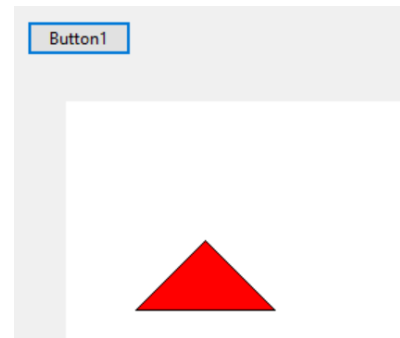


Príkaz polygon

Pre kreslenie ľubovoľných uzavretých čiar (polygónov) môžeme použiť príkaz **Polygon**. Ako vstup má množinu bodov, ktoré tvoria N-uholník.

```
Image1.Canvas.Polygon([Point(x1, y1), Point(x2, y2), Point(x3, y3), ...]);
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Image1.Canvas.Brush.Color:=clRed;
    Image1.Canvas.Polygon([Point(100,100), Point(50,150), Point(150,150)]);
end;
```



Konštanty

Konštanty sa podobajú premenným. Rovnako ako premenné, tak aj konštanty sú rezervované a pomenované časti pamäte, v ktorých je uložená nejaká hodnota. Od premenných sa líšia v tom, že ich hodnotu v programe nemožno meniť.

Prečo je vhodné požívať konštanty, keď sa ich hodnota v programe nemení?

Predstavme si, že sme vytvorili jednoduchý program, ktorý počíta náklady na dopravu z mesta A do mesta B, vzdialeného 25 kilometrov. Program funguje a programátor si začne pohrávať s myšlienkou jeho použitia pre mestá C a D, ktoré sú od seba vzdialené 30 kilometrov. Ak sa dal zlákať možnosťou ušetrenej práce a nepoužíval konštanty, musí v celom programe prepísať vzdialenosť z 25 na 30 km. Táto hodnota sa tam môže objavovať aj stokrát alebo tisíckrát. Každý si iste vie predstaviť náročnosť takejto zámeny. Problém rieši používanie konštánt. Pretože potom stačí prepísať starú hodnotu na novú len raz.

Konštanty sa deklarujú ešte pred premennými v časti, ktorá je uvedená kľúčovým slovom **const**. Konštanty sa podobajú premenným aj v tom, že sú rôzneho typu. Tu to však nemusí zadávať programátor, ale prekladač sám pozná, o aký typ konštanty ide podľa jej hodnoty.

Ukážky konštánt: N je typu integer, názov je typu string a znak typu char (s typmi string a char sa ešte stretneme ☺). PocetKilometrov je typu integer a konštante cesta sme nastavili hodnotu pomocou konštanty pocetKilometrov, avšak musí byť splnená podmienka, že pred použitím konštanty pocetKilometrov musela byť táto už definovaná.

```

procedure TForm1.Button1Click(Sender: TObject);

const
  N = 5;
  nazov = 'pondelok';
  znak = '@';
  pocetKilometrov = 25;
  cesta = 15*pocetKilometrov;

var a, b : integer;

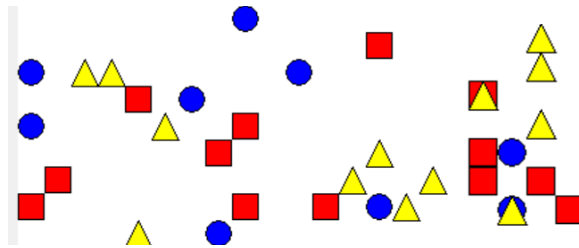
begin

  //náš program

end;

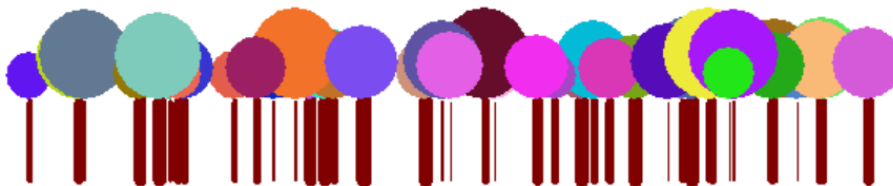
```

1. Napíšte procedúru `stvorec`, ktorá do obrázka nakreslí červený štvorec (dĺžka strany je a), kruh, ktorá nakreslí modrý kruh (priemer je a), trojuholník, ktorá nakreslí žltý rovnoramenný trojuholník (základňa je a , aj výška je a). Tlačidlo bude farebné útvary náhodne generovať do štvorcovej siete so štvorčekom veľkosti a (a je konštanta).

