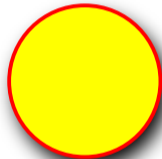


Objekty umiestňujeme na plátno v pravouhlej súradnicovej sústave. Zvolíme si vhodnú dĺžkovú jednotku, napr. 1 cm. Použijeme príkaz `unitsize(1cm)`. Pritom je dôležité si zapamätať, že príkazy ukončujeme bodkočiarkou. Nakreslíme si kruh s polomerom $r = 1$, pričom jeho stred umiestnime do bodu $S[-3, 0]$. Vnútro kruhu zafarbíme žltou farbou. Hraničnú kružnicu nakreslíme červenou farbou, pričom hrúbku čiary sme zvolili 0.5 mm. Zdrojový kód je tvorený príkazmi:

```
unitsize(1cm);  
fill(circle((-3,0),1),yellow);  
draw(circle((-3,0),1),red+linewidth(0.5mm));
```

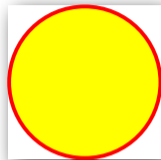


Pritom `red+linewidth(0.5mm)` možno skrátiť na `red+0.5mm`.

Obrázok má svoj minimálny ohraničujúci obdĺžnik (tzv. bounding box). Môžeme ho vyplniť napríklad bielou farbou, čím náš obrázok získa nepriehľadný podklad. Stačí pridať príkaz `shipout(bbox(Fill(white)))`.

Zdrojový kód je potom tvorený nasledujúcimi príkazmi:

```
unitsize(1cm);  
fill(circle((-3,0),1),yellow);  
draw(circle((-3,0),1),red+0.5mm);  
shipout(bbox(Fill(white))));
```



Ako vidíme na obrázku, súradnicová sústava sa nezobrazuje. Pri vytváraní grafiky by sme ju však potrebovali. Môžeme to riešiť napríklad dočasným umiestnením súradnicových osí a milimetrového papiera na plátno. Môžeme mať takýto milimetrový papier uložený v pomocnom súbore, napríklad s názvom `mm.pdf`. Ten potom umiestnime na plátno.

Niektoré inštalácie použitých programov spôsobujú, že sa nám pritom vygeneruje obrázok formátu A4. Vtedy skúste na začiatok zdrojového kódu pridať nasledujúci riadok:

```
texpreamble("\usepackage[nosetpagesize]{color,graphicx}");
```

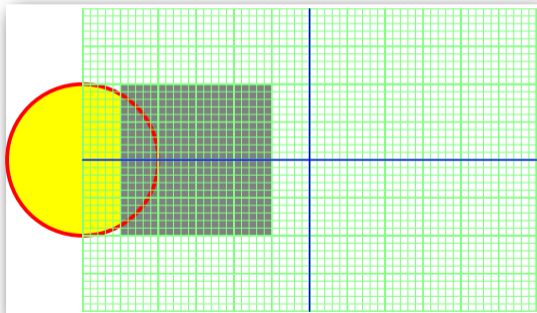
Vhodný milimetrový papier s vyznačenými súradnicovými osami si môžeme vygenerovať pomocou zdrojového súboru `mm.asy`, ktorý je tvorený nasledujúcimi príkazmi:

```
import graph; unitsize(1cm);
real a=-3, b=3; real c=-2, d=2; int i;
for(i=ceil(a);i<=b;++i){draw((i,c)--(i,d),squarecap+lightgreen+0.25mm);}
for(i=ceil(2*a);i<=2*b;++i){draw((i/2,c)--(i/2,d),squarecap+lightgreen+0.2mm);}
for(i=ceil(10*a);i<=10*b;++i){draw((i/10,c)--(i/10,d),squarecap+lightgreen+0mm);}
for(i=ceil(c);i<=d;++i){draw((a,i)--(b,i),squarecap+lightgreen+0.25mm);}
for(i=ceil(2*c);i<=2*d;++i){draw((a,i/2)--(b,i/2),squarecap+lightgreen+0.2mm);}
for(i=ceil(10*c);i<=10*d;++i){draw((a,i/10)--(b,i/10),squarecap+lightgreen+0mm);}
draw((a,0)--(b,0),blue+0.2mm); draw((0,c)--(0,d),blue+0.2mm);
clip(box((a,c),(b,d)));
```

Môžete si ho prispôbiť zmenou číselných hodnôt a , b , c , d . Odporúčame voliť tieto čísla tak, aby platilo $a = -b$, $c = -d$. Potom počiatok súradnicového systému bude v strede obrázku.

