

Método Main:

```
1 package ProyectoFiguras_U4;
2 import ProyectoFiguras.*;
3 public class ApFiguras {
4     public static void main(String[] args) {
5         int n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14;
6         n1= (int) (Math.random()*20+1);
7         n2= (int) (Math.random()*20+1);
8         n3= (int) (Math.random()*20+1);
9         n4= (int) (Math.random()*20+1);
10        n5= (int) (Math.random()*20+1);
11        n6= (int) (Math.random()*20+1);
12        n7= (int) (Math.random()*20+1);
13        n8= (int) (Math.random()*20+1);
14        n9= (int) (Math.random()*10+1);
15        n10=(int) (Math.random()*20+1);
16        n11=(int) (Math.random()*20+1);
17        n12=(int) (Math.random()*20+1);
18        n13=(int) (Math.random()*20+1);
19        n14=(int) (Math.random()*20+1);
20        //Objetos de figuras 2D
21        CCuadrado A = new CCuadrado();
22        CCirculo B = new CCirculo();
23        CRombo C = new CRombo();
24        CRectangulo D = new CRectangulo();
25        CTrapecioRec E = new CTrapecioRec();
26        CTrapecioIsos F = new CTrapecioIsos();
27        CTrianguloEq H = new CTrianguloEq();
28        CTrianguloIsos I = new CTrianguloIsos();
29        CTrianguloEsc J = new CTrianguloEsc();
30        if (n3<=n4) {
31            while (n3>n4){
32                n3=(int) (Math.random()*20+1);
```

```
32        n3=(int) (Math.random()*20+1);
33    }
34    }
35    if (n5==n6) {
36        while (n5!=n6){
37            n5=(int) (Math.random()*20+1);
38        }
39    }
40    if (n7<=n8) {
41        while (n7>n8){
42            n7=(int) (Math.random()*20+1);
43        }
44    }
45    if (n10>=n11) {
46        while (n11>n10){
47            n11=(int) (Math.random()*20+1);
48        }
49    }
50    if ((n12+n13)<=n14) {
51        while ((n12+n13)>n14){
52            n12=(int) (Math.random()*20+1);
53        }
54    }else if ((n13+n14)<=n12) {
55        while ((n13+n14)>n12){
56            n13=(int) (Math.random()*20+1);
57        }
58    }else if ((n12+n14)<=n13) {
59        while ((n12+n14)>n13){
60            n14=(int) (Math.random()*20+1);
61        }
62    }
63    A.setLado1(n1);
```

```
63    A.setLado1(n1);
64    A.Area();
65    A.Perimetro();
66    System.out.println("Cuadrado:");
67    System.out.println(A);
68    B.setLado1(n2);
69    B.Area();
70    B.Perimetro();
71    System.out.println("\nCirculo");
72    System.out.println(B);
73    C.setLado1(n3);
74    C.setLado2(n4);
75    C.Area();
76    C.Perimetro();
77    System.out.println("\nRombo");
78    System.out.println(C);
79    D.setLado1(n5);
80    D.setLado2(n6);
81    D.Area();
82    D.Perimetro();
83    System.out.println("\nRectangulo");
84    System.out.println(D);
85    E.setLado1(n7);
86    E.setLado2(n8);
87    E.setLado3(n9);
88    E.Area();
89    E.Perimetro();
90    System.out.println("\nTrapecio Rectangulo");
91    System.out.println(E);
92    F.setLado1(n7+5);
93    F.setLado2(n8+5);
94    F.setLado3(n9);
```

```
Source History
87 E.setLado3(n9);
88 E.Area();
89 E.Perimetro();
90 System.out.println("\nTrapecio Rectangulo");
91 System.out.println(E);
92 F.setLado1(n7+5);
93 F.setLado2(n8+5);
94 F.setLado3(n9);
95 F.Area();
96 F.Perimetro();
97 System.out.println("\nTrapecio Isosceles");
98 System.out.println(F);
99 H.setLado1(n1);
100 H.Area();
101 H.Perimetro();
102 System.out.println("\nTriangulo Equilatero");
103 System.out.println(H);
104 I.setLado1(n10);
105 I.setLado2(n11);
106 I.Area();
107 I.Perimetro();
108 System.out.println("\nTriangulo Isosceles");
109 System.out.println(I);
110 J.setLado1(n12);
111 J.setLado2(n13);
112 J.setLado3(n14);
113 J.Perimetro();
114 J.Area();
115 System.out.println("\nTriangulo Escaleno");
116 System.out.println(J);
117 }
118 }
```

```
Output - POO-2022 (run) X
run:
Cuadrado:
Lado= 6.0
Área= 36.0u²
Perimetro= 24.0u

Circulo
Radio= 3.0
Área= 28.274333882308138u²
Perimetro= 18.84955592153876u

Rombo
Diagonal mayor= 13.0
Diagonal menor= 12.0
Área= 78.0u²
Perimetro= 35.30361202590826u

Rectangulo
Base= 15.0
Altura= 12.0
Área= 228.0u²
Perimetro= 62.0u

Trapecio Rectangulo
Base mayor= 4.0
Base menor= 18.0
Altura= 7.0
Área= 77.0u²
Perimetro= 44.65247584249853u

Trapecio Isosceles
Base mayor= 9.0
Base menor= 23.0
```

```
Output - POO-2022 (run) X
Trapecio Rectangulo
Base mayor= 4.0
Base menor= 18.0
Altura= 7.0
Área= 77.0u²
Perimetro= 44.65247584249853u

Trapecio Isosceles
Base mayor= 9.0
Base menor= 23.0
Altura= 7.0
Área= 112.0u²
Perimetro= 51.798989873223334u

Triangulo Equilatero
Lado= 6.0
Área= 15.588457260119896u²
Perimetro= 18.0u

Triangulo Isosceles
Lado 1= 17.0
Lado 2= 14.0
Área= 94.5565307104697u²
Perimetro= 45.0u

Triangulo Escaleno
Lado 1= 16.0
Lado 2= 17.0
Lado 3= 17.0
Área= 120.0u²
Perimetro= 50.0u
BUILD SUCCESSFUL (total time: 1 second)
```

