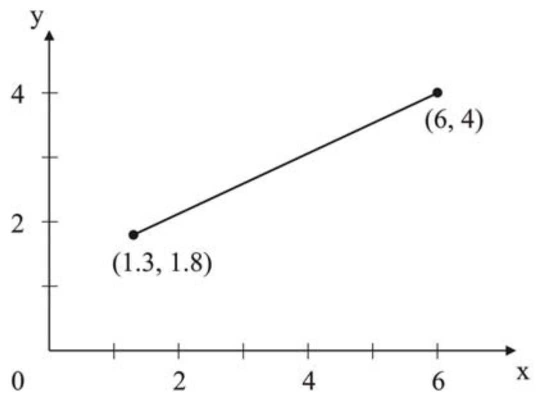
Základné grafické prvky

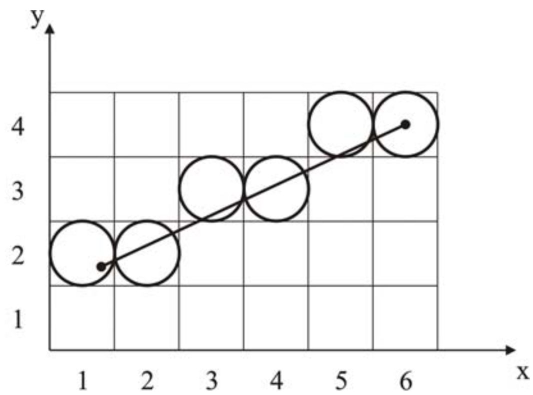
Podprogramy na vytváranie základných grafických prvkov sú určené na priame zobrazenie geometrických obrazcov na výstupných zariadeniach. Podľa typu výstupného zariadenia sú výsledkom algoritmov na ich vytváranie buď postupnosti bodov (pixlov) alebo postupnosti úsečiek. Prevod grafických prvkov na postupnosť pixlov sa volá <u>rasterizácia</u>.

Základné grafické prvky sú: *úsečky, kružnice, mnohouholníky, krivky, textové reťazce, vyplnené oblasti.*

<u>Úsečka</u>

- najčastejšie používané príkazy aplikačného programu sú príkazy na vykreslenie úsečky
- výstupné zariadenie musí vykresliť najkratšiu spojnicu medzi *2* zadanými koncovými bodmi
- analógové zariadenia - *vektorový displej* - úsečky sú kreslené ako súvislé čiary (obr. 1)
- číslicové zariadenia - *rastrový displej* - vytvárajú úsečku ako postupnosť bodov ležiacich medzi jej koncovými bodmi
- umiestnenie pixlov sa najprv vypočíta z rovnice priamky, na ktorej úsečka leží a potom sa zobrazia zodpovedajúce body na obrazovke
- súradnice pixlov sú celočíselné hodnoty, takže vykreslené čiary iba približne aproximujú skutočnú úsečku vedenú zadanými koncovými bodmi
- ak má byť napr. zobrazený bod (*1.3, 1.8*), tak je vykreslený pixel na pozícii (*1, 2*)





Obr. 1 Vektorové a rastrové zobrazenie úsečky

- toto zaokrúhľovanie na celočíselné hodnoty vytvára na vykreslených čiarach schodovité efekty, ktoré môžu byť na systémoch s malou rozlišovacou schopnosťou dosť výrazné (obr. 1)

- vzhľad rastrových čiar sa dá zlepšiť použitím systémov s vyššou rozlišovacou schopnosťou, a tiež pomocou metód na vyhladzovanie čiar