Ak máme milimetrový papier vhodných rozmerov uložený v pomocnom súbore s názvom `mm.pdf`, príkazom `label(graphic("mm.pdf"))` ho umiestnime na plátno. Pritom jeho stred bude v bode $[0,0]$. Preto sme súradnicové osi umiestnili na milimetrový papier tak, aby sa pretínali v jeho strede.

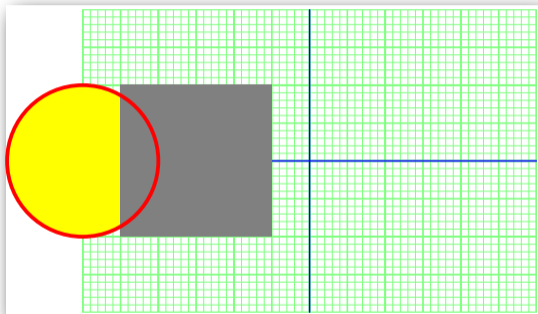
```
unitsize(1cm);  
label(graphic("mm.pdf"));  
fill(circle((-3,0),1),yellow);  
fill(box((-2.5,-1),(-0.5,1)),gray);  
draw(circle((-3,0),1),red+0.5mm);  
shipout(bbox(Fill(white)));
```



Ako vidíme, milimetrový papier prekrýva objekty, ktoré boli na plátno vložené neskôr.

Ak chceme, aby obrázok vložený príkazom `label` neprekrýval objekty na plátne, ktoré boli vložené neskôr, musí za ním nasledovať príkaz `layer()`.

```
unitsize(1cm);  
label(graphic("mm.pdf"));  
layer();  
fill(circle((-3,0),1),yellow);  
fill(box((-2.5,-1),(-0.5,1)),gray);  
draw(circle((-3,0),1),red+0.5mm);  
shipout(bbox(Fill(white))));
```

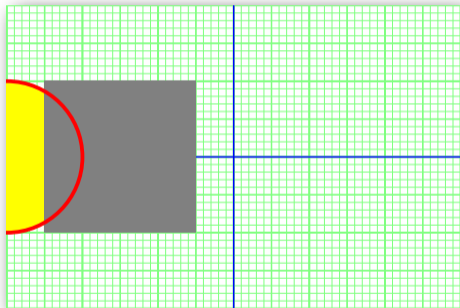


Obdĺžnik, ktorého strany sú rovnobežné so súradnicovými osami, sme vložili príkazom `box`, ktorého parametre sú ľavý dolný roh $A[-2.5, -1]$ a pravý horný roh $C[-0.5, 1]$.

Zámerne sme umiestnili objekty na plátne tak, aby sme názorne mohli ilustrovať orezanie.

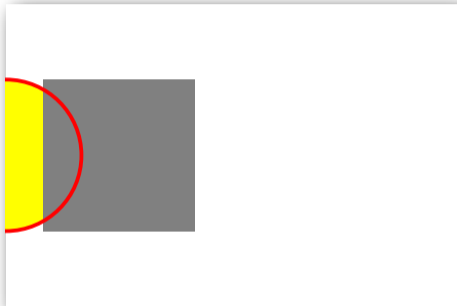
Obrázok orežeme vzhľadom na okraj vloženého milimetrového papiera. K tomu potrebujeme uzavretú krivku, vzhľadom na ktorú budeme obrázok orezávať. V našom prípade to bude obdĺžnik s vrcholmi v bodoch $[-3, -2]$, $[3, -2]$, $[3, 2]$ a $[-3, 2]$.

```
unitsize(1cm);  
label(graphic("mm.pdf"));  
layer();  
fill(circle((-3,0),1),yellow);  
fill(box((-2.5,-1),(-0.5,1)),gray);  
draw(circle((-3,0),1),red+0.5mm);  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white))));
```



