3. NOȚIUNI MATEMATICE FOLOSITE ÎN APLICAȚII DE GRAFICĂ PE CALCULATOR

3.1. Sisteme de coordonate. Coordonate carteziene.

Sistemele de coordonate reprezintă o metodă de localizare a punctelor în spațiu, un mod de caracterizare a obiectelor geometrice (linii, cercuri, plane) prin ecuații algebrice, aceasta fiind concepția de bază a reprezentării în geometria analitică.

O coordonată carteziană este o tuplă ordonată de numere, reprezentate prin (x,y) în spațiul bidimensional (2D) şi (x,y,z) în spațiul tridimensional (3D), care descrie distanța de la origine la punct, măsurată de-a lungul fiecărei axe. Valorile pot fi pozitive sau negative. Axele sistemului sunt perpendiculare şi se intersectează în originea sistemului de coordonate. Acest sistem (figura 3-1) poate fi generalizat pentru a reprezenta punctul în spațiul n-dimensional nD prin n-tupla $(x_1,x_2,....,x_n)$. Sistemul de coordonate în spațiul n-dimensional constă în n axe perpendiculare reciproc şi intersectându-se în origine.

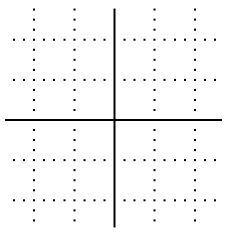


Figura 3-1. Sistem de coordonate carteziene 2D.

Originea unui sistem de coordonate este punctul de coordonate (0,0,....0) în care toate axele sistemului de coordonate se intersectează. Orice linie într-un sistem de coordonate poate fi folosită ca o axă pentru realizarea transformărilor geometrice. Axele principale sunt cele ce definesc sistemul de coordonate. Spre exemplu: în cazul sistemului de coordonate carteziene în 2D, axele principale sunt cele definite de ecuațiile: x=0, y=0.

3.2. Coordonate polare

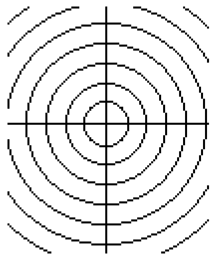


Figura 3-2. Sistem de coordonate polare 2D.

Coordonatele polare în 2D reprezintă o pereche ordonată (r,θ) definită astfel: pentru un punct P, r este distanța de la origine la P iar θ este unghiul dintre axa X şi segmentul generat de punctul P şi origine. Se pot considera mai multe valori valide pentru θ (valoare din intervalul $[0,2\pi]+2\pi$). Folosirea sistemului de coordonate polare poate simplifica în anumite cazuri calculele de reprezentare.

3.3. Conversia între sistemele de coordonate carteziene şi polare (în spaţiul bidimensional, 2D)

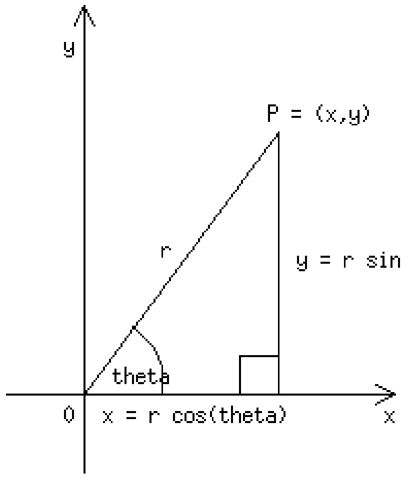


Figura 3-3. Determinarea coordonatelor carteziene ale unui punct caracterizat prin coordonate polare.