Interface:

```
Source History
1 package ProyectoFiguras_U4;
2 public interface Interface_Figuras {
3     public void Area();
4     public void Perimetro();
5 }
6
```

Clase Base por medio de la implementacion:

```
Source History
1 package ProyectoFiguras_U4;
2 import ProyectoFiguras.*;
3 public abstract class CFiguras2D implements Interface_Figuras{
4     protected double l1,l2,l3,perimetro,area;
5
6     public CFiguras2D() {}
7     public CFiguras2D(double l1){setLado1(l1);}
8     public CFiguras2D(double l1, double l2){
9         setLado1(l1);
10        setLado2(l2);
11    }
12    public CFiguras2D(double l1, double l2, double l3){
13        setLado1(l1);
14        setLado2(l2);
15        setLado3(l3);
16    }
17    public void setLado1(double l1){this.l1=l1;}
18    public void setLado2(double l2){this.l2=l2;}
19    public void setLado3(double l3){this.l3=l3;}
20    public double getArea(){return area;}
21    public double getPerimetro(){return perimetro;}
22    public String toString(){
23        return "\nArea= "+area+"u"\nPerimetro= "+perimetro+"u";
24    }
25 }
26
```

Clase Base por medio de métodos abstractos:

```
Source History
1 package ProyectoFiguras;
2 public abstract class CFiguras2D {
3     protected double l1,l2,l3,perimetro,area;
4
5     public CFiguras2D() {}
6     public CFiguras2D(double l1){setLado1(l1);}
7     public CFiguras2D(double l1, double l2){
8         setLado1(l1);
9         setLado2(l2);
10    }
11    public CFiguras2D(double l1, double l2, double l3){
12        setLado1(l1);
13        setLado2(l2);
14        setLado3(l3);
15    }
16    public void setLado1(double l1){this.l1=l1;}
17    public void setLado2(double l2){this.l2=l2;}
18    public void setLado3(double l3){this.l3=l3;}
19    public double getArea(){return area;}
20    public double getPerimetro(){return perimetro;}
21    public abstract void Area();
22    public abstract void Perimetro();
23
24    @Override
25    public String toString(){
26        return "\nArea= "+area+"u"\nPerimetro= "+perimetro+"u";
27    }
28 }
```

Clase Cuadrado:

```
Source History
1 package ProyectoFiguras;
2 public class CCuadrado extends CFiguras2D {
3     public CCuadrado() {}
4     public CCuadrado(double l1){super(l1);}
5     @Override
6     public void Area() {
7         area=Math.pow(l1, 2);
8     }
9     @Override
10    public void Perimetro() {
11        perimetro=l1*4;
12    }
13    @Override
14    public String toString(){
15        return "Lado= "+l1+super.toString();
16    }
17 }
18
```

Clase Circulo:

```
Source History
1 package ProyectoFiguras;
2 public class CCirculo extends CFiguras2D {
3     public CCirculo() {}
4     public CCirculo(double l1){super(l1);}
5     @Override
6     public void Area() {
7         area= Math.PI*Math.pow(l1, 2);
8     }
9     @Override
10    public void Perimetro() {
11        perimetro = Math.PI*2*l1;
12    }
13    @Override
14    public String toString(){
15        return "Radio= "+l1+super.toString();
16    }
17 }
18
```

Clase Rombo:

```
Source History
1 package ProyectoFiguras;
2 public class CRombo extends CFiguras2D {
3     public CRombo() {}
4     public CRombo(double l1,double l2){
5         super(l1,l2);
6     }
7     @Override
8     public void Area() {
9         area= (l1*l2)/2;
10    }
11    @Override
12    public void Perimetro() {
13        perimetro= Math.sqrt(Math.pow(l1/2, 2)+Math.pow(l2/2, 2))*4;
14    }
15    public String toString(){
16        return "Diagonal mayor= "+l1+"\nDiagonal menor= "+l2+super.toString();
17    }
18 }
19
```

