

Casting de tipos básicos

Una carpintería encarga realizar una calculadora. El programa tiene una única función que calcula el área de un tablero, introduciendo las medidas de largo y ancho de dicho tablero.

Tenemos las siguientes restricciones:

1. Los valores se dan en metros.
2. Los valores suministrados por la fábrica son números fraccionarios, positivos y mayores que cero.
3. Los valores que maneja la carpintería son enteros positivos y mayores que cero.

Las tareas por realizar son las siguientes:

- Crear un proyecto llamado “CastingBasico”.
- Crear la clase llamada TableroFabrica:
 - Atributos: largo y ancho.
 - Métodos:
 - Constructores: por defecto y parametrizado (con los dos valores).
 - Getters/setters.
 - Area: devuelve el valor calculado del área del tablero.

```
public class TableroFabrica {  
  
    protected double largo, ancho;  
  
    protected TableroFabrica() {  
        this.largo = 0.1;  
        this.ancho = 0.1;  
    }  
  
    protected TableroFabrica(double largo, double ancho) {  
        this.largo = valor(largo);  
        this.ancho = valor(ancho);  
    }  
  
    protected double getLargo() {  
        return largo;  
    }  
  
    protected void setLargo(double largo) {  
        this.largo = valor(largo);  
    }  
  
    protected double getAncho() {  
        return ancho;  
    }  
  
    protected void setAncho(double ancho) {  
        this.ancho = valor(ancho);  
    }  
  
    private double valor(double metros) {  
        return (metros > 0) ? metros : 0.1;  
    }  
  
    protected double area() {  
        return getLargo() * getAncho();  
    }  
  
}
```

- Crear la clase llamada TableroCarpinteria:
 - Atributos: largo y ancho.

- Métodos:

- Constructores: por defecto y parametrizado (con los dos valores).
- Getters/setters.
- Area: devuelve el valor calculado del área del tablero.

```
public class TableroCarpinteria {  
  
    protected int largo, ancho;  
  
    protected TableroCarpinteria() {  
        this.largo = 1;  
        this.ancho = 1;  
    }  
  
    protected TableroCarpinteria(int largo, int ancho) {  
        this.largo = valor(largo);  
        this.ancho = valor(ancho);  
    }  
  
    protected int getLargo() {  
        return largo;  
    }  
  
    protected void setLargo(int largo) {  
        this.largo = valor(largo);  
    }  
  
    protected int getAncho() {  
        return ancho;  
    }  
  
    protected void setAncho(int ancho) {  
        this.ancho = valor(ancho);  
    }  
  
    private int valor(int metros) {  
        return (metros > 0) ? metros : 1;  
    }  
  
    protected int area() {  
        return getLargo() * getAncho();  
    }  
  
}
```

- Crear la clase principal llamada Flow:

- Incluir el método "main":

- Crear un objeto de cada tipo con los mismos valores (largo: 1,8 m y ancho: 0,5 m).
 - Mostrar el área de cada tipo.
 - Mostrar la diferencia entre las distintas áreas.

- Crear un método llamado "diferencia" que devuelva la diferencia de área entre ambos tipos de tableros.

```
public class Flow {  
  
    public static void main(String[] args) {  
  
        final double LARGO = 1.8, ANCHO = 0.5;  
  
        TableroFabrica tablaFabrica = new TableroFabrica(LARGO, ANCHO);  
        // TableroCarpinteria tablaCarpinteria = new TableroCarpinteria((int)LARGO, (int)ANCHO);  
        TableroCarpinteria tablaCarpinteria = new TableroCarpinteria((int)Math.round(LARGO), (int)Math.round(ANCHO));  
  
        System.out.println("La tabla de fabrica: " + tablaFabrica.area() + " m²");  
        System.out.println("La tabla de carpinteria: " + tablaCarpinteria.area() + " m²");  
  
        System.out.println("La diferencia entre tablas es de " + diferencia(tablaFabrica.area(), (double)tablaCarpinteria.area()) + " m²");  
    }  
}
```

```
    }  
  
    private static double diferencia(double areaFabrica, double areaCarpinteria) {  
        return Math.max(areaFabrica, areaCarpinteria) - Math.min(areaFabrica, areaCarpinteria);  
    }  
  
}
```

¿Cómo funciona?

Presenta un problema de funcionamiento debido a que al realizar la conversión de variables enteras, estas se redondean hacia abajo, lo que resulta en la asignación de un valor de cero a algunas de ellas.

¿Qué solución propones para corregir las limitaciones del negocio?

He solucionado el problema aplicando un redondeo hacia arriba (`Math.round()`) en las variables enteras antes de enviarlas al constructor de la clase `TableroCarpinteria`.