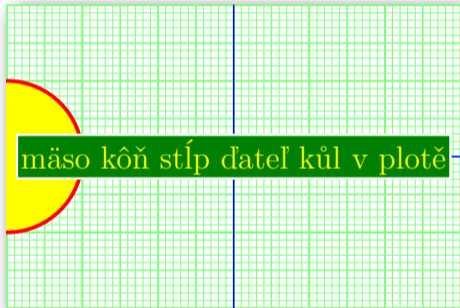
Na záver urobíme vložený obrázok s milimetrovým papierom neviditeľným. Stačí k tomu použiť nastavenie `invisible`.

```
unitsize(1cm);  
label(graphic("mm.pdf"),invisible);  
fill(circle((-3,0),1),yellow);  
fill(box((-2.5,-1),(-0.5,1)),gray);  
draw(circle((-3,0),1),red+0.5mm);  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white))));
```



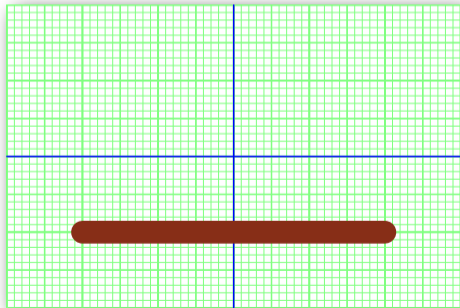
Do obrázku môžeme vložiť nápis v slovenčine. Stačí použiť príkaz `texpreable`, pomocou ktorého vložíme svoje obľúbené nastavenie slovenčiny/češtiny v L^AT_EXu.

```
texpreable("  
\usepackage[slovak]{babel}  
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage{lmodern}");  
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
fill(circle((-3,0),1),yellow);  
draw(circle((-3,0),1),red+0.5mm);  
label("mäso kôň stĺp ďateľ kúl v plotě",yellow+fontsize(12pt),  
      FillDraw(deepgreen,miterjoin+white+0.3mm));  
clip(box((-3,-2),(3,2))); shipout(bbox(Fill(white)));
```



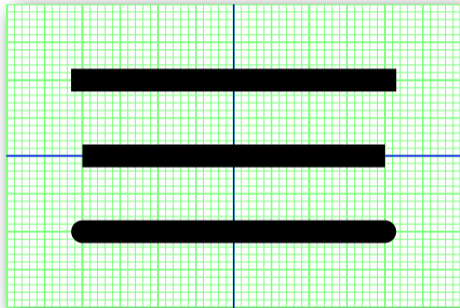
Nakreslíme si úsečku AB , kde $A[-2, -1]$ a $B[2, -1]$, širokú 3 mm. Farbu sme si namiešali sami. Použili sme farebný model RGB, pri ktorom svetlo želanej farby vzniká zmiešaním červeného (Red), zeleného (Green) a modrého (Blue) svetla požadovanej intenzity. K dispozícii máme 255 odtieňov každej farebnej zložky vyjadrené celými číslami 1 až 255, číslom 0 vyjadrujeme absenciu príslušnej farebnej zložky. Kód našej farby sme zapísali v šestnástkovej sústave. Namiesto `rgb("872E17")` sme mohli napísať `rgb(135/255,46/255,23/255)`; alebo `rgbint(135,46,23)`, to si však vyžaduje pridať na začiatok zdrojového kódu riadok `import x11colors;`

```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
draw((-2,-1)--(2,-1),rgb("872E17")+3mm);  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



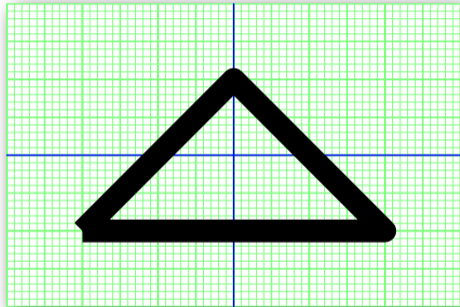
Nasledujúci obrázok ilustruje rôzne spôsoby ukončenia úsečky. Ako sme videli na predchádzajúcom obrázku, implicitné nastavenie je `roundcap`.

