp:TPoint;
begin
  xp:=StrToInt(pozx.Text);
  yp:=StrToInt(pozy.Text);
  dl:=StrToInt(Dlug.Text);
  ile:=StrToInt(Ilosc.Text);
  wsp.x:=0;
  wsp.y:=0;
  wspol.TopLeft:=wsp;
  wspol.x:=Obraz.Width;
  wspol.y:=Obraz.Height;
  wspol.BottomRight:=wsp;
  obraz.Canvas.FillRect(wspol);
  Kwad(xp,yp,dl,ile);
end;
```

Zadanie 3 indywidualne --> zapytaj nauczyciela

Treść zadania 4 (na ocenę bdb→nieobowiązkowe)

Narysuj określoną ilość kwadratów ułożonych jak poniżej



Dane do programu (wczytywane w okienkach):

- x,y → początek rysowania pierwszego kwadratu
- a → bok pierwszego kwadratu
- n → ilość kwadratów
- p → o ile procent mniejszy jest kwadrat następny

Treść przykładu 4

Temat

Wykonaj następujący formularz w/g poleceń

1. Utwórz nowy Projekt (1)
2. Zmień nazwę okna Formularza (projektu) nadaj nazwę *projekt2 numer_z_dziennika* (2)
3. Zachowaj na Twojej dyskiecie Projekt po nazwą *numer_z_dziennika_p2* (projekt oraz unit) (3)
4. Wstaw w formularzu dwa okrągłe przyciski wyboru (RadioButton) (10)

Pierwszy przycisk:

- Name autom
- Caption AUTOMATYCZNE

drugi przycisk:

- Name reczn
- Caption RĘCZNE

5. Wstaw w formularzu dwie listy przewijalne (ScrollBar) (11)

pierwsza lista przewijalna

- Name wiersze
- Min 0
- Max 10

drugie lista przewijalna

- Name kolumny
- Min 0
- Max 10

6. Wstaw trzy pole wprowadzania danych do Projektu nadaj nazwy zmiennym jak w tabelce poniżej poprzez odpowiedni zapis nazwy w Name oraz **brak tekstu** w okienku do wczytywania danych podczas uruchamiania programu (czyli nic nie powinno być w komórce Text) (4)

	nazwa Name	Znaczenie w programie
zmienna 1	Wart	służy do wprowadzania danych do tablicy
zmienna 2	w_r	numer wiersza przy sterowaniu ręcznym
zmienna 3	Kol_r	numer kolumny przy sterowaniu ręcznym

7. Wstaw następujące pola wyświetlające tekstu takie jak w tabeli (5)

nazwa Name	własność Caption	Opis
etykieta1	sterowanie automatyczne	napis w formularzu
etykieta2	Ręczne	napis w formularzu
etykieta3	Wiersze	napis w formularzu
etykieta4	Kolumna	napis w formularzu
etykieta5	liczba wierszy	napis w formularzu
etykieta6	liczba kolumna	napis w formularzu
etykieta7	podaj element	napis w formularzu
w_aut	puste pole	numer wiersza przy sterowaniu automatycznym, wpisz 1 cm za napisem wiersz na formularzu
kol_aut	puste pole	numer kolumny przy sterowaniu automatycznym, wpisz 1 cm za napisem kolumn na formularzu
Wier	0	wartość zero, numer wiersza określonego przy pomocy listwy przewijania, wpisz pod suwakiem i 1 cm za napisem liczba wierszy
Kol	0	wartość zero, numer kolumny określonej przy pomocy listwy przewijania, wpisz pod suwakiem i 1 cm za napisem liczba kolumn

8. Wprowadź na dole formularza siatkę. Do tej siatki wprowadzane będą elementy tablicy. (12)

Ustaw następujące własności siatki

Własność	Wartość
ColCount	10
RowCount	10
DefaultColWidth	60
DefaultRowHeight	21

Nadaj nazwę siatce obraz (właściwość Name)

9. Wstaw przycisk **Wprowadź** i **Koniec** (6)

pierwsze przycisk z napisem Wprowadź czyli:

- Name przycisk1
- Caption Wprowadź
- Font jak poniżej
- krój pisma Arial
- wielkość 10

drugie przycisk z napisem Koniec czyli:

- Name przycisk2
- Caption Koniec
- Font jak poniżej
- krój pisma Arial
- wielkość 10

10. Nadawanie wartości początkowych (7)

Tak aby otrzymać metodę.

```

procedure TForm1.FormCreate(Sender: TObject);
begin
  w_aut.Caption:=IntToStr(1);
  kol_aut.Caption:=IntToStr(1);
  m:=3;
  n:=3;
  obraz.RowCount:=3;
  obraz.ColCount:=3;
end;
```

w_aut.Caption	wartość początkowa numeru wiersza przy sterowaniu automatycznym
kol_aut.Caption	wartość początkowa numeru kolumny przy sterowaniu automatycznym
m	ustalenie wierszy rozmiarów tablicy
n	ustalenie kolumn rozmiarów tablicy
obraz.RowCount	ustalenie wierszy rozmiarów siatki (pierwotnie 10)
obraz.ColCount	ustalenie kolumny rozmiarów siatki (pierwotnie 10)

11. Zmiana rozmiarów tablicy

Należy tak zaprogramować aby przesuwanie suwakami list przewijania powodowało zmianę rozmiarów tablicy.

W tym celu:

- kliknąć listę przewijania
- w zakładce Events w oknie Object Inspector
- dwa razy szybko kliknąć na pustą komórkę w wierszu OnChange

i uzupełnij procedurę

```
procedure TForm1.wierszeChange(Sender: TObject);
begin
  obraz.RowCount:=wiersze.Position;
  obraz.ColCount:=kolumny.Position;
  m:=wiersze.Position;
  n:=kolumny.Position;
  wier.Caption:=IntToStr(m);
  kol.Caption:=IntToStr(n);
end;
```

Podobnie zrób dla drugiego suwaka.

12. Wprowadź akcje przypisaną do przycisku Wprowadź (6)

Kliknij dwa razy szybko przycisk Wprowadź i wpisz brakujące linie procedury tak aby otrzymać

```
procedure TForm1.przycisk1Click(Sender: TObject);
var
  x,y,i,j:integer;
begin
  if autom.checked then {tryb automatyczny}
  begin
    x := StrToInt(w_aut.caption);
    y := StrToInt(kol_aut.caption);
    tab[x,y] := StrToInt(wart.text);
    if y < n then
      kol_aut.caption := IntToStr(y + 1)
    else
      if x < m then
        begin
          kol_aut.caption := IntToStr(1);
          w_aut.caption := IntToStr(x + 1)
        end;
      end;
  end;
  if reczn.checked then {tryb ręczny}
  begin
    x := StrToInt(w_r.text);
    y := StrToInt(kol_r.text);
    if (x <= m) and (y <= n) then
      begin
        tab[x,y] := StrToInt(wart.text);
        w_aut.caption := w_r.text;
        kol_aut.caption := kol_r.text;
      end;
    end;
  end;
```

```

for i := 1 to m do
for j := 1 to n do
    obraz.Cells[j-1,i-1] := IntToStr(tab[i,j]);
end;

uwagi:
Funkcja StrToInt() zamienia tekst na liczbę.
Funkcja IntToStr() zamienia liczbę na tekst
if autom.checked then      sprawdzenie czy został wybrany przycisk automatycznie
if recz.checked then      sprawdzenie czy został wybrany przycisk ręcznie
tab[x,y] := StrToInt(wart.text);    wpisanie do tablicy wartości z okienka Wprowadź
obraz.Cells[j-1,i-1] := IntToStr(tab[i,j]);    wyświetlanie zawartości tablicy w kolejnych komórkach siatki

```

13.Wprowadź akcje przypisaną do przycisku Koniec (6)

Kliknij dwa razy szybko przycisk Koniec i wpisz brakujące linie procedury tak aby otrzymać

```

procedure TForm1.przycisk2Click(Sender: TObject);
begin
    Close
end;

```

Close zamyka program

16.Zadeklaruj zmienne w następujący sposób

```

var
    Form1: TForm1;
    m,n:integer;
    tab: array[1..10,1..10] of integer;

```

17.Dokonaj wykonania programu (8)

18.Dokonaj kompilacji na dysk programu (9)

Istnieją dwa sposoby wczytywania danych do tablicy (macierzy):

- a)AUTOMATYCZNE
- B)RĘCZNE

Sposób zaliczenia:

Ustaw macierz tablicę o wymiarze 3 na 3. Wstaw do niej dziewięć najmniejszych liczb pierwszych..

Poproś nauczyciela w celu zaliczenia

Opis sposobu wczytywania danych do tablicy (macierzy):

- 1)wybierz sposób AUTOMATYCZNY lub RĘCZNY.
- 2)gdy wybrałeś AUTOMATYCZNE teraz ustaw (suwakami)ilość wierszy i kolumn, w okienku Podaj element wpisz liczbę i wciśnij przycisk Wprowadź, następnie w okienku Podaj element wpisz liczbę i wciśnij przycisk Wprowadź elementy tablicy będą wpisywane wierszami
- 3)gdy wybrałeś RĘCZNE teraz ustaw (suwakami)ilość wierszy i kolumn, w okienku Wiersz wprowadź, w którym wierszu chcesz wprowadzać dane, w okienku Kolumna wprowadź, w której kolumnie chcesz wprowadzać dane, teraz w okienku Podaj element wpisz liczbę i wciśnij przycisk Wprowadź a zostanie wpisany w miejsce macierzy(tablicy) tam gdzie został zapisany w okienku Wiersz i Kolumna.

Treść zadania 5

Zmodyfikuj program poprzedni tak aby obliczyć i wyprowadzić :

- sumę przekątnej macierzy
- wyraz maksymalny oraz minimalny macierzy

Nagraj ten program w folderze ZADANIE5 pod inną nazwą z5_trzy_litery_nazwiska_p np. z5_kow_p oraz unit z5_trzy_litery_nazwiska_u np. z5_kow_u

Wskazówka:

Do zadania poprzedniego wstaw przyciski:

Sumę przekątnej macierzy

Wyraz max, min macierzy

Po ich naciśnięciu będą obliczenia. Pamiętaj, że wczytane elementy macierzy(tablicy) są w tablicy tab[,]

Obliczenia (kod programu w pascalu) w

Treść przykłady 5

Wykonaj program z wykorzystujący :

- przyciśnięcie lewego klawisza myszki dla losowej zmiany koloru tła formularza.
- ruch myszki wewnątrz formularza dla określenia miejsca położenia kursora
- przyciśnięcie dowolnego przycisku myszy i wyprowadzenie komunikatu, jaki klawisz został wciśnięty

Program powinien mieć następujący wygląd

Rozwiązanie:

1. Utwórz nowy Projekt (1)
2. Zmień nazwę okna Formularza (projektu) nadaj nazwę *Zadanie4* (2)
3. Zachowaj na Twojej dyskiecie Projekt po nazwą *numer_z_dziennika_p4* (projekt oraz unit) (3)
4. Wstaw przycisk **Koniec** (6)

czyli:

- Name przycisk1
- Caption Koniec
- Font jak poniżej

krój pisma	Arial
wielkość	20
kolor	ciemnozielony

5. Wstaw pole tekstowe **wspX** (5)

czyli:

- Name wspX
- Caption wspX

6. Wstaw pole tekstowe **wspY** (5)

czyli:

- Name wspY
- Caption wspY

7. Wprowadź akcje przypisaną do przycisku Koniec (6)

Kliknij dwa razy szybko przycisk Koniec i wpisz brakujące linie procedury tak aby otrzymać

```
procedure TForm1.przycisk1Click(Sender: TObject);
begin
    Close
end;
```

8. Wykonaj akcje dla następujących zdarzeń (Events):

będziesz projektował akcje dla formularza czyli w okienku musi być `Form1:TForm1`

- przyciśnięcie dowolnego przycisku myszy i wyprowadzenie komunikatu, jaki klawisz został wciśnięty

W tym celu:

- ♦ wybierz zakładkę Events w okienku Object Inspector
- ♦ w zakładce Events wybierz zdarzenie OnKeyPress
- ♦ dwa razy szybko kliknij po prawej stronie napisu OnKeyPress, uzyskasz okno z częścią programu i wpisz aby uzyskać

```
procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
begin
    MessageDlg(Key + ' naciśnięto ',mtInformation,[mbOK],0)
end;
```

- przyciśnięcie lewego klawisza myszki dla losowej zmiany koloru tła formularza.

W tym celu:

- ◆ wybierz zakładkę Events w okienku Object Inspector
- ◆ w zakładce Events wybierz zdarzenie OnClick
- ◆ dwa razy szybko kliknij po prawej stronie napisu OnClick, uzyskasz okno z częścią programu i wpisz aby uzyskać

```
procedure TForm1.FormClick(Sender: TObject);
begin
  Randomize;
  Color:=Random(65535);
end;
```

- ruch myszki wewnątrz formularza dla określenia miejsca położenia kursora

W tym celu:

- ◆ wybierz zakładkę Events w okienku Object Inspector
- ◆ w zakładce Events wybierz zdarzenie OnMouseMove
- ◆ dwa razy szybko kliknij po prawej stronie napisu OnMouseMove, uzyskasz okno z częścią programu i wpisz aby uzyskać