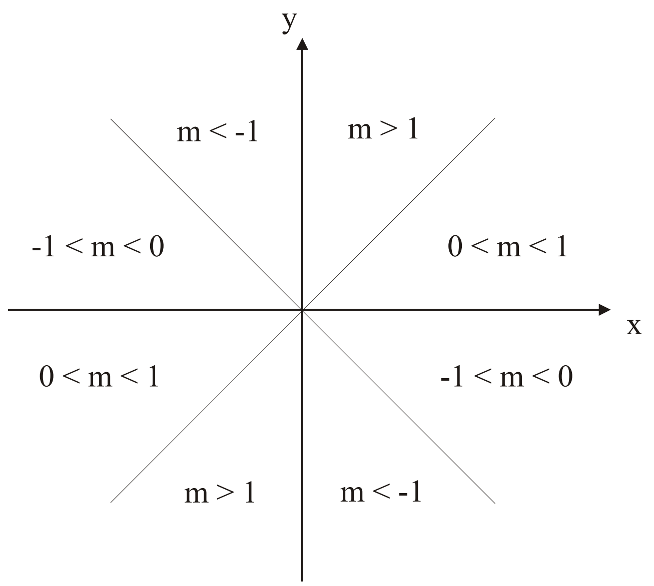
Algoritmy na vykreslenie úsečky

Algoritmus na vykreslenie úsečky z rovnice priamky, na ktorej úsečka leží

- priamku, ktorá nie je rovnobežná s osou *x*, ani s osou *y* môžeme vyjadriť rovnicou
  $$y = mx + b \tag{1}$$
  m - smernica, *b* - posun na osi *y*
- ak sú dané koncové body úsečky ($x1, y1$) , ($x2, y2$), potom smernica
  $$m = \frac{y2 - y1}{x2 - x1} = \frac{\Delta y}{\Delta x}$$
- ak $0 < m < 1$, potom sa úsečka vykresľuje tak, že novú súradnicu x_{i+1} vypočítame $x_{i+1} = x_i + 1$, a k nej prislúchajúcu súradnicu y_{i+1} vypočítame z rovnice $y_{i+1} = m.x_{i+1} + b$, a potom vykreslíme bod so súradnicami (round (x_{i+1}), round (y_{i+1}))
- ak $m > 1$, potom sa úsečka vykresľuje tak, že novú súradnicu y_{i+1} vypočítame $y_{i+1} = y_i + 1$, a k nej prislúchajúcu súradnicu x_{i+1} vypočítame z rovnice $y_{i+1} = m.\ x_{i+1} + b$, teda $x_{i+1} = (y_{i+1} - b)/m$ a potom vykreslíme bod so súradnicami (round (x_{i+1}), round (y_{i+1}))

Algoritmus DDA (Digital Differential Analyzer) - Prírastkový algoritmus

- je algoritmus na výpočet pozícií pixlov úsečky, založený na postupnom pripočítavaní konštantných prírastkov k obom súradniciam
- smernica môže nadobúdať rôzne hodnoty, ktoré závisia od sklonu úsečky (obr. 2)
- najprv sa budeme zaoberať úsečkou s kladnou smernicou
- ak je $0 \leq m \leq 1$, budeme meniť súradnicu *x* o jednotku a nasledujúcu súradnicu *y* vypočítame takto:
  $$x_{i+1} = x_i + 1$$
  $$m = \frac{y_{i+1} - y_i}{x_i + 1 - x_i} = y_{i+1} - y_i$$
  $$y_{i+1} = y_i + m$$
- v tomto prípade nazveme os *x riadiacou osou*
- pretože *m* nadobúda neceločíselné hodnoty, vypočítané súradnice *y* musíme zaokrúhľovať na celé čísla



Obr. 2 Hodnoty smernice m pre rôzne sklony úsečky

- pre úsečky s kladnou smernicou $m > 1$ je postup analogický, ale vymenia sa súradnice x a y
- to znamená, že o jednotku zvyšujeme súradnicu y a nasledujúcu súradnicu x dostaneme takto