```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
draw((-2,1)--(2,1),extendcap+3mm);  
draw((-2,0)--(2,0),squarecap+3mm);  
draw((-2,-1)--(2,-1),roundcap+3mm);  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



Podobne môžeme nakresliť aj lomenú čiaru. Pokiaľ to má byť uzavretá čiara, nestačí, aby jej koncový bod bol totožný s jej počiatočným bodom.

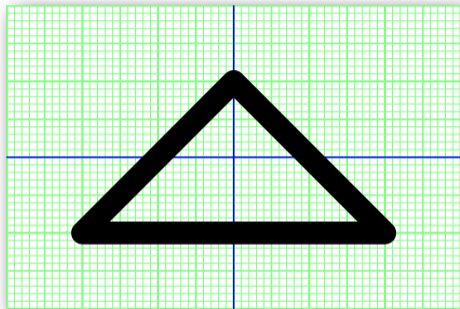
```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
draw((-2,-1)--(2,-1)--(0,1)--(-2,-1),  
      squarecap+3mm);  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



Že ide o uzavretú čiaru, musíme deklarovať. Namiesto koncového bodu (ktorý by tam bol zbytočný, keďže je totožný s počiatočným bodom čiar) napíšeme slovo **cycle**.

Nasledujúci obrázok ilustruje vykreslenie našej uzavretej lomenej čiary.

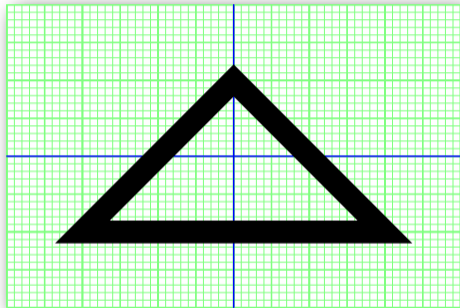
```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
draw((-2,-1)--(2,-1)--(0,1)--cycle,  
      squarecap+3mm);  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



Všimnime si, že nastavenie `squarecap` nemalo vplyv na spôsob zlomu. Na to slúži nastavenie `miterjoin`.

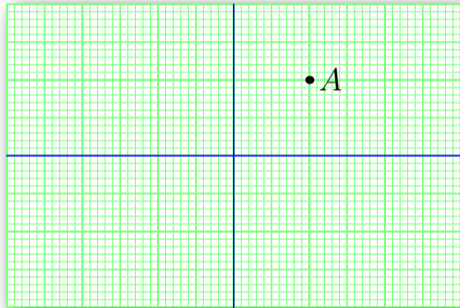
Nasledujúci obrázok už spĺňa všetky naše očakávania.

```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
draw((-2,-1)--(2,-1)--(0,1)--cycle,  
      miterjoin+3mm);  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



Na plátno umiestnime bod $A[1, 1]$, ako aj jeho označenie symbolom, ktoré umiestnime napravo od bodu A . Pritom na matematické výrazy používame L^AT_EX.

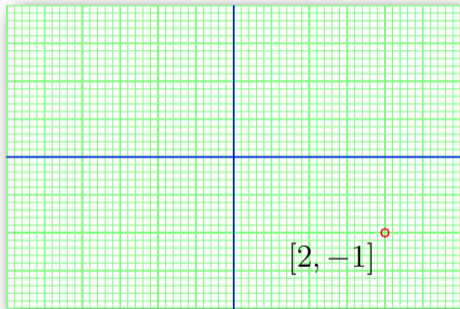
```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
pair A=(1,1);  
dot(A);  
label("$A$",A,right);  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



Keď nejaký objekt používame viackrát, je vhodné ho najskôr pomenovať. V našej ukážke sme bodu $A[1, 1]$ priradili písmeno A a deklarovali sme ho ako typ `pair`, čo je usporiadaná dvojica reálnych čísel. Samozrejme, mohli sme písať `dot((1,1))`, ako aj `label("A", (1,1), right)`.