```
procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  wspX.Caption:=IntToStr(X);
  wspY.Caption:=IntToStr(Y);
end;
```

Uwagi:

- ⇒ wyświetlanie współrzędnych kursora uzyskujemy poprzez wykorzystanie pól tekstowych o nazwach wspX oraz wspY czyli
- ```
wspX.Caption:=IntToStr(X);
wspY.Caption:=IntToStr(Y);
```

9.Dokonaj wykonania programu **(8)**

10.Dokonaj kompilacji na dysk programu **(9)**

## Treść przykładu 6

### Temat:

Napisz program do tablicowania funkcji :

$$f(x) = \begin{cases} 1 + x + x^2 / 2 & \text{dla } |x| < 1 \\ \sin(1 + x + x^2 / 2) & \text{dla } |x| \geq 1 \end{cases}$$

Z wczytaniem krokiem tablicowania, końcem oraz początkiem przedziału tablicowania.

Rozwiązanie:

Program tablicuje funkcję:

$$f(x) = \begin{cases} 1 + x + x^2 / 2 & \text{dla } |x| < 1 \\ \sin(1 + x + x^2 / 2) & \text{dla } |x| \geq 1 \end{cases}$$

Początek przedziału tablicowania

Koniec przedziału tablicowania

Krok tablicowania

**Początek przedziału musi być liczbą rzeczywistą !!!**

**Koniec przedziału musi być liczbą rzeczywistą większą od początku przedziału !!!**

**Krok musi być liczbą rzeczywistą większą od 0 oraz mniejszą niż długość przedziału !!!**



Tablicowanie funkcji

Program tablicuje funkcję:

$$f(x) = \begin{cases} 1+x+(x^2)/2 & \text{dla } |x| < 1 \\ \sin(1+x+(x^2)/2) & \text{dla } |x| \geq 1 \end{cases}$$

Początek przedziału tablicowania

Koniec przedziału tablicowania

Krok tablicowania

**Początek przedziału musi być liczbą rzeczywistą !!!**

**Liczba elementów jest większa niż 32766 !!!**

**Należy zmniejszyć przedział lub zwiększyć wartość kroku tablicowania !!!**

**Krok musi być liczbą rzeczywistą większą od 0 oraz mniejszą niż długość przedziału !!!**

Tablicuj Koniec

Tablicowanie funkcji

Program tablicuje funkcję:

$$f(x) = \begin{cases} 1+x+(x^2)/2 & \text{dla } |x| < 1 \\ \sin(1+x+(x^2)/2) & \text{dla } |x| \geq 1 \end{cases}$$

Początek przedziału tablicowania

Koniec przedziału tablicowania

Krok tablicowania

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

Tablicuj Koniec



1. Założenie nowej aplikacji poprzez wybranie opcji FILE, a następnie NEW APPLICATION
2. Zmianienie nazwy Form1
  - Caption: Tablicowanie funkcji
3. Wstawienie tekstów w polach LABEL według formularza jak powyżej:

Label1

- Caption: Program tablicuje funkcję:
- Font name: Arial
- Font size: 20

Label2

- Caption:  $f(x)=$

Label3

- Caption: {
- Font name: Arial
- Font size: 40

Label4

- Caption:  $1+x+(x/2)$  dla  $|x|<1$  (| uzyskujemy poprzez wciśnięcie Shift+\)

Label5

- Caption:  $\sin(1+x+(x/2))$  dla  $|x|\geq 1$

Label6

- Caption: 2 (należy wstawić do wzoru jako potęgę)

Label7

- Caption: 2 (należy wstawić do wzoru jako potęgę)

Label8

- Caption: Początek przedziału tablicowania

Label9:

- Caption: Koniec przedziału tablicowania

Label10

- Caption: Krok tablicowania

Label11

- Caption: Początek przedziału musi być liczbą rzeczywistą !!!
- Font name: Time New Roman Ce
- Font size: 16
- Font color: clRed
- Visible: False

Label12

- Caption: Koniec przedziału musi być liczbą rzeczywistą większą od początku przedziału
- Font name: Time New Roman Ce
- Font size: 16
- Font color: clLime
- Alignment: taCenter
- Word wrap: True (umożliwia łamanie tekstu)
- Visible: False

Label13

- Caption: Krok musi być liczbą rzeczywistą większą od 0 oraz mniejszą niż długość przedziału
- Font name: Time New Roman Ce
- Font size: 16
- Font color: clFuchsia
- Alignment: taCenter
- Word wrap: True
- Visible: False

4. Wstawienie obiektów Edit1, Edit2 i Edit3 oraz wyczyszczenie zawartości pola TEXT.
5. Wstawienie przycisku Button1
  - Caption: Tablicuj
6. Wstawienie przycisku Button2

- Caption: Koniec
7. Wstawienie pola Label14 na istniejące pola Label
    - Caption: Liczba elementów jest większa niż 32766 !!! Należy zmniejszyć przedział lub zwiększyć wartość kroku tablicowania !!!
    - Font name: Time New Roman
    - Font size: 16
    - Word wrap: True
    - Visible: False
  8. Wstawienie obiektu StringGrid1 zakrywającego obiekty Label według rysunku
    - Height: 177
    - Width: 418
    - Visible: False
  9. Wstawienie obiektu ScrollBar1 obok obiektu StringGrid1 jak na rysunku
    - Kind: sbVertical
    - Visible: False
  10. Stworzenie procedury dla przycisku Button2 poprzez podwójne kliknięcie lewym przyciskiem myszy na obiekcie. Procedura powinna mieć postać:
 

```
procedure TForm1.Button2Click(Sender: TObject);
begin
 close; {ta instrukcja po wywołaniu zamknie program}
end;
```
  11. Stworzenie procedury uruchamianej automatycznie po rozpoczęciu aplikacji. Należy podwójnie kliknąć lewym przyciskiem myszy w pustym obszarze okna. Procedura powinna być postaci:
 

```
procedure TForm1.FormCreate(Sender: TObject);
begin
 stringgrid1.cells[0,0]:='Lp.'; {wyświetlenie napisów w pierwszym wierszu tabeli}
 stringgrid1.cells[1,0]:='X';
 stringgrid1.cells[2,0]:='F(x)';
end;
```
  12. Stworzenie procedury dla przycisku Button1 poprzez podwójne kliknięcie lewym przyciskiem myszy na obiekcie. Procedura powinna być postaci:
 

```
procedure TForm1.Button1Click(Sender: TObject);
begin
 label14.visible:=false; {wymazanie ostrzeżenia o zbyt dużej ilości elementów}
 scrollbar1.max:=1; {ustawienie maksymalnej ilości elementów suwaka}
 val(edit1.text,min,status); {sprawdzenie, czy początek przedziału jest liczbą}
 if status<>0 then label11.visible:=true {jeżeli nie to wyświetlenie ostrzeżenia}
 else label11.visible:=false; {jeżeli tak to zmazanie wyświetlenia}
 val(edit2.text,max,status); {sprawdzenie, czy koniec przedziału jest liczbą}
 if (status<>0)or(max<=min) then label12.visible:=true {jeżeli nie lub mniejsze od początku}
 else label12.visible:=false; {to wyświetlenie ostrzeżenia, jeśli tak to zmazanie}
 val(edit3.text,krok,status); {sprawdzenie, czy krok jest liczbą z przedziału}
 if (status<>0)or(not label11.visible)and(not label12.visible)and((krok>max-min)or(krok<=0)) then
 label13.visible:=true {jeśli nie to wyświetl ostrzeżenie}
 else label13.visible:=false; {jeśli tak to zmazanie ostrzeżenia}
 if (krok<>0)and((max-min)/krok+2<=32767) then {jeśli zgadza się krok}
 begin
 if (not label11.visible)and(not label12.visible)and(not label13.visible) {jeśli nie ma błędu}
 then begin {rozpocznij tablicowanie funkcji}
 scrollbar1.visible:=true; {pokaż suwak}
 stringgrid1.visible:=true; {pokaż tablicę}
 stringgrid1.rowcount:=round((max-min)/krok)+2; {ustaw ilość wierszy}
 if (max-min)/krok-4>1 then scrollbar1.max:=round((max-min)/krok-4); {ustaw ilość elementów}
 lp:=1; {numer elementu}
 while lp<=6 do {wyświetlenie 6 kolejnych elementów}
 begin
 stringgrid1.cells[0,lp]:=inttostr(lp); {wyświetlenie numeru elementu}
 stringgrid1.cells[1,lp]:=currtostr(min+krok*(lp-1)); {wyświetlenie wartości X}
```

- ```

        stringgrid1.cells[2,lp]:=currtostr(wynik(min+krok*(lp-1)));      {wyświetlenie wyniku funkcji}
        lp:=lp+1;                                                       {zwiększenie numeru elementu}
    end;
end
else begin
    scrollbar1.visible:=false;                                          {jeżeli coś nie tak zmaż suwak}
    stringgrid1.visible:=false;                                         {oraz tabelę}
end;
end;
if (krok<>0)and((max-min)/krok+2>32767) then                            {jeżeli ilość kroków nieprawidłowa}
begin
    label14.visible:=true;                                              {wyświetl ostrzeżenie}
    stringgrid1.visible:=false;                                         {zmaż tabelę}
    scrollbar1.visible:=false;                                           {zmaż suwak}
end;
end;

```
13. Stworzenie procedury dla suwaka ScrollBar1 poprzez podwójne kliknięcie na obiekcie lewym przyciskiem myszy. Procedura powinna być postaci:
- ```

procedure TForm1.ScrollBar1Change(Sender: TObject);
begin
 if scrollbar1.position=1 then {jeżeli suwak na pozycji 1}
 begin
 lp:=1; {wyświetl pierwszych 6 elementów}
 while lp<=6 do
 begin
 stringgrid1.cells[0,lp]:=inttostr(lp);
 stringgrid1.cells[1,lp]:=currtostr(min+krok*(lp-1));
 stringgrid1.cells[2,lp]:=currtostr(wynik(min+krok*(lp-1)));
 lp:=lp+1;
 end;
 end;
 if scrollbar1.position>1 then {jeżeli suwak na inne pozycji}
 begin
 lp:=1; {ustaw numer elementu na 1}
 while lp<=6 do {wyświetl 6 kolejnych elementów}
 begin
 stringgrid1.cells[0,lp]:=inttostr(lp-1+scrollbar1.position); {numer elementu z suwaka}
 stringgrid1.cells[1,lp]:=currtostr(min+krok*(lp-2+scrollbar1.position)); {wyświetlenie wartości}
 stringgrid1.cells[2,lp]:=currtostr(wynik(min+krok*(lp-2+scrollbar1.position)));
 lp:=lp+1;
 end;
 end;
end;

```
14. Dopisanie zmiennych na początku źródła programu. Po uzupełnieniu zmienne powinny mieć następującą postać:
- ```

var
    Form1: TForm1;
    min,max,krok,i:real;
    lp,status:integer;
    t:string;

```
15. Dopisanie na początku programu procedury liczącej wartość funkcji $f(x)$. Procedura powinna mieć następującą postać:
- ```

