

Preparado por: Luis Daniel Benavides Navarro
email: dnielben@gmail.com

Ejercicio adaptado del libro "Testing Object Oriented Systems" por Binder (2000).

Problema:

1. Caja Negra: Usted es un Ingeniero de pruebas y se le pide considerar la siguiente especificación: El programa recibe tres enteros que son considerados como las longitudes de los lados de un triángulo. El programa imprime un mensaje que indica si el triángulo es escaleno, isósceles o equilátero. En una hoja de papel escriba casos de prueba especificando la descripción (1 frase), datos de entrada y datos de salida. Los caso de prueba deben probar completamente el programa.
2. Caja Blanca: Ahora considere que parte de la especificación de código y el diagrama de clases de una implementación del programa hecha en Java le es suministrada. ¿Qué otros casos de prueba puede sugerir? Escriba solamente la descripción del caso de prueba. Recuerde escriba la cantidad suficiente de casos de prueba para probar el software bajo estudio.

Ayuda 1:

Un triángulo válido debe cumplir dos condiciones. Ningún lado debe tener una longitud de cero y cada lado debe ser más corto que la suma de todos los lados divididos por dos. Es decir:

En un triángulo con lados a, b, c

Si $s = (a + b + c) / 2$

Entonces $s > a$, $s > b$ y $s > c$ siempre es verdadero

Un triángulo con todos los lados iguales es un triángulo equilátero. Un triángulo con dos lados iguales es un triángulo isósceles y un uno con todos los lados de diferente longitud es un triángulo escaleno.

Las figuras abajo muestran la implementación de algunas clases de la jerarquía de clases y el diagrama de clases usado en la implementación.

```

/* Java fragments of the figure hierarchy */

class Polygon extends Closed{
    abstract void draw(int r, int g, int b);/* Color closed area */
    abstract void erase();                /* Set background rgb */
    abstract float area();                /* Return area */
    abstract float perimeter();           /* Return sum of sides */
    abstract void center(int x, int y); /* Return centroid pixel */
}

/* Method implementations not shown */

class Triangle extends Polygon{
    public Triangle(LineSegment a, LineSegment b, LineSegment c){/*ctor*/}
    public void setA(LineSegment a){/* Change side a */}
    public void setB(LineSegment b){/* Change side a */}
    public void setC(LineSegment c){/* Change side a */}

    public LineSegment getA()    {/* Get side a */}
    public LineSegment getB()    {/* Get side a */}
    public LineSegment getC()    {/* Get side a */}

    public boolean is_isosceles() {/* Returns true if Isosceles*/}
    public boolean is_scalene()   {/* Returns true if Scalene*/}
    public boolean is_equilateral() {/* Returns true if Equilateral*/}

    public void draw(int r, int g, int b);/* Color closed area */
    public void erase();                /* Set background rgb */
    public float area();                /* Return area */
    public float perimeter();           /* Return sum of sides */
    public void center(int x, int y); /* Return centroid pixel */
}

class LineSegment extends Open{
    public LineSegment(int x1, int y1, int x2, int y2){/*ctor*/}
    public void setx1(int x1)    {/* Change x1*/}
    public void sety1(int y1)    {/* Change y1*/}
    public void setx2(int x2)    {/* Change x2*/}
    public void sety2(int y2)    {/* Change y2*/}

    public int getx1()           {/* get x1*/}
    public int gety1()           {/* get y1*/}
    public int getx2()           {/* get x2*/}
    public int gety2()           {/* get y2*/}
}

```

Listado de las clases Polygon, Triangle y Line Segments

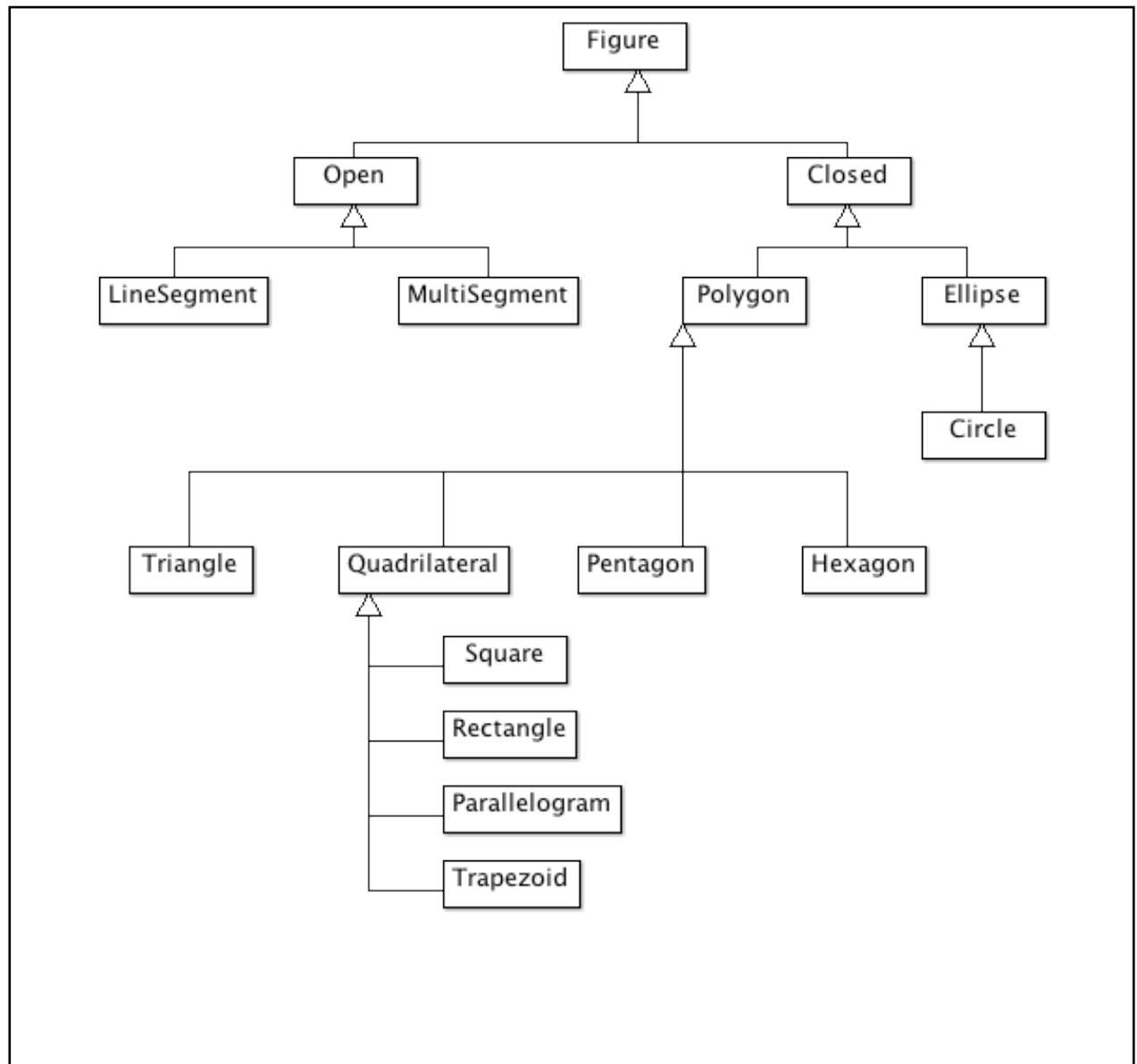


Diagrama de clases de la implementación.