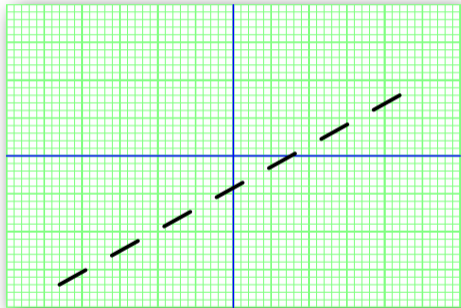
Bod môžeme vyznačiť aj prázdny krúžkom a jeho označenie môžeme umiestniť vľavo dole. Tiež môžeme nastaviť veľkosť fontu.

```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
pair B=(2,-1);  
dot(B,red+0.2mm,Fill(invisible));  
label("$[2,-1]$",B,down+left,fontsize(12pt));  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



Čiary môžu byť aj prerušované. Namiesto `dashed` môžeme použiť niektoré z nastavení `dotted`, `longdashed`, `dashdotted`, alebo `longdashdotted`.

```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
pair P=(2.2,0.8), Q=(-2.3,-1.7);  
draw(P--Q,dashed+0.5mm);  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



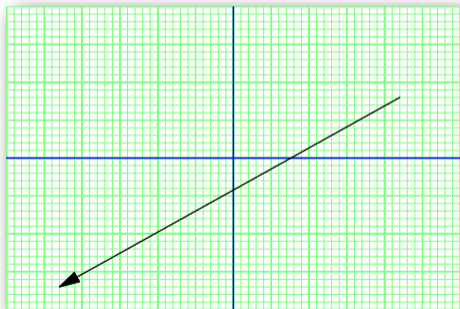
Úsečky môžeme ozdobiť šípkami.

Úsečku môžeme najskôr pomenovať. Čiary sú objekty typu `path`.

Našu úsečku PQ , kde $P[2.2, 0.8]$ a $Q[-2.3, -1.7]$, sme pomenovali `usecka`.

Šípku sme umiestnili do koncového bodu Q pomocou nastavenia `Arrow()`.

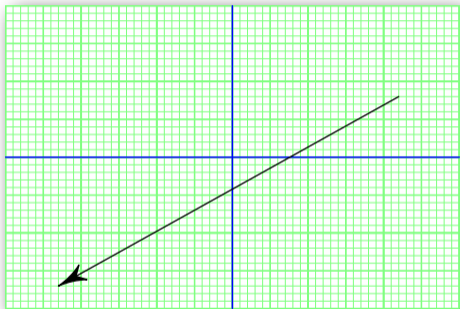
```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
pair P=(2.2,0.8), Q=(-2.3,-1.7);  
path usecka=P--Q;  
draw(usecka,Arrow());  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



Vyskúšajte si namiesto `Arrow()` napísať `MidArrow()`, prípadne `Arrows()`.

Môžeme meniť aj tvar šípky. Napríklad nastavenie HookHead.

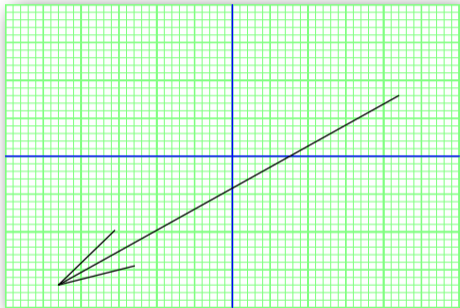
```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
pair P=(2.2,0.8), Q=(-2.3,-1.7);  
path usecka=P--Q;  
draw(usecka,Arrow(HookHead));  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



Vyskúšajte si namiesto HookHead napísať SimpleHead, prípadne TeXHead.