function wynik(x:real):real;
begin
 if abs(x)<1 then wynik:=1+x+(x*x)/2;
 if abs(x)>=1 then wynik:=sin(1+x+(x*x)/2);
end;

```
16. Zapisanie programu na dysku pod nazwą FUNKCJA.
17. Uruchomienie programu przez wciśnięcie klawisza F9.

**Zadanie 6**

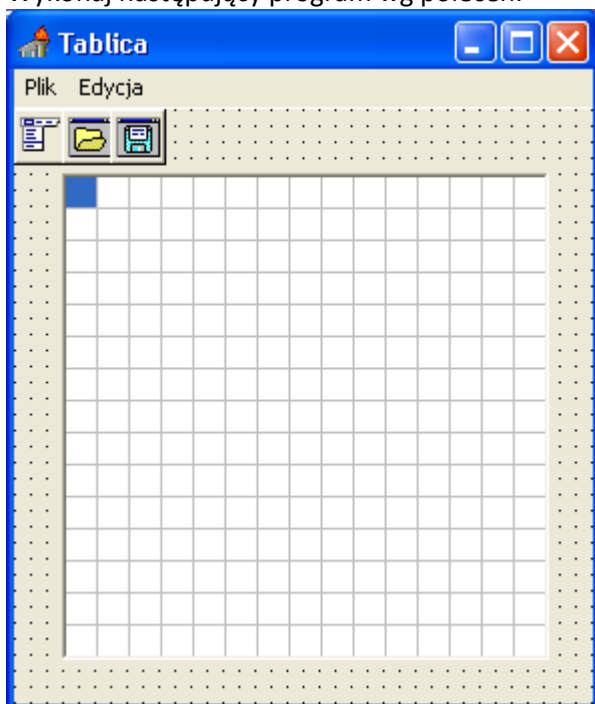
Napisz program znajdujący wszystkie trójki pitagorejskie dla **zadanego przedziału**. (konieczne będzie okienko do wczytywania tej danej) Trójka pitagorejska to ciąg trzech liczb naturalnych spełniających twierdzenie Pitagorasa np. 3, 4, 5 lub 5, 12, 13.


Użyj obiektu StringGrid1( patrz przykład poprzedni ) do wypisywania kolejnych trójek.

**Zadanie 4 indywidualne --> zapytaj nauczyciela**

Treść przykładu


Wykonaj następujący program wg poleceń.



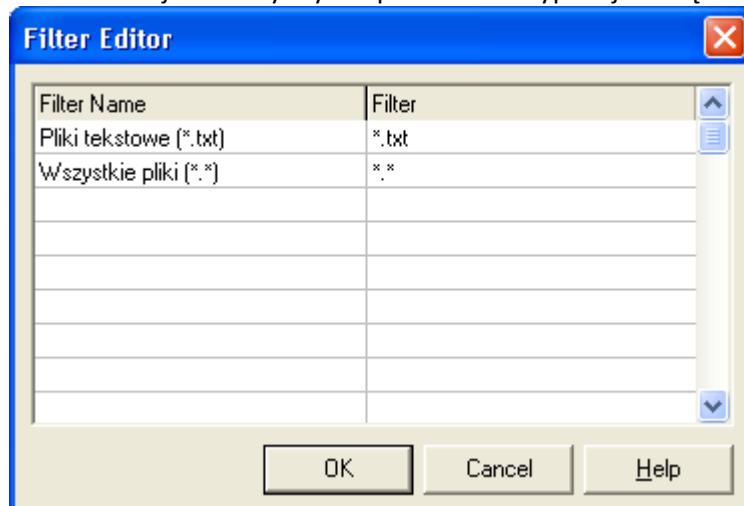
1. Utwórz nowy projekt **(1)**
2. Zmień nazwę okna Formularza (projektu) nadaj nazwę *Tablica\_nazwisko\_ucznia* **(2)**
3. Zachowaj na twoim nośniku projekt pod nazwą *Tab\_numer\_z\_dziennika* oraz unit pod nazwą *T\_u\_numer\_z\_dziennika* **(3)**
4. Wstaw w formularzu menu 
  - a) Kliknij ikonę MainMenu po czym kliknij w dowolne miejsce na formularzu.
  - b) Kliknij dwa razy szybko na wstawionym obiekcie. Otworzy się okno edytora menu o nazwie Form1.MainMenu1
  - c) W oknie Object Inspector uzupełnij właściwość Caption wpisując **Plik** i wciśnij enter
  - d) W oknie edytora menu kliknij Plik po czym kliknij białe pole pod przyciskiem Plik i uzupełnij właściwość Caption wpisując **Nowy**. Wciśnij enter.
  - e) W oknie edytora menu kliknij białe pole pod przyciskiem Nowy i uzupełnij właściwość Caption wpisując **Otwórz**. Wciśnij enter.
  - f) Uzupełnij resztę menu wg schematu:
 


|               |                                       |
|---------------|---------------------------------------|
| Plik          | Edycja                                |
| — Nowy        | — Zliczanie wszystkich znaków         |
| — Otwórz      | — Zliczanie znaków w podanej kolumnie |
| — Zapisz      | — Zliczanie znaków w podanym wierszu  |
| — Zapisz jako |                                       |
| — Zamknij     |                                       |
  - g) Zamknij okno edytora menu.
5. Wstaw do formularza siatkę (StringGrid) i ustaw następujące właściwości:
  - ColCount            15
  - RowCount           15
  - FixedCols           0
  - FixedRows           0
  - DefaultColWidth    15

- DefaultRowHeight 15
- ScrollBars ssNone
- Height 243
- Width 243
- Name StringGrid1

6. Wstaw do formularza obiekt OpenFileDialog  (znajduje się w zakładce Dialogs) i ustaw następujące właściwości:

- Name OpenFileDialog1
- Filter – kliknij dwa razy szybko pole Filter i wypełnij tabelę:



7. Wstaw do formularza obiekt SaveDialog  (znajduje się w zakładce Dialogs) i ustaw następujące właściwości:

- Name SaveDialog1
- Filter – kliknij dwa razy szybko pole Filter i wypełnij tabelę identycznie jak w OpenFileDialog

8. Nadawanie wartości początkowych **(7)**

Tak aby otrzymać procedurę:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
 Zapisano:=false;
end;
```

9. Wstawianie znaków O w tablicy.

Należy tak zaprogramować aby po kliknięciu w dowolną pustą komórkę tablicy (obiektu StringGrid1) pojawił się znak O a po kliknięciu w dowolną komórkę ze znakiem O znak z tej komórki ma zniknąć.

W tym celu:

- Kliknij obiekt StringGrid1
- Przejdź na zakładkę events w oknie Object Inspector
- Kliknij dwa razy szybko procedurę OnSelectCell i uzupełnij tą procedurę:

```
procedure TForm1.StringGrid1SelectCell(Sender: TObject; Col, Row: Longint;
 var CanSelect: Boolean);
begin
 if Stringgrid1.cells[col,row]='O' then
 Stringgrid1.cells[col,row]:=""
 else
```

```
Stringgrid1.cells[col,row]:='O';
end;
```

10. Wprowadź akcję przypisaną do przycisku Nowy w menu Plik.  
Na formularzu kliknij Plik i Nowy. Uzupełnij Procedurę:

```
procedure TForm1.Nowy1Click(Sender: TObject);
begin
 for i:=0 to 14 do
 for j:=0 to 14 do
 stringgrid1.cells[i,j]:='';
 zapisano:=false;
 Form1.caption:='Tablica';
end;
```

11. Wprowadź akcję przypisaną do przycisku Otwórz w menu Plik.  
Na formularzu kliknij Plik i Otwórz. Uzupełnij procedurę:

```
procedure TForm1.Otwrz1Click(Sender: TObject);
var
 t:string;
begin
 if OpenFileDialog1.execute=true then
 begin
 NazwaPliku:=OpenDialog1.FileName;
 Form1.caption:='Tablica - '+GetName(NazwaPliku);
 AssignFile(Plik,OpenDialog1.FileName);
 Reset(Plik);
 for i:=0 to 14 do
 begin
 ReadLn(Plik,t);
 if length(t)=15 then
 begin
 for j:=0 to 14 do
 begin
 if t[j+1]='1' then
 stringgrid1.cells[j,i]:='O';
 if t[j+1]='0' then
 stringgrid1.cells[j,i]:='';
 end;
 end;
 end;
 else
 MessageDlg('Błąd w linii '+inttostr(i+1)+' pliku wejściowego',mtinformation,[mbOK],0);
 end;
 CloseFile(Plik);
 zapisano:=true;
 end;
 end;
```

12. Wprowadź akcję przypisaną do przycisku Zapisz w menu Plik.  
Na formularzu kliknij Plik i Zapisz. Uzupełnij procedurę:

```

procedure TForm1.Zapisz1Click(Sender: TObject);
begin
if zapisano=false then
begin
 Zapiszjako1Click(Sender);
end
else
begin
 if nazwapliku<>' ' then
 begin
 AssignFile(Plik,NazwaPliku);
 Rewrite(Plik);
 for i:=0 to 14 do
 begin
 for j:=0 to 14 do
 begin
 if stringgrid1.cells[j,i]='O' then
 Write(Plik,'1')
 else
 Write(Plik,'0');
 end;
 Writeln(Plik);
 end;
 end;
 CloseFile(Plik);
 MessageDlg('Zapisano',mtInformation,[mbOK],0);
 end;
end;
end;

```

13. Wprowadź akcję przypisaną do przycisku Zapisz jako w menu Plik. Na formularzu kliknij Plik i Zapisz jako. Uzupełnij procedurę:

```

procedure TForm1.Zapiszjako1Click(Sender: TObject);
begin
if SaveDialog1.execute=true then
begin
 NazwaPliku:=SaveDialog1.filename;
 Form1.caption:='Tablica - '+GetName(NazwaPliku);
 AssignFile(Plik,SaveDialog1.FileName);
 Rewrite(Plik);
 for i:=0 to 14 do
 begin
 for j:=0 to 14 do
 begin
 if stringgrid1.cells[j,i]='O' then
 Write(Plik,'1')
 else
 Write(Plik,'0');
 end;
 Writeln(Plik);
 end;
 end;
end;

```



```

CloseFile(Plik);
Zapisano:=true;
MessageDlg('Zapisano',mtInformation,[mbOK],0);
end;
end;

```

14.. Wprowadź akcję przypisaną do przycisku Zamknij w menu Plik.  
Na formularzu kliknij Plik i Zamknij. Uzupełnij procedurę:

```

procedure TForm1.Zamknij1Click(Sender: TObject);
begin
 close;
end;

```

15. Wprowadź akcję przypisaną do przycisku Zliczanie wszystkich znaków w menu Edycja.  
Na formularzu kliknij Edycja i Zliczanie wszystkich znaków. Uzupełnij procedurę:

```

procedure TForm1.Zliczaniewszystkichznakw1Click(Sender: TObject);
begin
 suma:=0;
 for i:=0 to 14 do
 for j:=0 to 14 do
 if stringgrid1.cells[i,j]='O' then
 suma:=suma+1;
 MessageDlg('Ilość znaków: '+inttostr(suma),mtinformation,[mbOK],0);
 end;
end;

```

16. Wprowadź akcję przypisaną do przycisku Zliczanie znaków w podanej kolumnie w menu Edycja.  
Na formularzu kliknij Edycja i Zliczanie znaków w podanej kolumnie. Uzupełnij procedurę:

```

procedure TForm1.Zliczanieznakwwpodanejkolumnie1Click(Sender: TObject);
var
 y:string;
 y1:integer;
begin
 suma:=0;
 if InputQuery('Wprowadzanie','Podaj numer kolumny',Y) then
 begin
 y1:=strtoint(Y);
 if (y1>0)and(Y1<=15) then
 begin
 for i:=0 to 14 do
 begin
 if stringgrid1.cells[strtoint(Y)-1,i]='O' then
 suma:=suma+1;
 end;
 MessageDlg('Ilość znaków: '+inttostr(suma),mtinformation,[mbOK],0);
 end
 else
 MessageDlg('Numer kolumny musi być większy od zera i mniejszy od 15',mtinformation,[mbOK],0);
 end;
end;

```

end;

17. Wprowadź akcję przypisaną do przycisku Zliczanie znaków w podanym wierszu w menu Edycja. Na formularzu kliknij Edycja i Zliczanie znaków w podanym wierszu. Uzupełnij procedurę:

```
procedure TForm1.Zliczanieznakwwokrelonymwierszu1Click(Sender: TObject);
var
 y:string;
 y1:integer;
begin
 suma:=0;
 if InputQuery('Wprowadzanie','Podaj numer wiersza',Y) then
 begin
 y1:=strtoint(Y);
 if (y1>0)and(Y1<=15) then
 begin
 for i:=0 to 14 do
 begin
 if stringgrid1.cells[i,strtoint(Y)-1]='O' then
 suma:=suma+1;
 end;
 end;
 MessageDlg('Ilość znaków: '+inttostr(suma),mtinformation,[mbOK],0);
 end
 else
 MessageDlg('Numer wiersza musi być większy od zera i mniejszy od 15',mtinformation,[mbOK],0);
 end;
end;
```

18. Wpisz na początku programu (przed pierwszą procedurą) funkcję GetName:

```
function GetName(s:string):string;
var
 k:integer;
 en:boolean;
begin
 en:=true;
 for k:=Length(s) downto 1 do
 begin
 if (s[k]='\') and (en=true) then
 begin
 GetName:=Copy(s,k+1,(Length(s)-k));
 en:=false;
 end;
 end;
end;
```

19. Dopisz zmienne na początku programu. Zmienne powinny mieć następującą postać:

```
var
 Form1: TForm1;
 i,j,suma,kol:integer;
```

```
zapisano:boolean;
NazwaPliku:String;
Plik:TextFile;
```

20. Dokonaj kompilacji programu na dysk **(9)**

21. Dokonaj wykonania programu **(8)**

## Teoria → Instrukcja Delphi 2.0

### Definicja

**DELPHI** jest wizualnym środowiskiem programistycznym, w którym programy są budowane z **komponentów** czyli wizualnie przedstawienie określonych funkcji, które może wykonywać program np. RadioGroup, komponenty są ściągane z menu w którym jest ich spis. Następnie program uzupełniany jest odpowiednim fragmentem kodu w języku Pascal, który jest zależny od tego jakie zadania ma wykonywać pisany przez nas program.

### Okno Object Inspector

Object Inspector to miejsce w DELPHI w którym edytujemy właściwości danego komponentu lub okna.

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Zakładka:<br/> <b>Properties</b> → ustawianie właściwości komponentu (obiektu).<br/><br/> <b>Events</b> → ustawianie reakcji na zdarzenia, jak będzie reagował komponentu (obiektu).<br/><br/>         Nad Properties i Events znajduje się okienko w którym po rozwinięciu listy będziemy mieli wszystkie komponenty (obiekty). Po wyborze komponentu (obiektu) będziemy mogli dokonać zmiany jego właściwości oraz zdarzeń.</p> |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### 1. Properties (właściwości):

- \* **Align** to rozmieszczenie danego komponentu:.  
 Możemy ustawić tu wartości wyrównania:  
 alLeft, alRight, alTop, alBottom, alClient  
 (od lewej, od prawej, od góry, od dołu, na pełnej aplikacji).
- \* **Caption** to miejsce w którym możemy wpisywać tekst np. na pasku aplikacji, w etykietach typu Label.
- \* **Color** jak wskazuje nazwa służy do zmiany koloru danego obiektu.
- \* **Font** czyli czcionka umieszczona w programie.
- \* **Name** - nazwa zmiennej jakiej używamy w kodzie źródłowym odwołując się do nazwanego tak obiektu.
- \* **Items** - możemy tu umieścić elementy np. menu czy RadioGroup
- \* **Lines** lub **Text** - możemy w tym miejscu wstawić tekst jaki będzie widoczny w danym komponencie.
- \* **Picture** - tutaj wstawiamy obrazek.