Pentru conversia coordonatelor polare în coordonate carteziene se folosesc relaţiile:

$$(3\text{-}1)\quad \begin{cases} x = r * \cos(\theta) \\ y = r * \sin(\theta) \end{cases}$$

Relaţiile de conversie din coordonate carteziene în coordonate polare sunt:

$$(3\text{-}2)\quad \begin{cases} r = \sqrt{x^2 + y^2} \\ \theta = \operatorname{atan}(y / x) \end{cases}$$

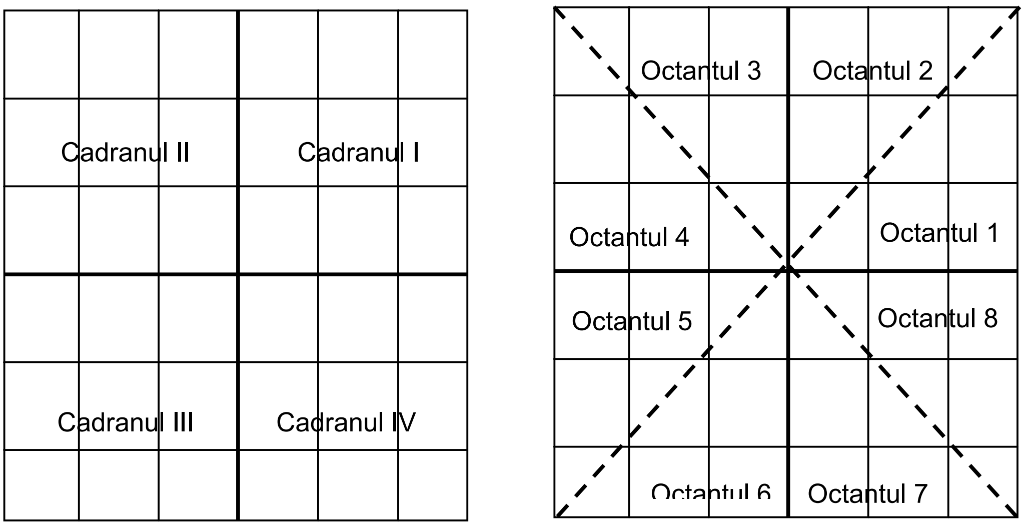


Figura 3-4. Împărţirea unei grile bidimensionale în cadrane şi octante.

O grilă bidimensională poate fi împărţită în patru subdiviziuni numite cadrane caracterizate prin semnul coordonatelor punctelor din fiecare zonă. Pentru un punct M(*x,y*), primul cadran este caracterizat de relaţia $x * y > 0$, al doilea cadran de relaţiile $x < 0, y > 0$, al treilea cadran de relaţiile $x < 0, y < 0$, al patrulea cadran prin relațiile $x > 0, y < 0$. Fiecare cadran se împarte în câte două octante numerotate în sens trigonometric (figura 3-4).

3.4. Linii

Fie o dreaptă ce trece prin punctele (x_1,y_1); (x_2,y_2) într-un sistem de coordonate cartezian 2D (figura 3-5). Atunci ecuația dreptei caracterizată de pantă este:

$$\text{(3-3)} \quad y = m * x + n\,,$$

unde panta dreptei este:

$$\text{(3-4)} \quad m = \frac{y_2 - y_1}{x_2 - x_1}),$$

iar parametrul *n* reprezintă ordonata din punctul în care dreapta intersectează axa Y.

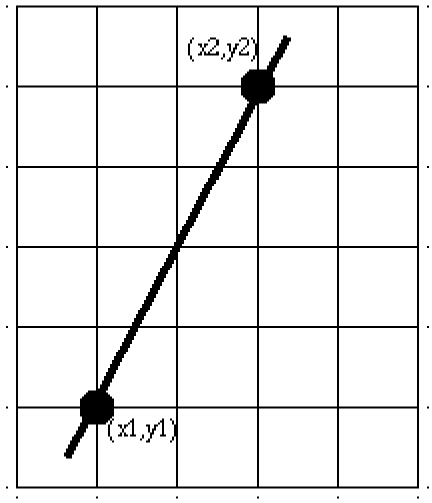


Figura 3-5. Caracterizarea unei linii într-un sistem cartezian 2D.

3.4.1. Ecuația parametrică a unei drepte

Fie dreapta *d* caracterizată de punctele $P_1(x_1,y_1)$ şi $P_2(x_2,y_2)$. Forma parametrică a ecuației dreptei *d* este:

$$\text{(3-5)} \quad \begin{cases} x = x_1 + t * (x_2 - x_1) \\ y = y_1 + t * (y_2 - y_1) \end{cases}, \quad 0 \le t \le 1$$

Pentru t=0 se obține chiar punctul $P_1(x_1,y_1)$, iar pentru t=1 relațiile sunt verificate de punctul $P_2(x_2,y_2)$. Variația lui t între 0 şi 1 conduce la obținerea punctelor de pe segmentul [P_1P_2].