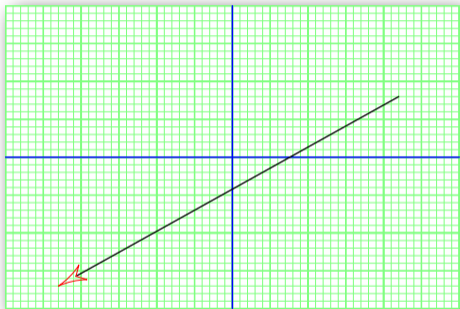
Môžeme meniť aj veľkosť šípky, napr. nastavením `size=1cm`.

```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
pair P=(2.2,0.8), Q=(-2.3,-1.7);  
path usecka=P--Q;  
draw(usecka,Arrow(SimpleHead,size=1cm));  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



Môžeme sa pohrať aj s farbami šípky.

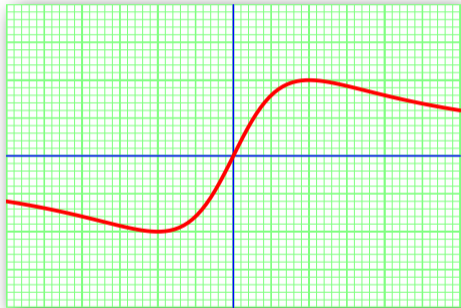
```
unitsize(1cm);  
label(graphic("mm.pdf")); layer();  
pair P=(2.2,0.8), Q=(-2.3,-1.7);  
path usecka=P--Q;  
raw(usecka,Arrow(HookHead,  
    filltype=FillDraw(invisible,red+0mm)));  
clip(box((-3,-2),(3,2)));  
shipout(bbox(Fill(white)));
```



Nastavenie hrúbky čiary na 0 mm nie je preklep. Takáto čiara sa vykreslí v hrúbke jeden pixel.

Nakreslíme si graf funkcie f , ktorá je definovaná predpisom $f(x) = \frac{2x}{1+x^2}$.
Pre tento účel potrebujeme najskôr načítať balík `graph`.

```
import graph;
unitsize(1cm);
label(graphic("mm.pdf")); layer();
real f(real x){return 2x/(1+x^2);}
draw(graph(f,-3,3),red+0.5mm);
clip(box((-3,-2),(3,2)));
shipout(bbox(Fill(white)));
```



Ak pre našu funkciu nechceme zaviesť označenie (tu to bolo `f`), môžeme jej graf vykresliť príkazom `draw(graph(new real(real x){return 2x/(1+x^2);},-3,3),red+0.5mm)`

Graf funkcie je v skutočnosti aproximovaný lomenou čiarou zloženou z úsečiek. Aby sme to mohli názorne vidieť, zmenšíme počet týchto úsečiek nastavením $n=5$.

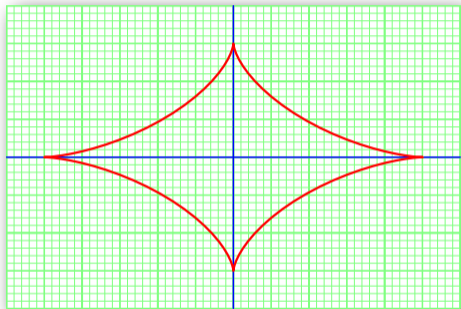
```
import graph;
unitsize(1cm);
label(graphic("mm.pdf")); layer();
real f(real x){return 2x/(1+x^2);}
draw(graph(f,-3,3,n=5),red+0.5mm);
dot(graph(f,-3,3,n=5),blue+0.2mm,
      Fill(invisible));
clip(box((-3,-2),(3,2)));
shipout(bbox(Fill(white)));
```



Implicitné nastavenie je $n=100$, čo väčšinou postačuje na vytvorenie ilúzie hladkej krivky. Kludne si vyskúšajte aj iné číselné hodnoty pre n .

Nakreslíme si krivku zadanú parametricky.