#### 2. Events (zdarzenia) które zachodzą gdy zostanie spełniony dany warunek:

- \* **OnClick** zdarzenie zachodzi gdy klikniemy na dany obiekt.
- \* **OnDblClick** zdarzenie zachodzi gdy dwukrotnie klikniemy na dany obiekt.
- \* **OnClose** lub **OnExit** zdarzenie zachodzi gdy aplikacja jest zamykana.
- \* **OnCreate** zdarzenie zachodzi gdy aplikacja jest uruchamiana.

### Tworzenie projektu ( formularza ) (1)

- Po uruchomieniu Delphi nowy projekt jest tworzony automatycznie.
- W celu utworzenia nowego projektu z File opcję New Application

### Zmiana nazwu okna Formularza (projektu) (2)

W okienku **Object Inspector** ( Własności ) w zakładce **Properties** w komórce **Caption** wpisz nazwę

### Zachowanie Projektu oraz Unitu (3)

Pliku Unitu mają rozszerzenie PAS

Pliki Projektu mają rozszerzenie DPR

Wybierz polecenie File/Save Project As. Najpierw nadaj nazwę Unitowi potem Projektowi

| Element                     | Delphi          | C++ Builder   |
|-----------------------------|-----------------|---------------|
| Plik projektu               | *.dpr           | *.bpr         |
| Plik grupy projektowej      | *.bpg           | *.bpg         |
| Plik kodu źródłowego*.pas   |                 | *.cpp         |
| Plik nagłówkowy             | ----            | *.h lub *.hpp |
| Plik formularza             | *.dfm           | *.dfm         |
| Skompilowany plik binarny   | *.dcu lub *.obj | *.obj         |
| Skompilowany plik zasobów   | *.res lub .dcr  | *.res         |
| Zapisane ustawienia pulpitu | *.dsk           | *.dsk         |
| Ustawienia projektu         | *.dof           | ----          |
| Pliki źródłowe pakietu      | *.dpk           | *.bpk         |
| Skompilowany pakiet         | *.bpl           | *.bpl         |
| Układ pulpitu               | *.dst           | *.dst         |

#### Uwaga:

Pliki z rozszerzeniami ~ to kopie zapasowe(pliki tymczasowe)

### Umieszczanie pola do wprowadzania danych (4)

Kliknij ikonę Edit ( to taka z literkami ab ). Następnie przejdź na okno Projektu i kliknij w miejscu gdzie chcesz je umieścić. Ustawiamy własności:

- Name nazwa zmiennej w programie
- Text tekst pokazujący się w podczas uruchamiania programu w okienku do wprowadzania danych
- Font kliknij dwa razy szybko na mały plus przed napisem Font ( plusik zamieni się na mały minusik i rozwinie się dodatkowe menu ) z tego menu wybierz Name i teraz wybierz czcionkę, możesz ustawiać również inne własności poprzez : kliknij w pustej komórce pojawią się trzy kropki, kliknij je a uzyskasz możliwość zmiany czcionek

Zmiany tych własności przeprowadzamy w **Object Inspector** ( Własności ) w zakładce **Properties** w odpowiednich komórkach.

### Wstawianie pól do wyświetlania tekstów (5)

Kliknij ikonę Label ( to taka z literką A ). Następnie przejdź na okno Projektu i kliknij w miejscu gdzie chcesz je umieścić. Ustawiamy własności:

- Name nazwa zmiennej w programie
- Caption tekst pokazujący się w polu tekstowym
- Font kliknij dwa razy szybko na mały plus przed napisem Font ( plusik zamieni się na mały minusik i rozwinie się dodatkowe menu ) z tego menu wybierz Name i teraz wybierz czcionkę, możesz ustawiać również inne własności poprzez : kliknij w pustej komórce pojawią się trzy kropki, kliknij je a uzyskasz możliwość zmiany czcionek

### Wstawianie przycisków sterujących (6)

Kliknij ikonę Button ( to taka z napisem OK ). Następnie przejdź na okno Projektu i kliknij w miejscu gdzie chcesz je umieścić. Ustawiamy własności:

- Name nazwa przycisku w programie
- Caption tekst pokazujący się na przycisku
- Font kliknij dwa razy szybko na mały plus przed napisem Font ( plusik zamieni się na mały minusik i rozwinie się dodatkowe menu ) z tego menu wybierz Name i teraz wybierz czcionkę, możesz ustawiać również inne własności poprzez : kliknij w pustej komórce pojawią się trzy kropki, kliknij je a uzyskasz możliwość zmiany czcionek

- Programowanie akcji, która ma być podjęta po naciśnięciu przycisku sterującego.
- Kliknij dwa razy na przycisk ( w fazie projektownia ) a otworzy się fragment programu gdzie będziesz mógł wpisać część programu, która podejmie akcję po wciśnięciu przycisku.

### Ustawianie wartości początkowych (7)

Wartości początkowe nie mogą być wpisane taka jak w pascalu poprzez przypisanie zmiennej wartości początkowej np. s:=0; Ustawienie wartości początkowej powinno odbyć się w metodzie. Najlepszą metodą jest metoda wykonywana przy tworzeniu formularza ( projektu ), co odpowiada zdarzeniu **OnCreate**. W celu znalezienia tego zdarzenia kliknij na formularz i wybierz zakładkę **Events** w oknie **Object Inspector** i następnie dwa razy szybko kliknął myszką w wierszu OnCreate i system przeniesie nas do edycji metody o nazwie **FormCreate**. W treść procedury TForm1.FormCreate wpisujemy wartości początkowe.

### Uruchamianie programu (8)

Klawisz F9 (podczas uruchamiania programu tworzone jest EXE )

### Kompilacja na dysk (9)

Klawisze Alt+F9 ( podczas kompilacji twirzone jest EXE )

### Wstawianie okrągłych przycisków wyboru ( RadioButton ) (10)

Kliknij ikonę RadioButton ( to taka z okrągłym kółeczkiem ). Następnie przejdź na okno Projektu i klikni w miejscu gdzie chcesz je umieścić. Ustawiamy własności:

- Name nazwa przycisku w programie
  - Caption tekst pokazujący się na przycisku
  - Font kliknij w pustej komórce pojawią się trzy kropki, kliknij je a uzyskasz możliwość zmiany czcionek
  - Programowanie akcji, która ma być podjęta po naciśnięciu okrągłego przycisku wyboru
- Kliknij dwa razy na przycisk ( w fazie projektownia ) a otworzy się fragment programu gdzie będziesz mógł wpisać część programu, która podejmie akcję po wciśnięciu przycisku.

### Wstawianie listy przewijalnej ( ScrollBar ) (11)

Kliknij ikonę ScrollBar ( to taka z symbolem listy). Następnie przejdź na okno Projektu i klikni w miejscu gdzie chcesz je umieścić. Ustawiamy własności:

- Name nazwa listy w programie
- Min jest to wartość minimalna jaka odpowiada skrajnemu lewemu położeniu wskaźnika w listwie
- Max jest to wartość maksymalna jaka odpowiada skrajnemu prawemu położeniu wskaźnika w listwie
- Programowanie akcji, która ma być podjęta po zmianie położenia suwaka.

Kliknij dwa razy na przycisk ( w fazie projektownia ) a otworzy się fragment programu gdzie będziesz mógł wpisać część programu, która podejmie akcję po wciśnięciu przycisku.

### Wstawianie siatki ( StringGrid ) (12)

Kliknij zakładkę Additional ( aby otrzymać dodatkowy pasek narzedzi ) ikonę StringGrid. Następnie przejdź na okno Projektu i klikni w miejscu gdzie chcesz je umieścić. Ustawiamy własności:

- ColCount liczba kolumn
- RowCount liczba wierszy
- DefaultColWidth szerokość jednej kolumny
- DefaultRowHeight wysokość jednego wiersza.

Kliknij dwa razy na przycisk ( w fazie projektownia ) a otworzy się fragment programu gdzie będziesz mógł wpisać część programu, która podejmie akcję po wciśnięciu przycisku.

### Wyrównywanie komponentów (13)

- wybierz okienko Align poprzez opcje menu View\Alignment Palette
- zaznacz elementy formularza, które chcesz wyrównywać jako blok
- wybierz opcję okienka Align poprzez kliknięcie odpowiedniej ikony

### Grupowanie (Radiogroup) okrągłych przycisków wyboru (Radiobotton) w (14)

- wybierz ikonę grupowania przycisków z menu (Radiogroup )
- dokonaj odpowiedniego umiejscowienia, wielkości oraz nazwy RadioGroup

- wstaw wszystkie przyciski wyboru (RadioButton)
- gdy najpierw wstawiłeś przyciski konieczne będzie:
- zaznacz zdefiniowane przyciski jako blok i wytnij je czyli Ctrl+X ( konieczne jest wycięcie ponieważ w pierwszej kolejności należy wstawić RadioGroup a dopiero później RadioButton )
  - wstaw grupę RadioGroup poprzez wybór odpowiedniej ikony z menu
  - zapisz Caption dla RadioGroup
  - wstaw RadioButton przechowywane w skrytce poprzez Ctrl+V

### Wykorzystanie MessageDlg() (15)

MessageDlg(parametr1 , parametr2 , parametr3 , parametr4)

jest to procedura czteroparametrowa

|                                                                          |                                                                                                                      |
|--------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>parametr1</b>                                                         | <b>text lub wartości zmiennych ,</b>                                                                                 |
| <b>parametr2</b>                                                         | <b>sposób wyświetlania informacji o oknie , w przykładzie poniżej jest litera I w niebieskim kole ,</b>              |
| mtWarning                                                                | A message box containing a yellow exclamation point symbol.                                                          |
| mtError                                                                  | A message box containing a red stop sign.                                                                            |
| mtInformation                                                            | A message box containing a blue "i".                                                                                 |
| mtConfirmation                                                           | A message box containing a green question mark.                                                                      |
| mtCustom                                                                 | A message box containing no bitmap. The caption of the message box is the name of the application's executable file. |
| <b>parametr3</b>                                                         | <b>tekst potwierdzenia ,</b>                                                                                         |
| mbYes                                                                    | A button with a green check mark and the text 'Yes' on its face                                                      |
| mbNo                                                                     | A button with a red circle and slash mark through the circle and the text 'No' on its face                           |
| mbOK                                                                     | A button with a green check mark and the text 'OK' on its face                                                       |
| mbCancel                                                                 | A button with a red X and the text 'Cancel' on its face                                                              |
| mbHelp                                                                   | A button with a cyan question mark and the text 'Help' on its face                                                   |
| mbAbort                                                                  | A button with a red check mark and the text 'Abort' on its face                                                      |
| mbRetry                                                                  | A button with two green circular arrows and the text 'Retry' on its face                                             |
| mbIgnore                                                                 | A button with a green man walking away and the text 'Ignore' on its face                                             |
| mbAll                                                                    | A button with a green double check marks and the text 'All' on its face                                              |
| mbYesNoCancel                                                            | A set that puts the Yes, No, and Cancel buttons in the message box                                                   |
| mbOkCancel                                                               | A set that puts the OK and Cancel buttons in the message box                                                         |
| mbAbortRetryIgnore                                                       | A set that puts an Abort, Retry, and Ignore buttons in the message box                                               |
| <b>Uwaga:</b> parametr ten pisz w nawiasach [ ]. Patrz przykład poniżej. |                                                                                                                      |
| <b>parametr4</b>                                                         | <b>numer pomocy kontekstowej przy naciśnięciu F1. Gdy zero to pomoc zdefiniowana</b>                                 |
| <b>np. MessageDlg(Key + ' naciśnięto ',mtInformation, [mbOK], 0)</b>     |                                                                                                                      |

### Zmiana nazwu okna Formularza (projektu ) (16)

W celu wgrania programu należy wgrać plik z rozszerzeniem DPR, automatycznie zostanie wgrany plik unitu z rozszerzeniem PAS.

## Delphi/Canvas, czyli rysowanie

Załóżmy, że Twój program ma nie tylko wyświetlać standardowe kontrolki Windows, ale także coś rysować. Samo rysowanie w Windows jest zadaniem dość kłopotliwym, ale Delphi dostarcza Ci do niektórych komponentów obiekt o nazwie Canvas ("płótno"). Rysując na obiekcie Canvas danego komponentu w rzeczywistości rysujesz na nim samym. Nie wszystkie komponenty mają swoje płótno.

Najprostszym komponentem z właściwością Canvas jest formularz. Utwórz nowy projekt i dodaj do niego przycisk (Button1). Po wciśnięciu przycisku program ma narysować coś na formularzu.

Jak już wiesz, zareagowanie na kliknięcie przycisku odbywa się w metodzie OnClick.