3.4.2. Ecuația parametrică a unei drepte într-un sistem cartezian 3D

Fie punctele $P_1(x_1,y_1,z_1)$ şi $P_2(x_2,y_2,z_2)$. Ecuația parametrică a dreptei care trece prin cele două puncte este:

$$\text{(3-6)} \quad \begin{cases} x = x_1 + t * (x_2 - x_1) \\ y = y_1 + t * (y_2 - y_1) \\ z = z_1 + t * (z_2 - z_1) \end{cases}, \quad 0 \le t \le 1.$$

Dacă se doreşte caracterizarea unei drepte în spaţiul cartezian 3D ce trece printr-un punct P_1 şi este paralelă cu vectorul V=[x_v, y_v, z_v], atunci:

$$(3\text{-}7)\quad \begin{cases} x = x_1 + t * x_v \\ y = y_1 + t * y_v \\ z = z_1 + t * z_v \end{cases},\ 0 \le t \le 1.$$

3.4.3. Algoritm de determinare a distanţei dintre un punct şi o dreaptă într-un sistem cartezian de coordonate 2D

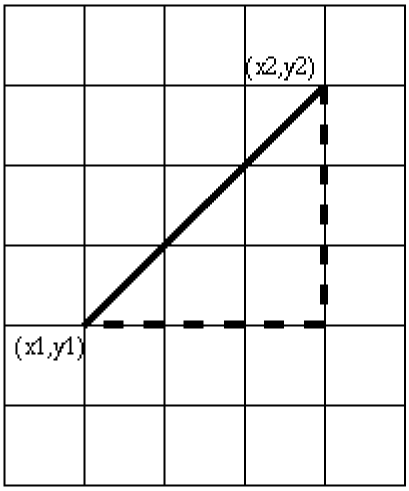


Figura 3-6. Distanţa dintre două puncte într-un sistem de coordonate carteziene 2D.

Distanţa între două puncte $P_1(x_1,y_1)$ şi $P_2(x_2,y_2)$ este dată într-un sistem cartezian de coordonate 2D (figura 3-6) de teorema lui Pitagora:

$$(3\text{-}8)\quad d(P_1,P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\ .$$

Distanţa între un punct şi o dreaptă reprezintă lungimea segmentului ce are un capăt în P_1 şi celălalt capăt în proiecţia P_2 a lui P_1 pe dreaptă (figura 3-7).

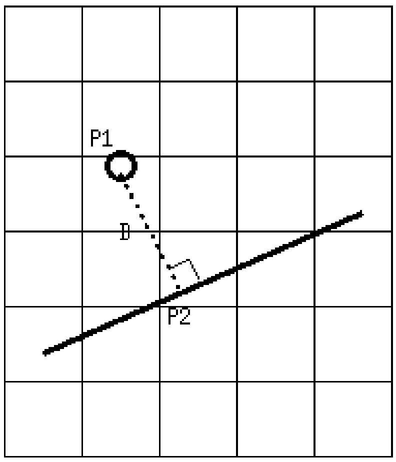


Figura 3-7. Distanţa dintre un punct şi o dreaptă într-un sistem cartezian de coordonate 2D.

Algoritmul de determinare a distanţei dintre P_1 şi proiecţia sa P_2 pe dreaptă este descris în continuare:

#1. Se determină panta $m' = -1 / n$

#2. Se determină ecuația dreptei generate de segmentul P_1P_2 astfel încât P_1 să aparțină dreptei (relația 3-3).

#3. Se determină coordonatele punctului P_2 ca intersecție a celor două drepte, rezolvând sistemul de ecuații prin care se verifică faptul că P_2 aparține fiecăreia dintre cele două drepte.

#4. Se calculează distanța dintre P_1 şi P_2, (conform relației 3-8).