  $y_{i+1} = y_i + 1$

  $$m = \frac{y_i + 1 - y_i}{x_{i+1} - x_i} = \frac{1}{x_{i+1} - x_i}$$

  $$x_{i+1} = x_i + \frac{1}{m}$$

- algoritmus sa dá zapísať pomocou procedúry, vstupnými parametrami ktorej sú súradnice začiatočného a koncového bodu úsečky ($x1$, $y1$) , ($x2$, $y2$)

  deltax = x2 - x1

  deltay = y2 - y1

- rozdiel s väčšou absolútnou hodnotou určuje hodnotu premennej *kolko_bodov*, ktorá určuje počet bodov vypočítaných pri vykresľovaní úsečky
- k súradniciam *x1, y1* sú postupne pripočítavané konštantné prírastky, a tak sú získané ďalšie body úsečky

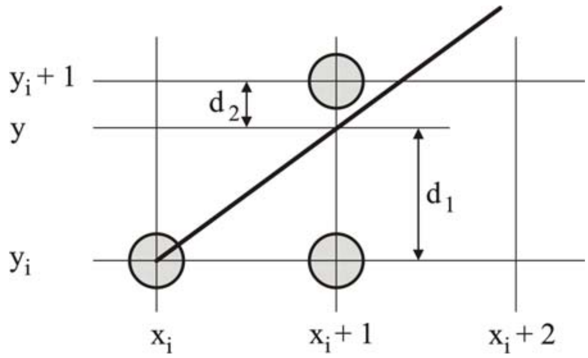
```
procedure Usecka_DDA ( x1, y1, x2, y2 : integer );
var     deltax , deltay , kolko_bodov , k : integer;
            x_prírastok, y_prírastok, x, y : real;
begin
        deltax := x2 - x1;
        deltay := y2 - y1;
        if abs (deltax) > abs (deltay)  then kolko_bodov := abs (deltax)
                                          else kolko_bodov := abs (deltay);
        x_prírastok := deltax / kolko_bodov;
        y_prírastok := deltay / kolko_bodov;
        x := x1; y := y1;
```

```
        Putpixel (round(x), round(y));
        for k := 1 to kolko_bodov do
                begin
                        x := x + x_prírastok;
                        y := y + y_prírastok;
                        Putpixel (round(x) , round(y));
                end;
end;
```

- algoritmus DDA je rýchlejší ako výpočet súradníc pixlov priamo z rovnice (1), ktorej použitie by znamenalo prevádzať násobenie pre každý vypočítaný pixel
- výpočet je spomaľovaný delením potrebným pre zistenie prírastkov, použitím typu REAL a zaokrúhľovaním
- navyše metóda nie je dosť presná pri vykresľovaní dlhých čiar, keď sa pričítavaním prírastku kumuluje chyba

Bresenhamov algoritmus

- tento algoritmus nájde body, ktoré ležia najbližšie ku skutočnej úsečke pomocou *celočíselnej aritmetiky*



Obr. 3 Výber pixlov úsečky

- začneme s úsečkou so smernicou $0 < m < 1$
- jej riadiaca os je os x
- pre jednotlivé súradnice x menené o 1 vypočítavame zodpovedajúce súradnice y
- po zobrazení bodu (x_i , y_i) sa má vybrať, či bude ďalší bod vykreslený na pozícii s tou istou súradnicou y alebo so súradnicou o 1 väčšou (obr. 3)
- pixel, ktorý vyberieme je ten, ktorého y - ová súradnica sa nachádza bližšie ku skutočnej hodnote y na danej úsečke
- predpokladáme, že pixel (x_i , y_i) už bol vykreslený a teraz máme rozhodnúť o ďalšom pixli
- dva možné pixle sú na pozíciách $(x_i + 1 , y_i)$ a $(x_i + 1 , y_i +1)$
- y je skutočná súradnica na úsečke v bode $x_i + 1$

$y = m\,(x_i + 1) + b$

$d_1 = y - y_i = m\,(x_i + 1) + b - y_i$

$d_2 = y_i + 1 - y = y_i + 1 - m\,(x_i + 1) - b$

- rozdiel týchto dvoch vzdialeností

$\Delta d = d_1 - d_2 = m\,(x_i + 1) + b - y_i - (y_i + 1 - m\,(x_i + 1) - b) =$

$$= m\,(x_i + 1) + b - y_i - y_i - 1 + m\,(x_i + 1) + b = 2m(x_i + 1) - 2\,y_i + 2b - 1 \qquad (2)$$