Obiekt Canvas ma szereg instrukcji służących do rysowania elementów graficznych. Narysujmy koło:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
 Canvas.Ellipse(20,20,80,80);

end;
```

Uruchom program i sprawdź, jak działa.

Metoda Ellipse rysuje elipsę wewnątrz prostokąta o podanych współrzędnych. Spróbujmy narysować elipsę na całym formularzu:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
 Canvas.Ellipse(0,0,Width,Height);

end;
```

Właściwości Width i Height to szerokość i wysokość formularza. Jeśli uruchomisz program, to zauważysz, że elipsa jest nieco przycięta od dołu (a także z prawej strony). Odpowiada za to pasek tytułowy formularza: samo płótno jest mniejsze od formularza. Niestety, nie wiemy, jak wysoki jest pasek tytułowy okna (na każdym komputerze zależy to od indywidualnych ustawień użytkownika). Sam obiekt Canvas jest dowolnego rozmiaru – możesz rysować nawet poza widoczną częścią okna. Na szczęście, formularz ma także właściwości ClientWidth i ClientHeight. Odpowiadają one za wymiary widocznej i dostępnej części okna. Zmodyfikuj program używając tych dwóch właściwości:

```
Canvas.Ellipse(0,0,ClientWidth,ClientHeight);
```

Teraz cała elipsa jest widoczna na ekranie.

Spróbuj teraz zmienić rozmiary swojego okna: najpierw zmniejsz je, a potem zwiększ. Możesz również zminimalizować i przywrócić okno. Co się stało? Część elipsy została utracona. Czemu?

W pewnym momencie system Windows stwierdza, że okno musi zostać przerysowane (narysowane na nowo). Przerysowaniem wszystkich komponentów zajmuje się ich rodzic, a więc przycisk jest przerysowany automatycznie. Jednak po każdej takiej operacji niewidoczna część płótna jest bezpowrotnie tracona. Jak można odzyskać narysowaną elipsę?

Kiedy system domaga się przerysowania okna, wysyła do niego komunikat WM\_PAINT. Nie wnikajmy na razie, czym jest WM\_PAINT. Dla nas istotne jest, że ten komunikat wywołuje na formularzu zdarzenie OnPaint. Wewnątrz metody obsługującej to zdarzenie możemy przerysować nasze okno.

Wybierz formularz i znajdź jego metodę OnPaint. Dla uproszczenia, usuń przycisk Button1 i przypisaną mu metodę Button1Click. Teraz nadpiszemy metodę OnPaint:



```
procedure TForm1.FormPaint(Sender: TObject);
begin
```

```
 Canvas.Ellipse(0,0,ClientWidth,ClientHeight);
```

```
end;
```

Uruchom program, zminimalizuj i zmaksymalizuj go. Zawartość okna nie została utracona (w rzeczywistości, została utracona, ale przywrócenie okna do widoku spowodowało narysowanie jego zawartości na nowo). Teraz pobaw się zmianą rozmiaru okna... Czy coś jest nie tak?

Zmiana rozmiaru okna powoduje wywołanie komunikatu żądania przerysowania okna, ale dla oszczędności, nie całe okno jest przerysowywane, a jedynie ta część, która zostaje odsłonięta przy zwiększaniu rozmiaru. Jak obejść ten problem? Zmiana rozmiaru okna wywołuje zdarzenie OnResize formularza. Wpiszmy więc w tej metodzie:

```
procedure TForm1.FormResize(Sender: TObject);
begin
```

```
 Canvas.Ellipse(0,0,ClientWidth,ClientHeight);
```

```
end;
```

Ponieważ metody FormPaint i FormResize w tym konkretnym przypadku mają robić to samo, można skorzystać z zapisu:

```
procedure TForm1.FormResize(Sender: TObject);
begin
```

```
 FormPaint(Sender);
```

```
end;
```

Uruchom program i sprawdź, jak zachowuje się rysunek przy zmianie rozmiaru okna.

Prawie dobrze. Przydałoby się jeszcze oczyścić płótno formularza przed narysowaniem, prawda?

```
procedure TForm1.FormPaint(Sender: TObject);
var
```

```
 prostokat: TRect; // zmienna pomocnicza, zawierająca współrzędne prostokąta
```

```
begin
```

```
 prostokat:=Rect(0,0,ClientWidth,ClientHeight); // funkcja Rect tworzy prostokąt o zadanych współrzędnych
 Canvas.FillRect(prostokat); //wypełnij prostokąt, czyli wyczyść płótno
 Canvas.Ellipse(prostokat); //narysuj elipsę wewnątrz prostokąta
```

```
end;
```

Teraz program działa prawidłowo również przy zmienianiu jego rozmiarów. Więcej na temat typu TRect i funkcji Rect znajdziesz w pomocy do Delphi, tutaj omówimy tylko metodę FillRect.

FillRect wypełnia zadany prostokąt aktualnym pędzlem. W naszym przypadku pędzel jest koloru tła formularza. Czym jest pędzel?

[edytuj] Pędzel

Pędzel (Brush) jest obiektem znajdującym się wewnątrz każdego płótna (Canvas). Pędzel ma szereg właściwości; służy do określenia tekstury, jaką będą wypełniane obiekty. Teksturą może być albo bitmapa, albo jedna ze standardowych tekstur (zobacz w pliku pomocy pod hasłem TBrush.Style). Pędzel ma również swój kolor. [edytuj] Pióro

Pióro (Pen) zawiera informacje o wyglądzie pióra na danym płótnie. Pióro ma kolor (Color), grubość (Width) oraz tryb (Mode). Dokładniejsze informacje znajdziesz w Pomocy.

Spróbujmy narysować drugą elipsę innym piórem, a także zastosować biały pędzel:

```
procedure TForm1.FormPaint(Sender: TObject);
var

 prostokat1, prostokat2: TRect;

begin

 prostokat1:=Rect(0,0,ClientWidth,ClientHeight);
 Canvas.Brush.Color:=clWhite; // clWhite to stała odpowiadająca za kolor biały
 Canvas.FillRect(prostokat1); // czyszczenie płótna

 // rysujemy pierwszą elipsę
 Canvas.Pen.Width:=2;
 Canvas.Pen.Color:=clRed; // clRed jest stałą odpowiadającą za kolor czerwony
 Canvas.Ellipse(prostokat1);

 // rysujemy drugą elipsę
 prostokat2:=Rect(10,10,ClientWidth-10,ClientHeight-10);
 Canvas.Pen.Width:=1;
 Canvas.Pen.Color:=clBlue; // clBlue jest stałą odpowiadającą za kolor niebieski
 Canvas.Ellipse(prostokat2);

end;
```

Zapamiętaj, że pióro odpowiada za rysowanie linii, a pędzel - za wypełnienia.

Pojedyncze punkty możesz obsługiwać za pomocą właściwości Pixels płótna. Właściwość ta jest tablicą, wymagającą podania dwóch współrzędnych X i Y. Odczytywanie jej daje w wyniku kolor danego punktu, zaś zapisywanie – ustawia zadanemu punktowi odpowiedni kolor.

**Zadanie:**

Narysuj na formularzu wypełniony białym kolorem prostokąt (o stałych wymiarach) o niebieskim brzegu szerokości 2 pixeli. Podpowiedź: pusty prostokąt rysuje metoda `FrameRect`, wypełniony prostokąt bez brzegu – `FillRect`, zaś wypełniony prostokąt z brzegiem – `Rectangle`.

**Rozwiązanie zadania**

Poniżej znajduje się metoda obsługująca zdarzenie `OnPaint` formularza, rysująca biały prostokąt z niebieskim obramowaniem o szerokości dwóch pikseli.

```
procedure TForm1.FormPaint(Sender: TObject);
var
 prostokat: TRect;

begin
 prostokat:=Rect(20,20,80,40); // współrzędne prostokąta
 // pędzel
 Canvas.Brush.Color:=clWhite; // clWhite to stała odpowiadająca za kolor biały
 // pióro
 Canvas.Pen.Width:=2;
 Canvas.Pen.Color:=clBlue; // clBlue jest stałą odpowiadającą za kolor niebieski

 Canvas.Rectangle(prostokat); // prostokąt

end;
```

Spróbuj też porównać w ostatniej linijce efekty działania metod `FillRect` oraz `FrameRect`.

Zadania 1.

Napisz program znajdowania objętości oraz pola powierzchni ściętego stożka. Użyj przycisków Wprowadź oraz Koniec.

Pogram sprawdza czy wczytane są poprawne i wydaje stosowny komunikat.

Zadania 1.a

Po podaniu A B C dla funkcji kwadratowej obliczyć:

-współrzędne wierzchołka i wydrukować w postaci W(.....,.....)

-punkty przecięcia z osiami X i Y wydrukować w postaci X1(.....,0), X2(.....,0) Y(0,.....)

Pogram sprawdza czy wczytane odpowiadają funkcji kwadratowej i wydaje stosowny komunikat.

Zadania 2.

Program zamieniający temperaturę w stopniach fahrenheit na Celsjusza i odwrotnie. Pogram sprawdza czy wczytane są poprawne i wydaje stosowny komunikat.

Dane do testowania:

C= -40 to F= -40

Zadania 3.

Program znajdujący idealną długość nóg kobiety na podstawie wzrostu i odwrotnie. Pogram sprawdza czy wczytane są poprawne i wydaje stosowny komunikat.

$$N=(w*(\sqrt{5}-1))/2$$

w-wzrost

n-długość nóg

Zadania 4.

Napisz program, który będzie obliczał pole pierścienia kołowego po podaniu Rd (większy) i Rm (mniejszy) wczytanych w okienkach. Program powinien sprawdzać wczytane dane. Użyj przycisków Wprowadź oraz Koniec.

Zadania 5.

Napisz program, który będzie obliczał pole oraz obwód wycinka oraz odcinka kołowego po podaniu R i alfa wczytanych w okienkach. Program powinien sprawdzać wczytane dane. Użyj przycisków Wprowadź oraz Koniec.

Zadania 6.

Napisz program, który będzie rozstrzygał położenie dwóch okręgów po podaniu promieni R1 i R2 oraz odległości między środkami. ( siedem położen) Program powinien sprawdzać wczytane dane. Użyj przycisków Wprowadź oraz Koniec.

Zadania 7.

Napisz program, który będzie rozstrzygał położenie okręgu i prostej po podaniu środka oraz promienia R okręgu, prosta podawana jest w postaci  $A*x+B*y+C=0$ . Program powinien sprawdzać wczytane dane. Użyj przycisków Wprowadź oraz Koniec. Możliwe trzy położenia. Wskazówka oblicz odległość D środka okręgu od prostej i na tej podstawie relacji między D i R rozstrzygnij jakie to położenie

Dane do testowania:

A=0, B=1, C=1, Xo=0,yo=0,R=1 → styczna

Zadania 8.

Program znajdujący idealną wagę po podaniu wzrostu i odwrotnie. Pogram sprawdza czy wczytane są poprawne i wydaje stosowny komunikat.

Zadania 9.

Napisz program, który będzie obliczał pole oraz obwód trapezu równoramiennego po podaniu jego boków wczytanych w okienkach. Program sprawdza wczytane dane i wydaje stosowny komunikat. Użyj przycisków Wprowadź oraz Koniec.

Zadania 10.

Napisz program, który będzie obliczał pole oraz obwód kwadratu po podaniu przekątnej wczytanych w okienkach. Program sprawdza wczytane dane i wydaje stosowny komunikat. Użyj przycisków Wprowadź oraz Koniec.

Zadania 11.

Napisz program, który będzie obliczał pole oraz obwód prostokąta po podaniu jego przekątnej oraz jednego z boków wczytanych w okienkach. Program powinien sprawdzać wczytane dane. Użyj przycisków Wprowadź oraz Koniec.

Zadania 12.

Napisz program, który będzie obliczał pole oraz obwód odcinka kołowego po podaniu R i alfa wczytanych w okienkach. Kąta w alfa podawany jest w stopniach. Program powinien sprawdzać wczytane dane. Użyj przycisków Wprowadź oraz Koniec.

Zadania 13.

Napisz program, który będzie obliczał pole oraz obwód rombu po podaniu jego przekątnych wczytanych w okienkach. Użyj przycisków Wprowadź oraz Koniec. Sprawdź wprowadzane dane. Przy wyprowadzaniu danych wypisywane są jednostki.

Zadania 14.

Napisz program, który będzie obliczał pole oraz obwód trójkąta równoramiennego po podaniu jego ramienia oraz wysokości wczytanych w okienkach. Program sprawdza czy wczytane są poprawne i wydaje stosowny komunikat. Przy wyprowadzaniu danych wypisywane są jednostki. Użyj przycisków Wprowadź oraz Koniec.

Dane do testowania:

Ramię=10

Wysokość=6

To

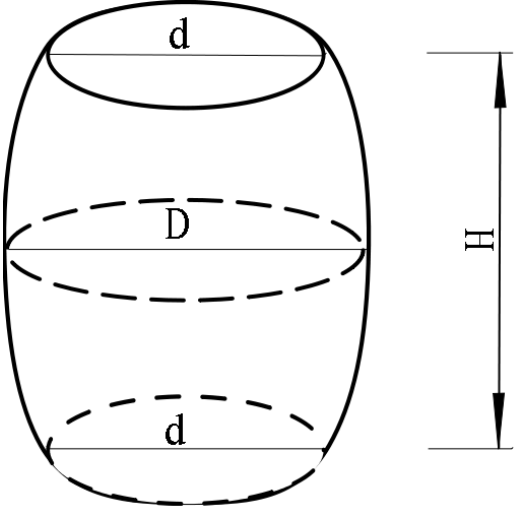
Pole=48 cm<sup>2</sup>

Obwód=36 cm

Zadania 15.