3.4.4. Determinarea intersecției a două drepte

Fie două drepte d_1 şi d_2. Intersecția lor este punctul de coordonate (*x,y*) ce satisface ecuațiile dreptelor:

$$\text{(3-9)} \quad \begin{cases} y = m_1 * x + n_1 \\ y = m_2 * x + n_2 \end{cases}.$$

Două drepte paralele nu vor avea puncte de intersecție. Pantele dreptelor paralele: sunt identice: $m_1=m_2$. Două drepte paralele având termenii liberi identici, $n_1=n_2$, se confundă. În cazul în care se doreşte determinarea intersecției a mai mult de două drepte, trebuie folosită o metodă mult mai generală pentru rezolvarea ecuațiilor simultane, metoda eliminării (Gauss).

3.4.5. Algoritm de determinare a apartenenței unui punct de pe o linie dată la un segment dat

Pentru a determina dacă punctul $P_3(x_3,y_3) \in [P_1P_2]$ se parcurg următorii paşi:

#1. Se determină valoarea *t* din ecuația parametrică pentru *x* sau pentru *y*:

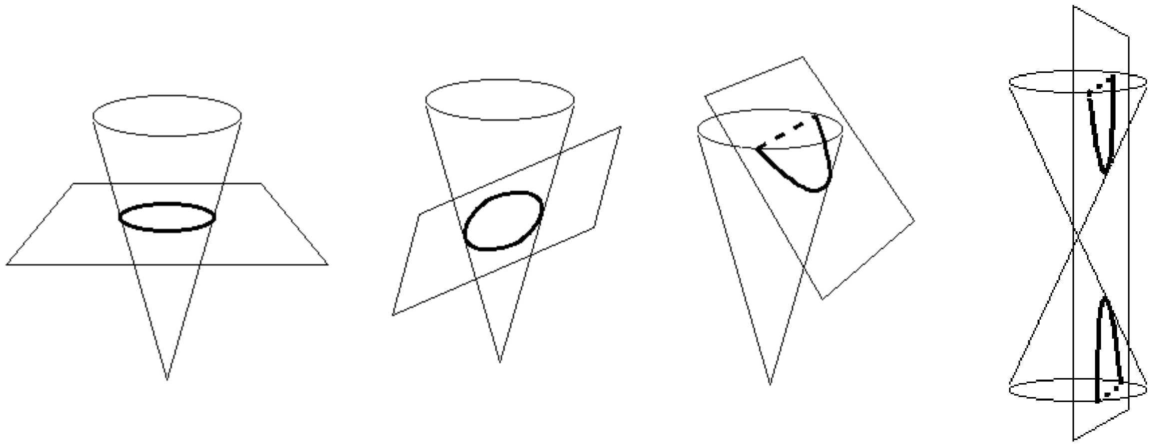
$$t = \frac{x - x_1}{x_2 - x_1}$$

#2. Se pleacă de la ipoteza $x = x_3$ şi se calculează *t*.

#3. Dacă $0 \le t \le 1$, atunci $P_3 \in [P_1P_2]$.

3.5. Secțiuni conice

Secțiunile conice se obțin prin intersecția unui con cu un plan care nu trece prin vârful conului. Fie planul caracterizat de ecuația $ax+by+cz=1$. Dacă unghiul dintre plan şi axa conului este drept, intersecția planului cu suprafața conică are ca rezultat secțiunea circulară (a=b=0, figura 3-8a). Dacă unghiul dintre plan şi axa conului este cuprins între valoarea unghiului ce descrie conul şi unghiul drept, atunci intersecția planului cu conul este o elipsă ($a^2+b^2<c^2$, figura 3-8b). Primul tip de secțiune conică prezentat (cercul) este prin urmare un caz particular de elipsă, pentru care unghiul dintre planul de intersecție şi axa conului este unghiul drept, una dintre extremele domeniului ce defineşte secțiunea eliptică. În cazul în care unghiul format de planul de secțiune cu axa conului este egal cu unghiul ce descrie conul, intersecția cu pânza conului este practic vidă. O parabolă se formează la intersecția unui plan cu conul, cu condiția ca planul să fie paralel cu o generatoare a conului ($a^2+b^2=c^2$, v. figura 3-8c). Hiperbola se obține prin intersecția unui plan paralel cu axa conului ($a^2+b^2>c^2$, figura 3-8d).