Clase Rectángulo:

```
Source History
1 package ProyectoFiguras;
2 public class CRectangulo extends CFiguras2D{
3     public CRectangulo() {}
4     public CRectangulo(double l1, double l2){
5         super(l1,l2);
6     }
7     @Override
8     public void Area() {
9         area= l1*l2;
10    }
11    @Override
12    public void Perimetro() {
13        perimetro= 2*(l1+l2);
14    }
15    @Override
16    public String toString(){
17        return "Base= "+l1+"\naltura= "+l2+super.toString();
18    }
19 }
20
```

Clase Trapecio:

```
Source History
1 package ProyectoFiguras;
2 public abstract class CTrapecio extends CFiguras2D {
3     public CTrapecio() {}
4     public CTrapecio(double l1, double l2, double l3) {
5         super(l1,l2,l3);
6     }
7 }
8
```

Clase Trapecio Rectángulo:

```
Source History
1 package ProyectoFiguras;
2 public class CTrapecioRec extends CTrapecio {
3     public CTrapecioRec() {}
4     public CTrapecioRec(double l1,double l2,double l3) {
5         super(l1,l2,l3);
6     }
7     @Override
8     public void Area() {
9         area=(l1+l2)*l3/2;
10    }
11    @Override
12    public void Perimetro() {
13        perimetro=l1+l2+l3+Math.sqrt(Math.pow(l1-l2, 2)+Math.pow(l3, 2));
14    }
15    @Override
16    public String toString() {
17        return "Base mayor= "+l1+"\nBase menor= "+l2+"\nAltura= "+l3+super.toString();
18    }
19 }
20
```

Clase Trapecio Isósceles:

```
Source History
1 package ProyectoFiguras;
2 public class CTrapecioIsos extends CTrapecio {
3     public CTrapecioIsos() {}
4     public CTrapecioIsos(double l1, double l2, double l3) {
5         super(l1,l2,l3);
6     }
7     @Override
8     public void Area() {
9         area= (l1+l2)*l3/2;
10    }
11    @Override
12    public void Perimetro() {
13        perimetro= l1+l2+2*(Math.sqrt(Math.pow((l1-l2)/2, 2)+Math.pow(l3, 2)));
14    }
15    @Override
16    public String toString() {
17        return "Base mayor= "+l1+"\nBase menor= "+l2+"\nAltura= "+l3+super.toString();
18    }
19 }
20
```

Clase Triangulo:

```
Source History
1 package ProyectoFiguras;
2 public abstract class CTriangulo extends CFiguras2D{
3     public CTriangulo() {}
4     public CTriangulo(double l1){
5         super(l1);
6     }
7     public CTriangulo(double l1, double l2){
8         super(l1,l2);
9     }
10    public CTriangulo(double l1, double l2, double l3){
11        super(l1,l2,l3);
12    }
13 }
14
```

Clase Triangulo Equilátero:

```
Source History
1 package ProyectoFiguras;
2 public class CTrianguloEq extends CTriangulo{
3     public CTrianguloEq() {}
4     public CTrianguloEq(double l1){
5         super(l1);
6     }
7     @Override
8     public void Area() {
9         area= l1*Math.sqrt(Math.pow(l1, 2)-Math.pow(l1/2, 2))/2;
10    }
11    @Override
12    public void Perimetro(){
13        perimetro=l1*3;
14    }
15    @Override
16    public String toString(){
17        return "Lado= "+l1+super.toString();
18    }
19 }
20
```

Clase Triangulo Isósceles:

```
Source History
1 package ProyectoFiguras;
2 public class CTrianguloIsos extends CTriangulo{
3     public CTrianguloIsos() {}
4     public CTrianguloIsos(double l1, double l2){
5         super(l1,l2);
6     }
7     @Override
8     public void Area() {
9         area=l1*Math.sqrt(Math.pow(l2, 2)-Math.pow(l1/2, 2))/2;
10    }
11    @Override
12    public void Perimetro(){
13        perimetro=l1+l2*2;
14    }
15    @Override
16    public String toString(){
17        return "Lado 1= "+l1+"\nLado 2= "+l2+super.toString();
18    }
19 }
20
```

Clase Triangulo Escaleno:

```
Source History
1 package ProyectoFiguras;
2 public class CTrianguloEsc extends CTriangulo{
3     public CTrianguloEsc() {}
4     public CTrianguloEsc(double l1, double l2, double l3){
5         super(l1,l2,l3);
6     }
7     @Override
8     public void Perimetro(){
9         perimetro=l1+l2+l3;
10    }
11    @Override
12    public void Area() {
13        area=Math.sqrt((perimetro/2)*((perimetro/2)-l1)*((perimetro/2)-l2)*((perimetro/2)-l3));
14    }
15    @Override
16    public String toString(){
17        return "Lado 1= "+l1+"\nLado 2= "+l2+"\nLado 3= "+l3+super.toString();
18    }
19 }
20
```