Napisz program znajdowania sumy dwóch liczb dwójkowych co najwyżej siedmiobitowych. Użyj przycisków Wprowadź oraz Koniec.

Zadania 16.

|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Objętość beczki oblicza się wg wzoru :<br/> <math>V = \frac{1}{12} \pi (2D^2 + d^2) H</math>,<br/>         Gdzie:<br/>         D - średnica w miejscu najszerszym,<br/>         d- średnica dna,<br/>         H- wysokość beczki.</p> <p>Oblicz objętość beczki po podaniu obwodu beczki w miejscu najszerszym OBW, promienia r dna oraz wysokości h beczki.</p> |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Zadania 17.

Napisz program zamiany liczby podanej w systemie dziesiętnym na:

Opcja 1

Na system dwójkowy

Opcja 2

Ósemkowy

Opcja 3

Szesnastkowy.

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.

Użyj przycisków Wprowadź oraz Koniec. Nazwa formularz to Twoje nazwisko a nie Form1.

Dane do testowania:

255 to →11111111

4095 to →7777

65535 to →FFFF

Zadania 18.

Napisz program, który będzie po podaniu wymiarów prostokąta  $a$  i  $b$  w metrach poda pole tego prostokąta w:

Opcja 1

Metrach kwadratowych

Opcja 2

hektarach

Opcja 3

arach

Opcja 4

Morgach (morga)

Opcja 5

Stopach kwadratowych

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.

Użyj przycisków Wprowadź oraz Koniec. Po wyniku podaj jednostkę.

Zadania 19.

Napisz program, który będzie po podaniu wymiarów prostopadłościanu  $a$  i  $b$  i  $c$  w decymetrach poda objętość tego prostokąta w:

Opcja 1

litrach

Opcja 2

galonach

Opcja 3

hektolitry

Opcja 4

Metry sześciennie.

Opcja 5

Calach sześciennych.

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.

Użyj przycisków Wprowadź oraz Koniec. Po wyniku podaj jednostkę.

Dane do testowania:

A=2

B=2

C=2

Litry=8

Galony=1,75

Hektolitry=0,08

Metry sześciennie=0,008

Cale sześciennie=488,19

#### Zadania 20.

Napisz program, który będzie po podaniu pięciu oporów elektrycznych R1, R2, R3, R4, R5 będzie :

Opcja 1

Opór zastępczy R1, R2 dla połączenia szeregowego

Opcja 2

Opór zastępczy R1, R2 dla połączenia równoległego

Opcja 3

Opór zastępczy R1, R2, R3 dla połączenia szeregowego

Opcja 4

Opór zastępczy R1, R2, R3 dla połączenia równoległego

Opcja 5

Opór zastępczy R1, R2, R3, R4, R5 dla połączenia równoległego

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageBox.

Użyj przycisków Wprowadź oraz Koniec. Po wyniku podaj jednostkę.

#### Zadania 21.

Napisz program, który będzie po podaniu odległości d w kilometrach poda odległość w:

Opcja 1

Metrach

Opcja 2

milach

Opcja 3

Stopach (angielskich)

Opcja 4

calach

Opcja 5



milimetrach

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.  
Użyj przycisków Wprowadź oraz Koniec. Po wyniku podaj jednostkę.

#### Zadania 22.

Napisz program, który będzie po podaniu ilości terabajtów poda ilość bitów w:

- Opcja 1
- Giga bitach
- Opcja 2
- Mega bitach
- Opcja 3
- kilobitach
- Opcja 4
- bajtach
- Opcja 5
- bitach

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.  
Użyj przycisków Wprowadź oraz Koniec. Po wyniku podaj jednostkę.

Użyj typu danych double, gdy liczba jest za duża oraz w tym przypadku użyj do konwersji FloatToStr.

#### Zadania 23.

Napisz program, który będzie po podaniu ilości jajek poda ilość w:

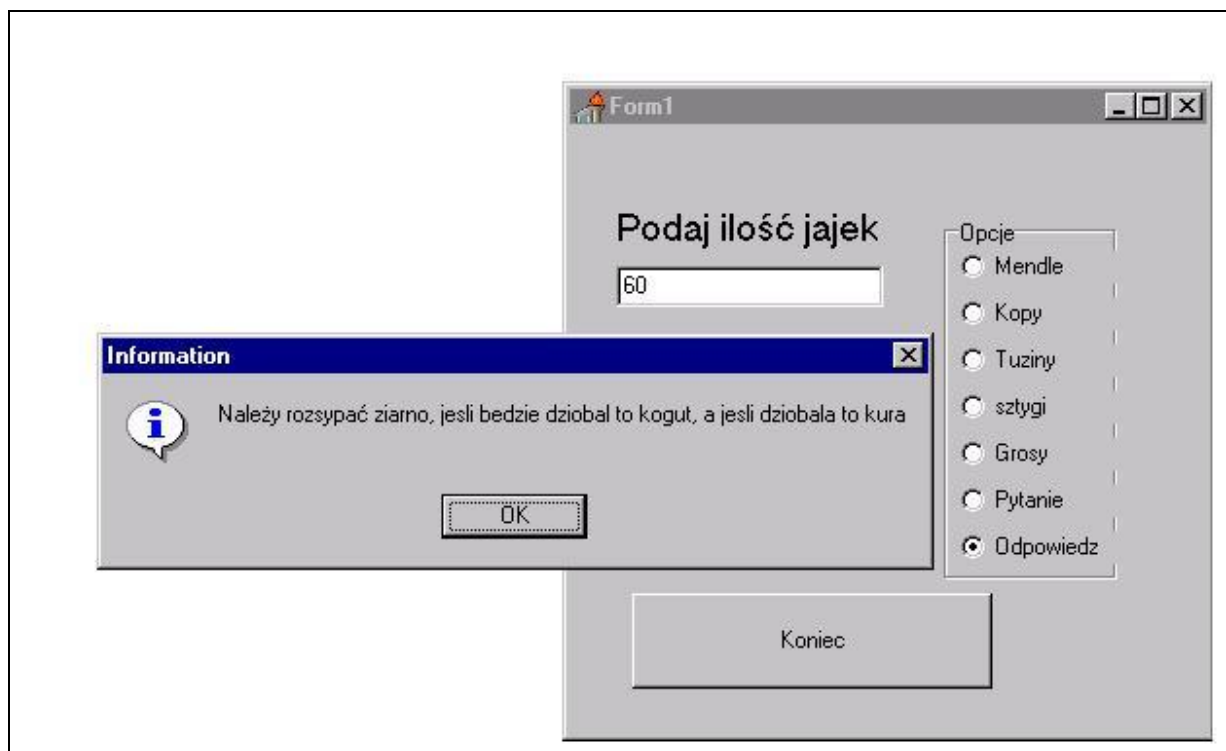
- Opcja 1
- mendlach
- Opcja 2
- kopach
- Opcja 3
- tuzinach
- Opcja 4
- szttyga
- Opcja 5
- gros

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.  
Użyj przycisków Wprowadź oraz Koniec. Po wyniku podaj jednostkę.

The image displays two screenshots of a Windows application titled "Form1".

**Top Screenshot:** The main window contains the text "Podaj ilość jajek" above a text box with the value "60". To the right is a group box "Opcje" containing a list of radio buttons: "Mendle" (selected), "Kopy", "Tuziny", "szytygi", "Grosy", "Pytanie", and "Odpowiedz". At the bottom is a button labeled "Koniec". An "Information" dialog box is open in the foreground, displaying an information icon, the text "ilosc mendli=4", and an "OK" button.

**Bottom Screenshot:** The main window is identical to the top one. However, the "Information" dialog box now displays the text "Jak odrzoznic kure od koguta ?" and the "Pytanie" radio button in the "Opcje" group is now selected.

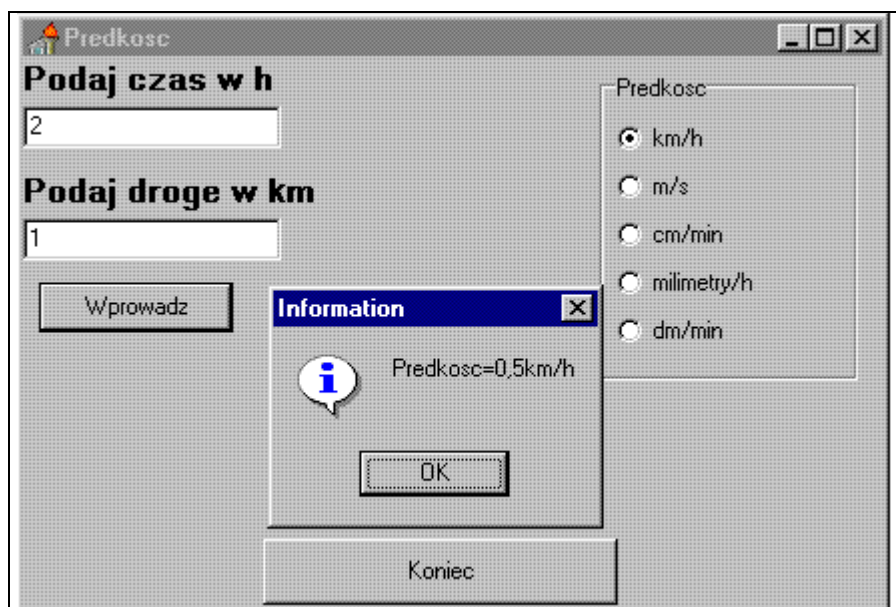


#### Zadania 24a

Wykonaj program obliczający prędkość po podaniu czasu w h i drogi w km.

Do wyboru mają być warianty:

- km/h;
- m/s;
- cm/min;
- mm/h;
- dm/min;



#### Zadania 24.

Napisz program, który będzie obliczał równanie podstawy trójkąta oraz wysokości opuszczonej na ten bok po podaniu współrzędnych trzech punktów

Opcja 1

obliczał oraz podawał wzór funkcji liniowej po podaniu punktów podstawy ( program będzie sprawdzał czy punkty nie pokrywają się oraz czy odcięte punktów nie są takie same ).

Opcja 2

obliczał oraz podawał wzór funkcji liniowej będącej wysokością. ( program będzie sprawdzał czy punkty nie pokrywają się oraz czy odcięte punktów nie są takie same ).

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.

Użyj przycisków Wprowadź oraz Koniec.

#### Zadania 25.

Napisz program, który będzie:

Opcja 1

obliczał oraz podawał wzór funkcji liniowej po podaniu dwóch punktów przez które będzie przechodziła ta funkcja ( program będzie sprawdzał czy punkty nie pokrywają się oraz czy odcięte punktów nie są takie same ).

Opcja 2

obliczał oraz podawał wzór funkcji liniowej po podaniu trzech punktów A B C. Prosta powinna być równoległa do prostej przechodzącej przez punkty A B i przechodzić przez punkt C. ( program będzie sprawdzał czy punkty nie pokrywają się oraz czy odcięte punktów nie są takie same ).

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.

Użyj przycisków Wprowadź oraz Koniec.

#### Zadania 26.

Napisz program, który będzie:

Opcja 1

obliczał oraz podawał wzór funkcji liniowej po podaniu punktu przez które będzie przechodziła ta funkcja oraz współczynnika kierunkowego

Opcja 2

obliczał oraz podawał wzór funkcji liniowej prostopadłej do danej i przechodzącej przez podany punkt.

Dane: prosta czyli współczynniki a i b

punkt przez, który ma przechodzić prosta prostopadła

Opcja 3

obliczał oraz miejsce zerowe funkcji liniowej po podaniu punktu przez które będzie przechodziła ta funkcja oraz współczynnika kierunkowego ( uwzględnij przypadek, że funkcja może nie mieć miejsca zerowego)

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.

Użyj przycisków Wprowadź oraz Koniec.

#### Zadania 27.

Napisz program, który będzie:

Opcja 1

obliczał oraz podawał wzór funkcji liniowej równoległej do danej i przechodzącej przez podany punkt.

Dane: prosta czyli współczynniki a i b

punkt przez, który ma przechodzić prosta równoległa

Opcja 2

obliczał punkt przecięcia z osią Y funkcji liniowej po podaniu punktu przez które będzie przechodziła ta funkcja oraz współczynnika kierunkowego

Opcja 3

obliczał oraz podawał wzór funkcji liniowej po podaniu punktu przez które będzie przechodziła ta funkcja oraz współczynnika kierunkowego

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.

Użyj przycisków Wprowadź oraz Koniec.

#### Zadania 28.

Napisz program, który będzie:

Opcja 1

obliczał oraz podawał wzór funkcji liniowej po podaniu dwóch punktów przez które będzie przechodziła ta

funkcja ( program będzie sprawdzał czy punkty nie pokrywają się oraz czy odcięte punktów nie są takie same ).  
Opcja 2

obliczał oraz podawał wzór funkcji liniowej po podaniu trzech punktów A B C. Prosta powinna być prostopadła do prostej przechodzącej przez punkty A B i przechodzić przez punkt C. ( program będzie sprawdzał czy punkty nie pokrywają się oraz czy odcięte punktów nie są takie same ).

Do wyboru opcji wybierz użyj przycisków. Do wyprowadzania danych użyj MessageDlg.