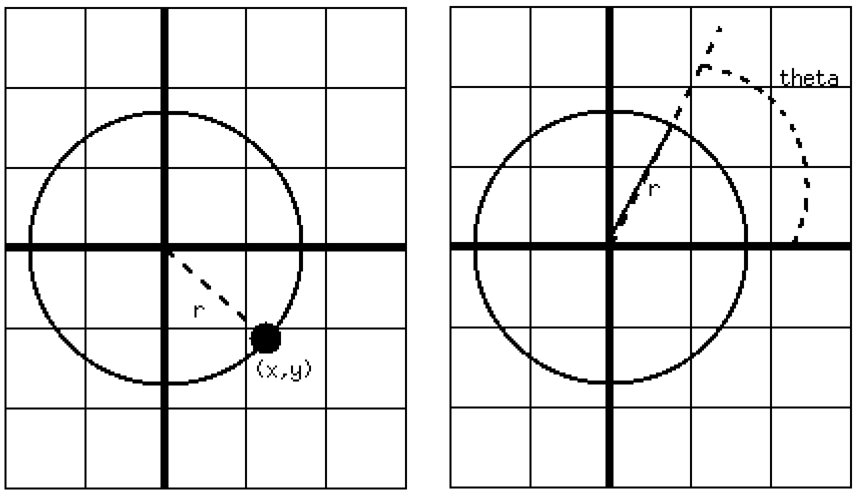
(a) (b) (c) (d)

Figura 3-8. Intersecția unui con cu un plan:(a) cerc; (b) elipsă; (c) parabolă; (d) hiperbolă.

3.6. Cercul

Ecuația carteziană a unui cerc care trece prin originea sistemului de coordonate este (figura 3-9a):

$$\text{(3-10)}\quad x^2 + y^2 = r^2 \Leftrightarrow y = \pm\sqrt{r^2 - x^2}, unde - r \le x \le r \,.$$



(a) (b)

Figura 3-9. Descrierea cercului cu centrul în originea unui sistem de coordonate: (a) carteziene; (b) polare.

Într-un sistem de coordonate polare 2D, un cerc având centrul în originea sistemului de coordonate (figura 3-9b) este descris de ecuațiile:

$$\text{(3-11)}\quad \begin{cases} r = x \\ 0 \le \theta \le 2\pi(radiani) \end{cases}.$$

Ecuația tangentei la un cerc C într-un punct dat $M_1(x_1,y_1)$ se determină folosind derivata ecuației cercului. Fie cercul C: $(x - x_0)^2 + (y - y_0)^2 = r^2$ cu centrul $M_0(x_0,y_0)$ şi raza r, iar punctul $M_1(x_1,y_1)$ un punct dat pe cercul C. Vom face substituirile (dedublări): $x^2 \to xx_1$, $x \to ½*(x+x_1)$, $y^2 \to yy_1$, $y \to ½*(y+y_1)$, $xy \to ½*(xy_1+x_1y)$.

Înlocuind în ecuația cercului, vom obține:

$x^2-2xx_0+x_0^2+y^2-2yy_0+y_0^2=r^2$

$\Leftrightarrow xx_1-2*½*(x+x_1)x_0+x_0^2+yy_1-2*½*(y+y_1)y_0+y_0^2=r^2$

$\Leftrightarrow xx_1-xx_0-x_1x_0+x_0^2+yy_1-yy_0-y_1y_0+y_0^2=r^2$

$\Leftrightarrow x(x_1-x_0)-x_0(x_1-x_0)+y(y_1-y_0)-y_0(y_1-y_0)=r^2$

$\Leftrightarrow (x_1-x_0)(x-x_0)+(y_1-y_0)(y-y_0)=r^2$.