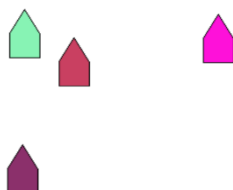


2. Napíšte program, ktorý na stlačenie tlačidla nakreslí 50 rôznofarebných stromov s náhodnou hrúbkou kmeňa (minimálne 1, maximálne 10) a s náhodnou farbou a náhodnou veľkosťou koruny (minimálne 30, maximálne 60). Na vykreslenie stromu použite procedúru `strom`, pričom strom má náhodnú x-ovú súradnicu, y-nová je vždy rovnaká (je konštanta).

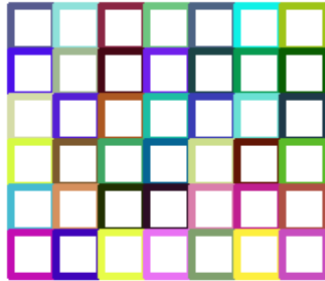
Na vypnutie a zapnutie pera použite vlastnosť: `pen.style:=psClear / psSolid`



3. Napíšte procedúru `dom`, ktorá bude pomocou príkazu `polygon` kresliť domček (strana domčeka a výška strechy domčeka je rovnaká). Tlačidlo bude pri každom kliknutí kresliť jeden domček na náhodnej pozícii.



4. Napíšte program, ktorý na stlačenie tlačidla nakreslí farebné štvorce (štvorcovú sieť). Štvorec nakreslite pomocou procedúry `stvorec`, ktorá bude kresliť rôznofarebný štvorec s hrúbkou pera 5, veľkosť štvorca aj počet štvorcov v riadku a stĺpci zadefinujte ako konštanty.



Formálne parametre procedúr

Aby sme si niektoré hodnoty nemuseli nastavovať pomocou premenných, ktoré sú pre tlačidlo lokálne ale pre procedúru globálne, môžeme tieto hodnoty do podprogramu poslať pomocou formálnych parametrov.

```
procedure TForm1.Button1Click(Sender: TObject);  
  
var strana : integer;  
  
procedure stvorec;  
begin  
    Image1.Canvas.Rectangle(10,30,10+strana,30+strana);  
end;  
  
begin  
    strana:=20;  
    stvorec;  
end;
```

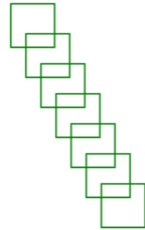
```
procedure TForm1.Button1Click(Sender: TObject);  
  
procedure stvorec (strana: integer);  
begin  
    Image1.Canvas.Rectangle(10,30,10+strana,30+strana);  
end;  
  
begin  
    stvorec(20);  
end;
```

Procedúra môže mať viac parametrov:

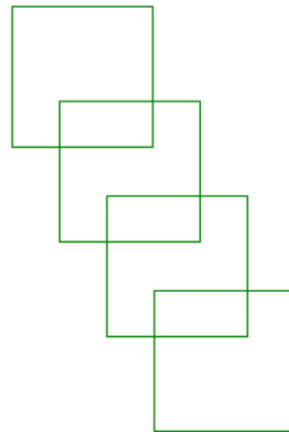
- **procedure** vypis(x, y: integer);
 - volanie procedúry: `vypis(30,40);`
- **procedure** cesta(d1, d2, d3: integer; cena: real);
 - volanie procedúry: `cesta(10,15,9,5.6);`

5. Napíšte procedúru `stvorce(n, a: integer)`, ktorá nakreslí n prekryvajúcich sa štvorcov, a je tretina strany štvorca, pričom každý ďalší štvorec sa začína v $2/3$ výšky a $1/3$ šírky štvorca.

`stvorce(7,10);`

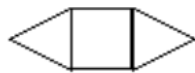


`stvorce(4,30);`



6. Procedúru `vzor(a, x, y: integer)`, ktorá bude kresliť štvorec aj výšku trojuholníkov veľkosti a , na pozícii x, y a procedúru `opasok(n: integer)`, ktorá vykreslí n vzorov.

`vzor(30,100,400);`



`opasok(7);`



7. Napíšte procedúru `kachlicka(a: integer)`, ktorá nakreslí jednu kachličku zloženú z trojuholníka a štvorca, kde a je veľkosť strany štvorca.



8. Napíšte procedúru `naramok(n, s: integer; farba: TColor)`, ktorá nakreslí opasok z n 4uholníkov pomocou príkazu `polygon`, kde s je výška aj veľkosť základne trojuholníka a parameter `farba` určuje farbu pera, ktorým sa náramok vykreslí.

`naramok(5, 20, clRed);`