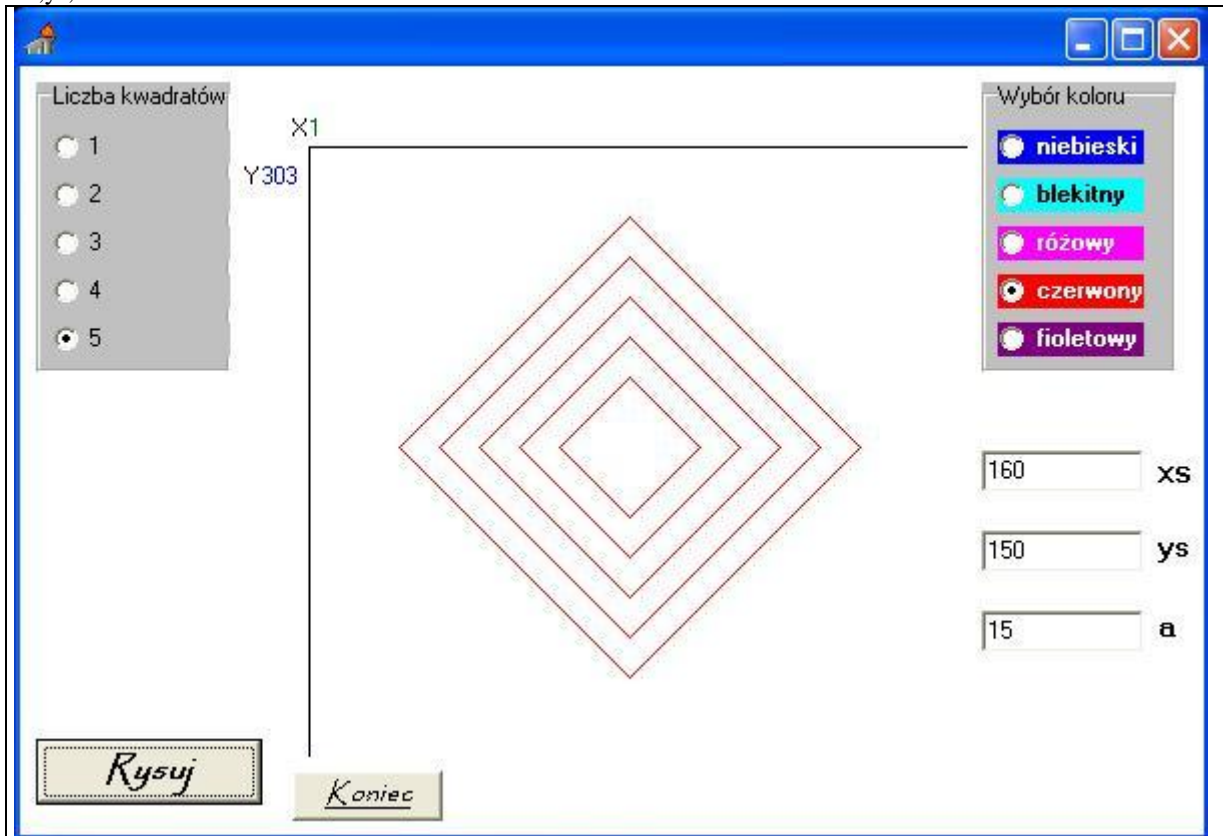
Użyj przycisków Wprowadź oraz Koniec.

Uwaga zmiana koloru rysowania

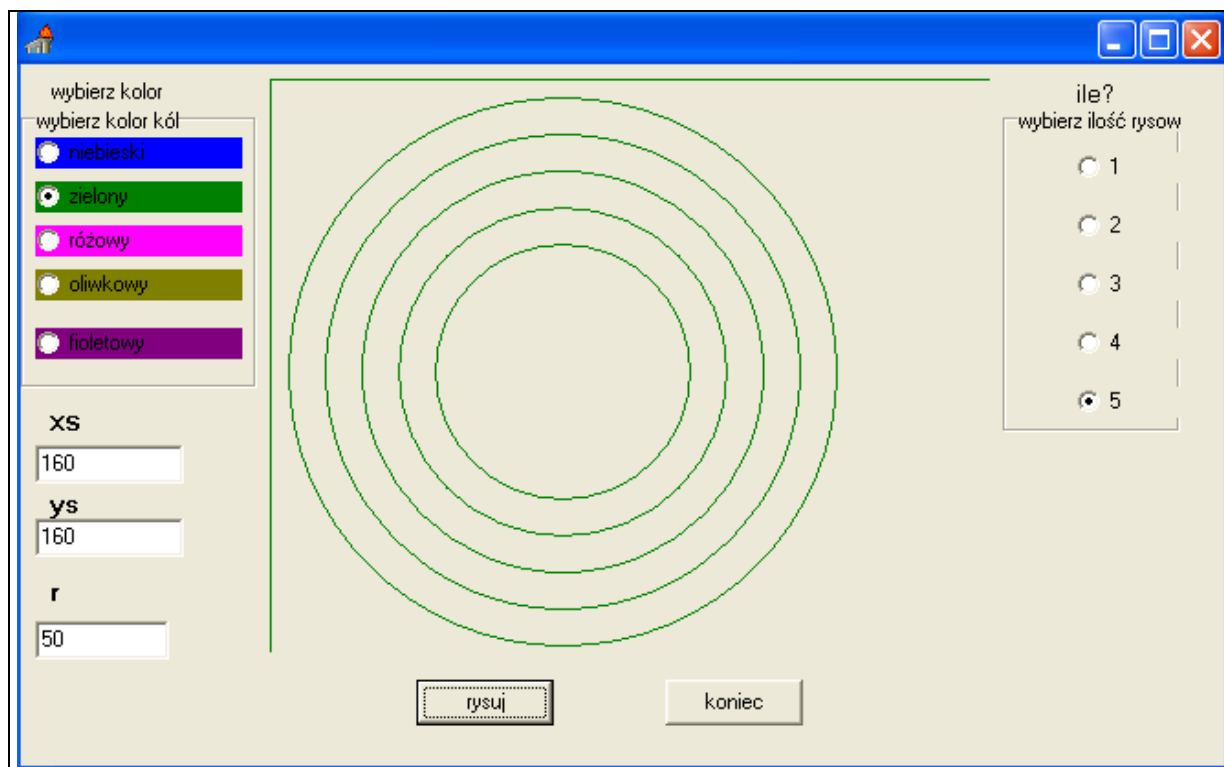
```
Form1.obraz.canvas.pen.color:=CIBlue;
```

Zadania 29.

Napisz program, który rysuje określoną liczbę kwadratów (1-5) obróconych o 45 stopni, o wybranych przez osobę kolorach (5 kolorów). Program ma zawierać button rysuj, zakończ, oraz trzy pola tekstowe do których należy wpisać  $x_s, y_s, a$ .

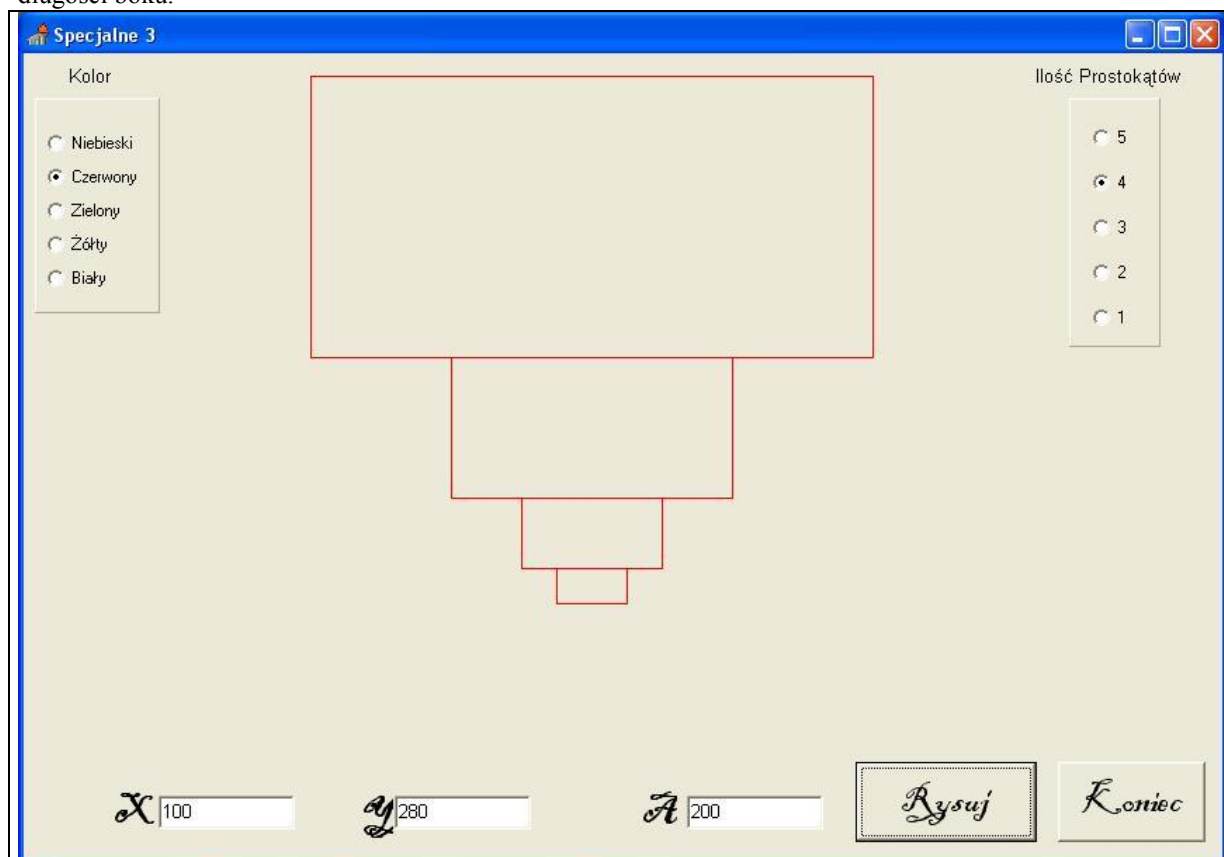
Zadania 30.

Napisz program, który rysuje okręgi z wybranego punktu ( $x_s, y_s$ ). Podaj promień okręgów a także zrób tak by była możliwość wyboru ilości rysowanych okręgów oraz ich kolor.



### Zadania 31.

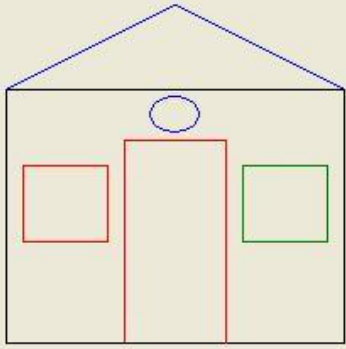
Program ma rysować wybraną ilość prostokątów oraz wybranym kolorem po podaniu współrzędnych oraz długości boku.



### Zadania 32.

Program rysujący domek z 6 elementami, każdy element musi mieć 4 kolory do wyboru. Domek w zmiennych wymiarach.

Form1



|                                             | Czarny                           | Niebieski                        | Czerwony                         | Zielony                          |
|---------------------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| <input checked="" type="checkbox"/> Budynek | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            |
| <input checked="" type="checkbox"/> Dach    | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/>            |
| <input checked="" type="checkbox"/> Okno1   | <input type="radio"/>            | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/>            |
| <input checked="" type="checkbox"/> Okno2   | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input checked="" type="radio"/> |
| <input checked="" type="checkbox"/> Drzwi   | <input type="radio"/>            | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/>            |
| <input checked="" type="checkbox"/> Swiatlo | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/>            |