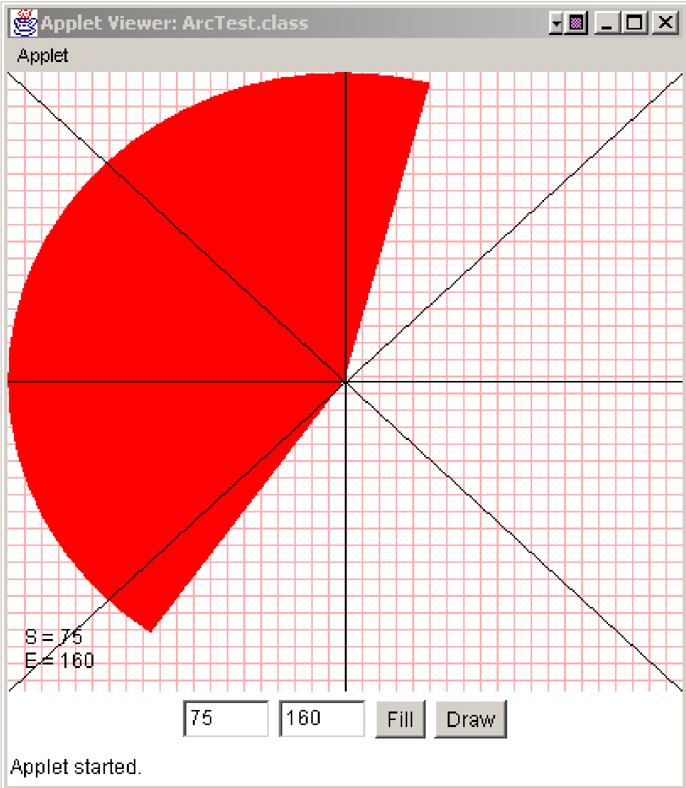


Figura 3-10. Trasarea sectoarelor de cerc în coordonate carteziene.

Pentru exemplificare, vom face apel la unul din applet-urile demo propuse de firma Sun în documentația jdk1.3.1 (v. http://www.java.sun). Rezultatul rulării applet-ului poate fi urmărit în figura 3-10.

Applet 3-1. ArcTest.java: trasarea sectoarelor de cerc în coordonate carteziene.

```
/* ____________________________________________________
* @(#)ArcTest.java  1.9 95/09/01 Sami Shaio
* Copyright (c) 1994-1996 Sun Microsystems, Inc. All *
Rights Reserved.
* Modificat si adaptat: 2001 Dan Neagu
* ______________________________________________________/

import java.awt.*;
import java.applet.*;
```

```
public class ArcTest extends Applet {
    ArcControls controls;
    public void init() {
     setLayout(new BorderLayout());
     ArcCanvas c = new ArcCanvas();
     add("Center", c);
     add("South", controls = new ArcControls(c));
    }

    public void start() {
     controls.enable();
    }

    public void stop() {
     controls.disable();
    }

    public boolean handleEvent(Event e) {
     if (e.id == Event.WINDOW_DESTROY) {
         System.exit(0);
     }
     return false;
    }

    public static void main(String args[]) {
     Frame f = new Frame("ArcTest");
     ArcTest   arcTest = new ArcTest();

     arcTest.init();
     arcTest.start();

     f.add("Center", arcTest);
     f.resize(300, 300);
     f.show();
    }
}

class ArcCanvas extends Canvas {
    int        startAngle = 0;
    int        endAngle = 45;
    boolean    filled = false;
    Font  font;

    public void paint(Graphics g) {
     Rectangle r = bounds();
     int hlines = r.height / 10;
     int vlines = r.width / 10;
     g.setColor(Color.pink);
     for (int i = 1; i <= hlines; i++) {
         g.drawLine(0, i * 10, r.width, i * 10);
     }
```

