

Tarea 6

Herencia

Ejercicio 6.1. Triángulos

- Construye las clases `Triangulo`, `Equilatero`, `Isosceles` y `Escaleno` de tal manera que podamos construir cada tipo de triangulo como se muestra en la clase `Test`.

Hints

- `Triangulo` es super clase y contiene los métodos necesarios para construir cualquier tipo de triángulo (de los mencionados).
- Todos los triángulos tienen 3 lados, esto es una generalidad.
- Usa el constructor adecuado en cada clase derivada para construir el objeto.

```

public class Test {

    public static void main(String[] args) {
        Equilatero equi = new Equilatero(2); // Se construye con un lado
        double area = equi.calcularArea();
        double peri = equi.calcularPerimetro();
        System.out.println("--- EQUILATERO ---");
        System.out.println("area = " + area);
        System.out.println("perimetro = " + peri);
        equi.imprimirLados();

        Isosceles iso = new Isosceles(2, 5); // Se construye con dos lados
        area = iso.calcularArea();
        peri = iso.calcularPerimetro();
        System.out.println("--- ISOSCELES ---");
        System.out.println("area = " + area);
        System.out.println("perimetro = " + peri);
        iso.imprimirLados();

        Escaleno esca = new Escaleno(3, 4, 5); // Se construye con 3 lados.
        area = esca.calcularArea();
        peri = esca.calcularPerimetro();
        System.out.println("--- ESCALENO ---");
        System.out.println("area = " + area);
        System.out.println("perimetro = " + peri);
        esca.imprimirLados();
    }
}

```

Salida:

```

run:
--- EQUILATERO ---
area = 1.7320508075688772
perimetro = 6.0
Lados [2.00, 2.00, 2.00]
--- ISOSCELES ---
area = 4.898979485566356
perimetro = 12.0
Lados [2.00, 5.00, 5.00]
--- ESCALENO ---
area = 6.0
perimetro = 12.0
Lados [3.00, 4.00, 5.00]
BUILD SUCCESSFUL (total time: 0 seconds)

```

Ejercicio 6.2. Matriz

- Escriba una clase que construya una matriz cuadrada de $N \times N$, donde el valor inicial de los elementos empiece en 0 y termine en $N^2 - 1$.
 - El valor N lo ingresará el usuario por teclado al inicial el programa.
 - Se deberá mostrar en pantalla la matriz que se acaba de crear al utilizar el método

`imprimirMatriz()`

- **Extienda** la funcionalidad de la clase anterior (no la modifique) y agregue la funcionalidad para que el usuario, además de ingresar N , pueda ingresar la posición de un elemento de la matriz y sustituirlo por un valor que él ingrese.
 - Se deberá poder cambiar un valor al usar el método

`sustituirElemento(int i, int j, int valor)`

- Crea una clase Test para cada clase que pruebe su funcionalidad correctamente.