- podľa premennej Δd môžeme ľahko určiť, ktorý z dvoch pixlov leží bližšie ku skutočnej úsečke

$\Delta d < 0$, $d_1 - d_2 < 0$, $d_1 < d_2$, teda bližšie leží pixel so súradnicou y_i

$\Delta d > 0$, $d_1 - d_2 > 0$, $d_1 > d_2$, teda bližšie leží pixel so súradnicou $y_i + 1$

- teda skutočná hodnota Δd nie je dôležitá, ale jej znamienko
- vplyvom m nie je premenná Δd celočíselná
- rovnicu (2) môžeme previesť do celočíselnej aritmetiky vynásobením Δx

$\Delta d = 2m(x_i + 1) - 2\,y_i + 2b - 1 =$

$$= 2\,\frac{\Delta y}{\Delta x}\,(x_i + 1) - 2y_i + 2b - 1 \qquad /.\Delta x$$

$p_i = \Delta d\ \Delta x = 2\,\Delta y\,(x_i + 1) - 2\,\Delta x\,y_i + 2\,b\,\Delta x - \Delta x =$

$= 2\,\Delta y\,x_i - 2\,\Delta x\,y_i + 2\,\Delta y + 2b\,\Delta x - \Delta x$

$$p_i = 2\,\Delta y\,x_i - 2\,\Delta x\,y_i + \underbrace{2\,\Delta y + \Delta x\,(2\,b - 1)}_{\text{konšt.}} \qquad (3)$$

- rovnica (3) je stále zložitá pre výpočty súradníc
- zjednodušíme ju ďalšími úpravami
- p_i - podľa jej znamienka určujeme súradnice nasledujúcich pixlov

$p_i < 0$, $\Delta d < 0$, $d_1 < d_2$, vyberieme bod $(x_i + 1\,,\,y_i)$

$p_i > 0$, $\Delta d > 0$, $d_1 > d_2$, vyberieme bod $(x_i + 1\,,\,y_i + 1)$

vypočítame p_{i+1} *z* p_i *tak, že namiesto i dosadíme i+1*

vypočítame rozdiel $p_{i+1} - p_i$ *a z neho vyjadríme* p_{i+1} *pomocou* p_i

$x_{i+1} = x_i + 1$

$y_{i+1} = y_i + 1$

$p_{i+1} = 2\,\Delta y\,x_{i+1} - 2\,\Delta x\,y_{i+1} + \text{konšt.}$

$p_{i+1} - p_i = 2\,\Delta y\,x_{i+1} - 2\,\Delta x\,y_{i+1} + \text{konšt.} - (2\,\Delta y\,x_i - 2\,\Delta x\,y_i + \text{konšt.}) =$

$= 2\,\Delta y\,(x_i + 1) - 2\,\Delta x\,y_{i+1} + \text{konšt.} - 2\,\Delta y\,x_i + 2\,\Delta x\,y_i - \text{konšt.} =$

$= 2\,\Delta y\,(x_i + 1 - x_i) - 2\,\Delta x\,y_{i+1} + 2\,\Delta x\,y_i = 2\,\Delta y - 2\,\Delta x\,(y_{i+1} - y_i)$

$p_{i+1} - p_i = 2\,\Delta y - 2\,\Delta x\,(y_{i+1} - y_i)$

- pomocou tejto rovnice môžeme vypočítať každé ďalšie p z predchádzajúcej hodnoty
- ak $p_i < 0$, $\Delta d < 0$, $d_1 < d_2$, $y_{i+1} = y_i$, $y_{i+1} - y_i = 0$, $p_{i+1} = p_i + 2\,\Delta y$
- ak $p_i > 0$, $\Delta d > 0$, $d_1 > d_2$, $y_{i+1} = y_i + 1$, $p_{i+1} = p_i + 2\,\Delta y - 2\,\Delta x$