Długość budynku  
200

Wysokość budynku  
150

Wysokość dachu  
50

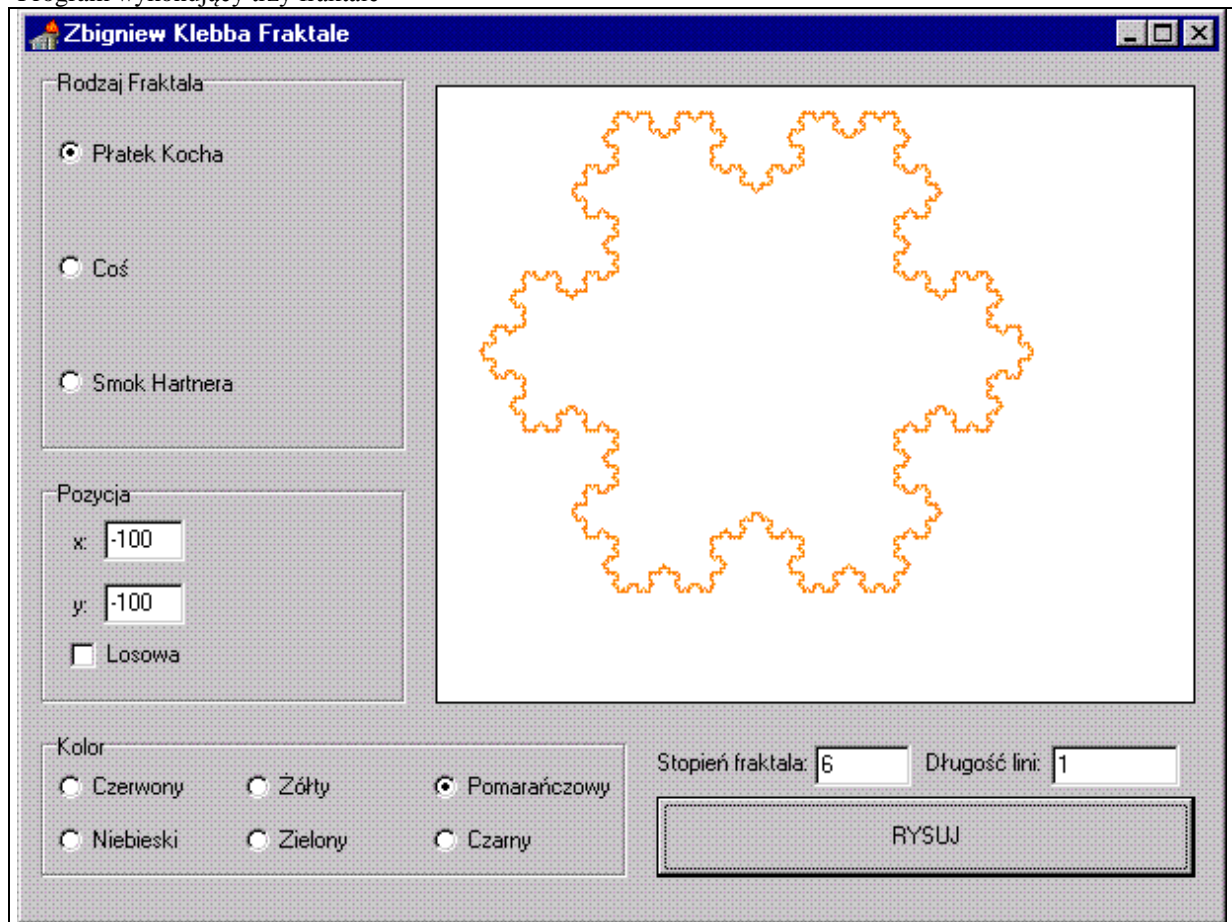
Rysuj

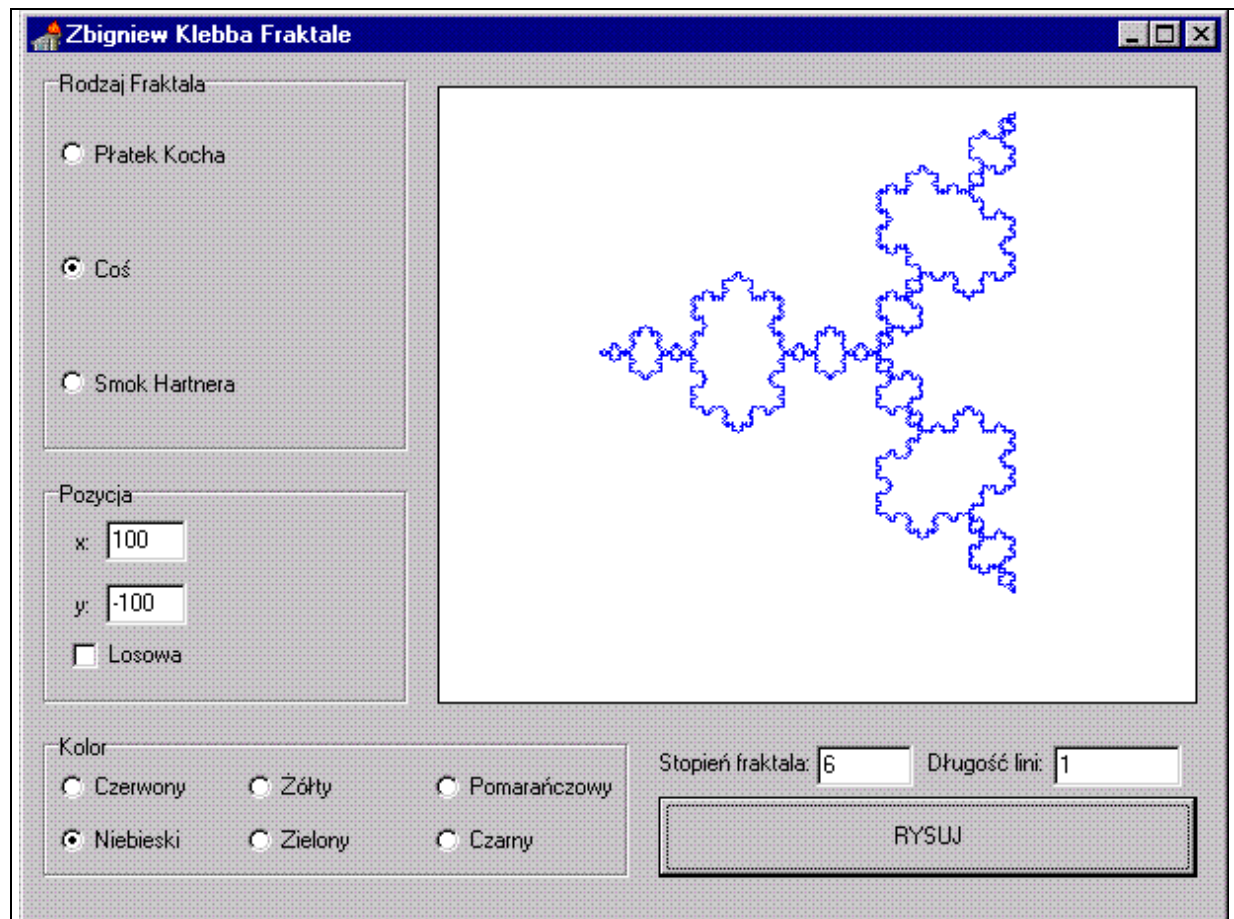
Koniec



Zadania 33.

Program wykonujący trzy fraktale





Zadania 34.

Program losuje 1000 liczb z danego przedziału, potem wypisuje w siatce StringGrid1 te, które tworzą kwadrat z kolejnymi liczbami (i na odwrót) mieszczący się w przedziale.

Zadania 35.

Napisz program znajdujący wszystkie czwórki liczb naturalnych, które spełniają warunek dla danego przedziału na max wczytanego w okienku.

$$a^2 + b^2 + c^2 + d^2 \leq \max$$

Użyj obiektu StringGrid1 do wypisywania kolejnych danych.

| Lp. | a   | a <sup>2</sup> | b   | b <sup>2</sup> | c   | c <sup>2</sup> | d   | D <sup>2</sup> | a <sup>2</sup> + b <sup>2</sup> +c <sup>2</sup> + d <sup>2</sup> | Max |
|-----|-----|----------------|-----|----------------|-----|----------------|-----|----------------|------------------------------------------------------------------|-----|
| 1   | 1   | 1              | 2   | 4              | 2   | 4              | 3   | 9              | 18                                                               | 30  |
| 2   | 2   | 4              | 2   | 4              | 3   | 9              | 3   | 9              | 26                                                               | 30  |
| ... | ... | ...            | ... | ...            | ... | ...            | ... | ...            | ...                                                              | ... |

Zadania 36.

Napisz program znajdujący wszystkie trójki liczb naturalnych, które spełniają warunek dla danego max wczytanego w okienku.

$$a^2 + b^2 + c^2 = \max$$

Użyj obiektu StringGrid1 do wypisywania kolejnych trójki.

| Lp. | a   | a <sup>2</sup> | b   | b <sup>2</sup> | c   | c <sup>2</sup> | a <sup>2</sup> + b <sup>2</sup> +c <sup>2</sup> | Max |
|-----|-----|----------------|-----|----------------|-----|----------------|-------------------------------------------------|-----|
| 1   | 1   | 1              | 2   | 4              | 2   | 4              | 9                                               | 9   |
| 2   | 2   | 4              | 2   | 4              | 1   | 1              | 9                                               | 9   |
| ... | ... | ...            | ... | ...            | ... | ...            | ...                                             | ... |

Zadania 37.

Napisz program znajdujący wszystkie trójki liczb naturalnych ( zero przyjmujemy jako liczba naturalna), które spełniają warunek dla danego przedziału na max wczytanego w okienku

$$a^3 + b^2 + c \leq \max$$

Użyj obiektu StringGrid1 do wypisywania kolejnych czwórek.

| Lp. | a   | a <sup>2</sup> | b   | b <sup>2</sup> | c   | a <sup>2</sup> + b <sup>2</sup> +c | Max |
|-----|-----|----------------|-----|----------------|-----|------------------------------------|-----|
| 1   | 1   | 1              | 2   | 4              | 2   | 7                                  | 20  |
| 2   | 2   | 4              | 2   | 4              | 1   | 9                                  | 20  |
| ... | ... | ...            | ... | ...            | ... | ...                                | ... |

Zadania 38.

Napisz program znajdujący wszystkie trójki liczb naturalnych ( zero przyjmujemy jako liczba naturalna), które spełniają warunek dla danego przedziału na **min** i **max** na liczby a,b,c wczytanego w okienku

$$a^2 + b^2 + c^2 \leq a \bullet b \bullet c$$

Użyj obiektu StringGrid1 do wypisywania kolejnych trójek.

| Lp. | a   | a <sup>2</sup> | b   | b <sup>2</sup> | c   | c <sup>2</sup> | a <sup>2</sup> + b <sup>2</sup> +c <sup>2</sup> | a*b*c | Min_a_b_c | Max_a_b_c |
|-----|-----|----------------|-----|----------------|-----|----------------|-------------------------------------------------|-------|-----------|-----------|
| 1   | 5   | 25             | 5   | 25             | 5   | 25             | 75                                              | 125   | 4         | 5         |
| 2   | 4   | 16             | 4   | 16             | 5   | 25             | 57                                              | 80    | 4         | 5         |
| ... | ... | ...            | ... | ...            | ... | ...            | ...                                             | ...   | ...       | ...       |

Zadania 39.

Napisz program znajdujący wszystkie trójki liczb, które mogą utworzyć trójkąt dla danego przedziału. Użyj obiektu StringGrid1( patrz przykład poprzedni ) do wypisywania kolejnych trójek.

| Lp. | a   |   | b+c | b   |   | a+c | c |   | a+b | Max_a_b_c |
|-----|-----|---|-----|-----|---|-----|---|---|-----|-----------|
| 1   | 1   | < | 4   | 2   | < | 3   | 2 | < | 3   | 7         |
| 2   | 7   | < | 14  | 7   | < | 14  | 7 | < | 14  | 7         |
| ... | ... |   |     | ... |   |     |   |   |     | ...       |
|     |     |   |     |     |   |     |   |   |     |           |

Zadania 40.

Napisz program znajdowania oraz wyprowadzania kolejnych liczb parzystych (większych od 3) wraz z ich rozkładem na liczby pierwsze. Użyj obiektu StringGrid1 do wypisywania kolejnych rozkładów.

| Lp. | Liczba parzysta | Pierwsza liczba pierwsza | Druga liczba pierwsza |
|-----|-----------------|--------------------------|-----------------------|
| 1   | 4               | 2                        | 2                     |
| 2   | 6               | 3                        | 3                     |
| 3   | 8               | 5                        | 3                     |
| ... | ...             | ...                      | ...                   |

Zadania 41.

Napisz program znajdujący wszystkie liczby, które należą do danego przedziału i są obliczane jak pokazuje tabela poniżej. Wstaw okienko do wartości min i max. są to liczby naturalne. Zwiększanie liczb między max a min o jeden.

Użyj obiektu StringGrid1( patrz przykład poprzedni ) do wypisywania kolejnych obliczonych liczb.

| Lp. | I   | I <sup>2</sup> | I <sup>3</sup> | Sqrt(I) | Pierw_czartego_stopnia(I) |
|-----|-----|----------------|----------------|---------|---------------------------|
| 1   | 10  | 100            | 1000           | 3.16    | 1.78                      |
| 2   | 11  | 121            | 1331           | 3.16    | 1.82                      |
| ... | ... |                |                | ...     |                           |
|     |     |                |                |         |                           |

Zadania 42.

Napisz program znajdowania oraz wyprowadzania na ekran rozstęp dla wszystkich kolumn. Rozstęp jest to różnica między maksymalną oraz minimalną wartością w rozpatrywanej kolumnie . Użyj tablicy z możliwością ustawiania tabeli za pomocą suwaków. Użyj przycisków Wprowadź oraz Koniec.

Zadania 43.

Napisz program znajdowania oraz wyprowadzania na ekran rozstępu macierzy. Rozstęp macierzy jest to różnica między maksymalną oraz minimalną wartością w rozpatrywanej macierzy . Użyj tablicy z możliwością ustawiania tabeli za pomocą suwaków. Użyj przycisków Wprowadź oraz Koniec.

Zadania 44.

Napisz program znajdowania oraz wyprowadzania na ekran rozstęp dla głównej przekątnej macierzy. Rozstęp jest to różnica między maksymalną oraz minimalną wartością . Użyj tablicy z możliwością ustawiania tabeli za pomocą suwaków. Użyj przycisków Wprowadź oraz Koniec.

Zadania 45.

Napisz program znajdowania oraz wyprowadzania na ekran minimalnej wielkości z maksymalnych wielkości wszystkich kolumn. Użyj tablicy z możliwością ustawiania tabeli za pomocą suwaków. Użyj przycisków Wprowadź oraz Koniec.

Zadania 46.

Napisz program znajdowania sumę wszystkich kolumn oraz znajdowania maksymalnej oraz minimalnej sumy kolumny. Użyj tablicy z możliwością ustawiania tabeli za pomocą suwaków. Użyj przycisków Wprowadź oraz Koniec.