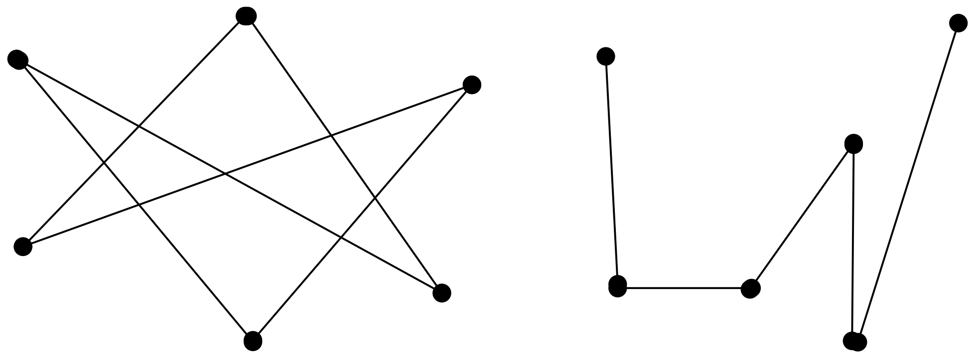
```
     for (int i = 1; i <= vlines; i++) {
         g.drawLine(i * 10, 0, i * 10, r.height);
     }

     g.setColor(Color.red);
     if (filled) {
         g.fillArc(0, 0, r.width - 1, r.height - 1,
startAngle, endAngle);
     } else {
         g.drawArc(0, 0, r.width - 1, r.height - 1,
startAngle, endAngle);
     }

     g.setColor(Color.black);
     g.setFont(font);
     g.drawLine(0, r.height / 2, r.width, r.height / 2);
     g.drawLine(r.width / 2, 0, r.width / 2, r.height);
     g.drawLine(0, 0, r.width, r.height);
     g.drawLine(r.width, 0, 0, r.height);
     int sx = 10;
     int sy = r.height - 28;
     g.drawString("S = " + startAngle, sx, sy);
     g.drawString("E = " + endAngle, sx, sy + 14);
    }

   public void redraw(boolean filled, int start, int end){
     this.filled = filled;
     this.startAngle = start;
     this.endAngle = end;
     repaint();
    }
}

class ArcControls extends Panel {
    TextField s;
    TextField e;
    ArcCanvas canvas;

    public ArcControls(ArcCanvas canvas) {
     this.canvas = canvas;
     add(s = new TextField("0", 4));
     add(e = new TextField("45", 4));
     add(new Button("Fill"));
     add(new Button("Draw"));
    }

    public boolean action(Event ev, Object arg) {
     if (ev.target instanceof Button) {
         String label = (String)arg;
         canvas.redraw(label.equals("Fill"),
               Integer.parseInt(s.getText().trim()),
```

```
                Integer.parseInt(e.getText().trim()));
            return true;
        }
        return false;
      }
}
```

3.7. Poligoane

O **polilinie** reprezintă o secvență de linii (muchii) ce leagă între ele o secvență de puncte (vârfuri). O *polilinie* este *închisă* (figura 3-11a) dacă cele două capete ale sale coincid. O *polilinie este simplă* (figura 3-11b) daca nu se întretaie (spre exemplu, două vârfuri coincid, un vârf al poliniei se află în interiorul unei muchii distincte sau două muchii se intersectează). O polilinie în plan poate fi reprezentată ca o simplă serie de coordonate (x,y) ale vârfurilor sale.

Modalitatea de redare a proprietăților grafice ale poliliniei (rendering) este determinată de un set de proprietăți numite **atribute grafice**. Acestea includ culoarea liniei, lățimea liniei, stilul liniei (solidă, punctată, întreruptă), modalitatea de unire a segmentelor consecutive (rotunjită, dură) etc.



(a) (b)

Figura 3-11. Polilinii: (a) polilinie inchisă; (b) polilinie simplă.

Multe sisteme grafice furnizează cazuri speciale de curbe: cercuri, elipse, suprafețe circulare, curbe Bezier sau B-spline. Vom considera curbele ca o generalizare a poliliniilor. Unul dintre argumente este acela că, practic, multe sisteme grafice de desenare realizează curbele prin algoritmi de aproximare printr-un număr cât mai mare de polilinii cât mai mici.

Un **poligon** este o figură plană compusă din vârfuri şi muchii cu condiția ca seria segmentelor (muchiilor) sale să formeze o polilinie închisă plană.