Na začiatku musíme vypočítať p_1

$$p_1 = 2\,\Delta y\,x_1 - 2\,\Delta x\,y_1 + 2\,\Delta y + \Delta x\,(2\,b - 1) = 2\,(y2 - y1)\,x_1 - 2\,(x2 - x1)\,y_1 + 2\,\Delta y + 2\,b\,\Delta x - \Delta x$$

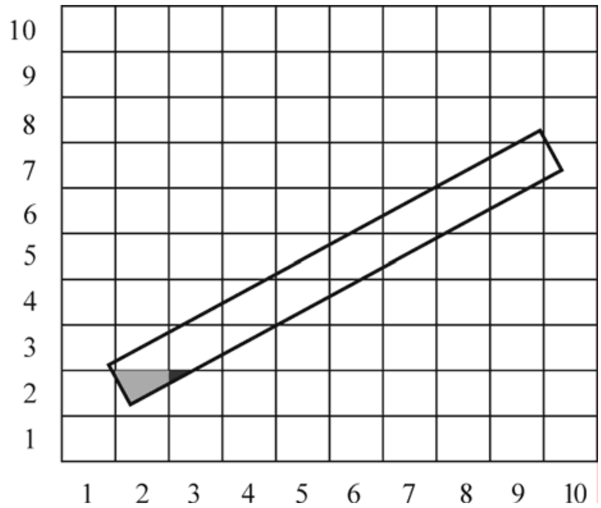
$$p_1 = 2\,\Delta y - \Delta x$$

```
procedure Usecka_Bresenham ( x1, y1, x2, y2 : integer );
        var     deltax, deltay, konst1, konst2 : integer;
                x, y, x_koncove, p : integer;
begin
        if x1 > x2 then  begin
                                x := x2; y := y2; x_koncove := x1;
                         end
                   else
                         begin
                                x := x1; y := y1; x_koncove := x2;
                         end ;
        deltax := x_koncove – x;
        deltay := abs (y1 – y2);
        konst1 := 2 * deltay;
        konst2 := 2 * (deltay – deltax);
        p := 2 * deltay – deltax;
        Putpixel (x , y);
        while x < x_koncove do
                begin
                        x := x + 1;
                        if p < 0 then p := p + konst1
                                else
                                        begin   y := y + 1;
                                                p := p + konst2;
                                        end;
                        Putpixel (x , y);
                end;
end;
```

Vyrovnávanie úsečiek

- použitím uvedených metód sa dajú na obrazovke vykresliť čiary, ktoré majú schodovitý vzhľad
- na zjemnenie tohto vzhľadu použijeme metódy vyrovnávania a vyhladzovania – sú založené na upravovaní jasu pixlov tvoriacich úsečku
- dajú sa použiť iba na zariadeniach, ktoré dovoľujú vykreslenie pixlov s niekoľkými stupňami jasu

- hlavnou myšlienkou je, že geometrické objekty na obrazovke majú určitú veľkosť (body a úsečky)



Obr. 4 Úsečka položená do rastra

- úsečky majú nenulovú hrúbku – *1* pixel
- pixel nie je matematicky nekonečne malý bod, ale je to svetelná plocha pokrývajúca malú časť obrazovky
- na obr. 4 je zobrazená úsečka nenulovej hrúbky položená do rastra
- každý obrazový bod je zobrazený obdĺžnikom
- obrazový bod prekrývajúci sa s úsečkou danej hrúbky bude mať *intenzitu* zobrazenia *úmernú spoločnej časti prieniku plôch*
- na tomto príklade vidíme, že pre bielu úsečku na čiernom pozadí má bod (*2, 2*) v spoločnom prieniku *50* % (teda vysvietime ho *50* %-nou intenzitou z max. intenzity) a bod (*3, 2*) približne len *10* % (teda *10* %-ou intenzitou)
- tieto algoritmy na výpočet spoločnej plochy sú náročné na strojový čas