```
import graph;
unitsize(1cm);
label(graphic("mm.pdf")); layer();
real x(real t){return 2.5*cos(t)^3;}
real y(real t){return 1.5*sin(t)^3;}
draw(graph(x,y,0,2*pi),red+0.3mm);
clip(box((-3,-2),(3,2)));
shipout(bbox(Fill(white)));
```

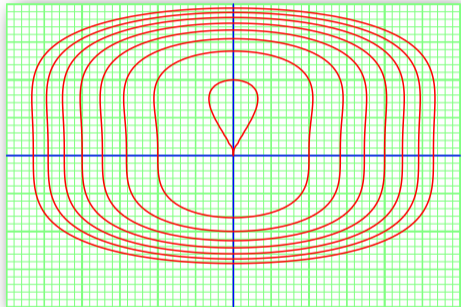


Nakreslíme krivky dané implicitne rovnicou $x^2 - y^3 + y^4 = c$ pre $c \in \{0, 1, 2, 3, 4, 5, 6, 7\}$. Ide vlastne o vrstevnice grafu funkcie dvoch premenných $F(x, y) = x^2 - y^3 + y^4$.

```
import contour;
import graph;
unitsize(1cm);
label(graphic("mm.pdf")); layer();
real F(real x,real y){return x^2-y^3+y^4;}
real[] c={0,1,2,3,4,5,6,7};
draw(contour(F,(-3,-2),(3,2),c),red+0.2mm);
clip(box((-3,-2),(3,2)));
shipout(bbox(Fill(white)));
```

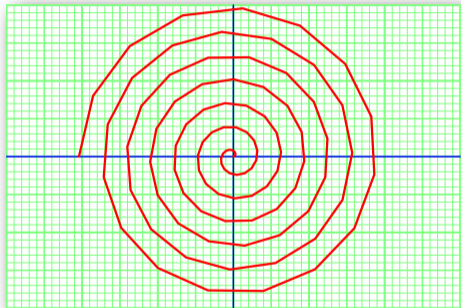
Ak chceme nakresliť len jednu z nich, napr. $x^2 - y^3 + y^4 = 3$, v zozname c bude len jedno číslo:

```
real[] c={3}
```



Môžeme nakresliť aj krivku zadanú v polárnych súradniciach. Na ukážku sme si zvolili špirálu definovanú predpisom $r = \varphi/20$ pre $0 \leq \varphi < 13\pi$.

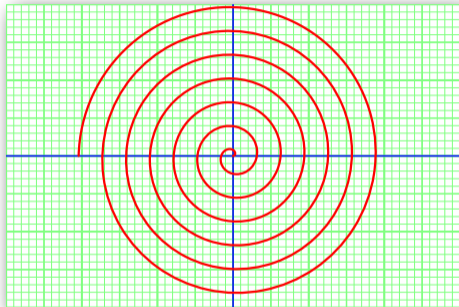
```
import graph;
unitsize(1cm);
label(graphic("mm.pdf")); layer();
real r(real phi){return phi/20;}
draw(polargraph(r,0,13*pi),red+0.3mm);
clip(box((-3,-2),(3,2)));
shipout(bbox(Fill(white)));
```



Opäť vidíme, že graf je vlastne aproximovaný lomenou čiarou zloženou z úsečiek. Nastavenie `operator..` spôsobí, že úsečky budú nahradené zakrivenými oblúkmi.

Tá istá špirála, ale vykreslená s nastavením `operator..` v príkaze `polargraph`.

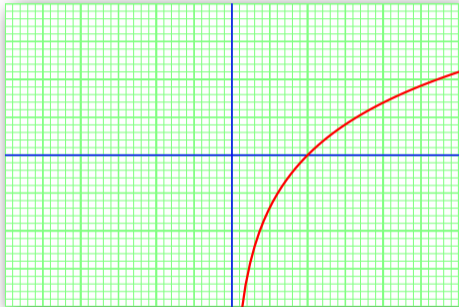
```
import graph;
unitsize(1cm);
label(graphic("mm.pdf")); layer();
real r(real phi){return phi/20;}
draw(polargraph(r,0,13*pi,operator..),red+0.3mm);
clip(box((-3,-2),(3,2)));
shipout(bbox(Fill(white)));
```



Nastavenie `operator..` môžeme používať napríklad aj v príkaze `graph`.

Jedno úskalí nás čeká v súvislosti s definičnými obormi. Keď chceme nakresliť napríklad graf funkcie $y = \ln x$, musíme explicitne uviesť podmienku, že $x > 0$. Takáto podmienka je typu `bool3`.

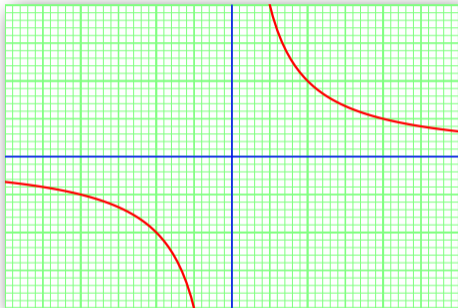
```
import graph;
unitsize(1cm);
label(graphic("mm.pdf")); layer();
bool3 len_pre_kladne(real x){return(x>0);}
draw(graph(log,-3,3,len_pre_kladne),red+0.3mm);
clip(box((-3,-2),(3,2)));
shipout(bbox(Fill(white)));
```



Určite ste si všimli, že prirodzený logaritmus má označenie `log`. Naproti tomu, desiatkový logaritmus má označenie `log10`.

Podobná situácia je pri grafe funkcie $y = 1/x$. Tu musíme explicitne uviesť podmienku, že $x \neq 0$.

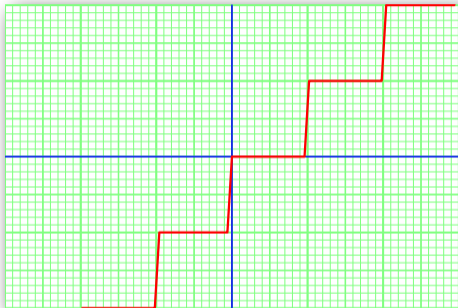
```
import graph;
unitsize(1cm);
label(graphic("mm.pdf")); layer();
real f(real x){return 1/x;}
bool3 okrem_nuly(real x){return(x!=0);}
draw(graph(f,-3,3,okrem_nuly),red+0.3mm);
clip(box((-3,-2),(3,2)));
shipout(bbox(Fill(white)));
```



Určite ste si všimli, že podmienka $x \neq 0$ sa zapisuje v tvare $x \neq 0$.

Pozrime sa na funkciu $y = \lfloor x \rfloor$ (dolná celá časť reálneho čísla).

```
import graph;
unitsize(1cm);
label(graphic("mm.pdf")); layer();
real f(real x){return floor(x);}
draw(graph(f,-3,3),red+0.3mm);
clip(box((-3,-2),(3,2)));
shipout(bbox(Fill(white)));
```



Graf je aproximovaný lomenou čiarou zloženou z úsečiek, čo nie je vhodné pre nespojitú funkciu.

Jednou z možností je vykresľovať graf funkcie $y = \lfloor x \rfloor$ postupne po jednotlivých intervaloch, ktoré neobsahujú jej body nespojitosti. Pretože ide o otvorené intervaly, ktorých krajné body sú celé čísla, musíme explicitne uviesť podmienku $x \notin \mathbb{Z}$, čo môžeme vyjadriť nerovnosťou $\lfloor x \rfloor < x$.

```
import graph; unitsize(1cm);
label(graphic("mm.pdf")); layer();
real f(real x){return floor(x);}
bool3 okrem_celych(real x){return(f(x)<x);}
draw(graph(f,-2,-1,okrem_celych),red+0.3mm);
draw(graph(f,-1,0,okrem_celych),red+0.3mm);
draw(graph(f,0,1,okrem_celych),red+0.3mm);
draw(graph(f,1,2,okrem_celych),red+0.3mm);
draw(graph(f,2,3,okrem_celych),red+0.3mm);
clip(box((-3,-2),(3,2)));
shipout(bbox(Fill(white)));
```