3.7.1. Convexitatea şi concavitatea poligoanelor

Un poligon este numit convex dacă pentru orice puncte P_1, P_2 aflate în interiorul poligonului, toate punctele de pe segmentul $[P_1P_2]$ aparțin interiorului poligonului. Un poligon este concav dacă este neconvex, deci un poligon pentru care există două puncte P_1, P_2 din interiorul poligonului cu proprietatea că segmentul $[P_1P_2]$ are puncte ce nu aparțin interiorului poligonului. Un poligon ce se autointersectează este un poligon concav pentru care cel puțin două laturi se intersectează într-un punct, altul decât vârfurile lor.

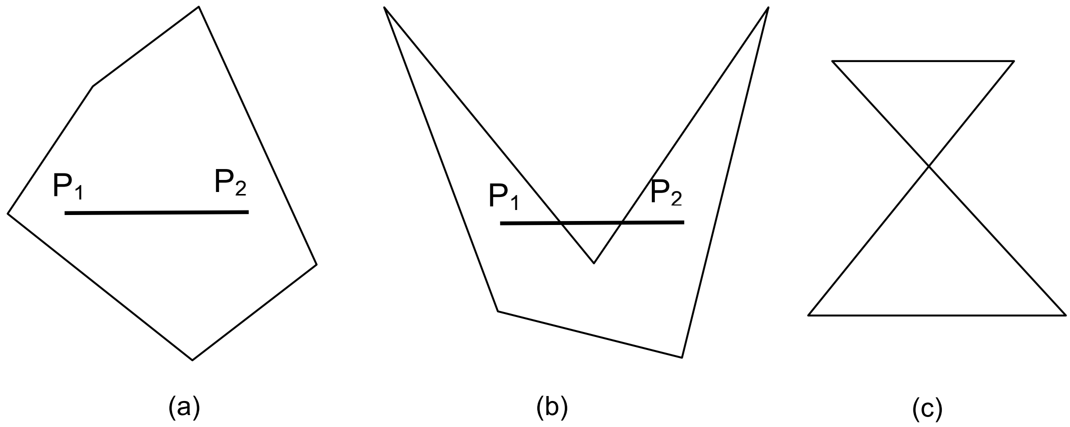


Figura 3-12. Tipuri de poligon: (a) convex; (b) concav; (c) poligon ce se autointersectează.

Gradul unui poligon este dat de numărul laturilor (poligoanele din figurile 3-12a,b sunt de grad cinci, numite şi pentagoane, iar cel din figura 3-12c este de grad patru). Poligonul din figura 3-12a este convex, în timp ce poligoanele din figurile 3-12b,c sunt concave (există două puncte aparținând interiorului poligonului, astfel încât segmentul având capete cele două puncte să nu aparțină în totalitate interiorului poligonului, v. segmentul $[P_1P_2]$).

3.8. Plane

Un **plan** este o suprafață continuă extinsă pe două direcții. Planul este definit de trei puncte necoliniare sau de un punct şi un vector (paralel cu planul sau perpendicular pe acesta).

Ecuația generală a planului este dată de:

$$\text{(3-12)}\ \ Ax + By + Cz = 0 \ \text{ sau}$$

$$\text{(3-13)}\ \ A'x + B'y + C'z + D' = 0$$

$$\text{(3-14)}\ \ A' = \frac{A}{d}, B' = \frac{B}{d}, C' = \frac{C}{d}, D' = \frac{D}{d}, d = \sqrt{A^2 + B^2 + C^2}$$

3.8.1. Algoritm de caracterizare a unui vector normal la plan

Fie P_1, P_2, P_3 trei puncte date. Un **vector normal la plan** este caracterizat de relația:

$$\text{(3-15)}\ N=[P_2-P_1]\times[P_3-P_1]$$

unde x este produsul vectorial.

Exemplu: dat fiind planul: $Ax + By + Cz + D = 0$, vectorul [A B C] este normal la plan. Algoritmul pentru determinarea ecuației unui plan caracterizat prin trei puncte distincte necoliniare: P_1, P_2, P_3.

#1. Se determină N = [A B C];

#2. Se determină $D=-N\bullet P_i$, i =1,2,3, unde • este produsul scalar

#3. Se scrie ecuația planului folosind relația (3-13) şi coeficienții A, B, C, D.