Nombre y Apellido: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Cantidad de hojas entregadas: \_\_ Nota:\_\_\_\_\_\_\_\_

|  |
| --- |
| **Lea atentamente TODO el enunciado antes de empezar a desarrollar, recodar generar todas las firmas de las funciones y las especificaciones correspondientes para que el proyecto compile antes de programar la solución** |

1) La clase AreaOperations, es una clase especialmente pensada para hacer operaciones de area con figuras geometricas en 2D. Para escribir esta clase solo se necesita un método que funcione de manera polimorfica, es decir, que no necesite conocer exactamente sobre que instancia esta actuando, y que funcione para cualquier objeto que implemente a Figura. (5ptos)

a) Escriba el método: public static double sumarNAreas(ArrayList<Figura> lista). Que calcula el area de todas las figuras y la suma, para luego devolver el resultado final como un numero flotante (El metodo suma las areas independientemente de la figura que se trate).

Pista: En cada elemento de la lista se encuentra un objeto que implementa a Figura. Pero al no ser una clase, nada es instancia de Figura

b) Crear las clases Circulo(radio), Rectangulo(base,altura) y Cuadrado(lado). Donde en su constructor reciben como double, los valores indicados entre parentesis, todos deben implementar a la interfaz Figura, con un único método, double area(). Que permite el polimorfismo del metodo implementado en a)

IMPORTANTE: Los atributos de las figuras deben llamarse exactamente como estan entre parentesis y deben ser declaradas como **public double**

c) Crear la clase LineaRecta que implementa a Figura, pero que al calcular el area lance la excepción NoTieneAreaException

2) Utilizando el TP Electrodomestico realizando en la actividad 3. Se sabe que la tabla de consumos sufrió una pequeña modificación

**Consumos permitidos según el voltaje**

|  |  |  |  |
| --- | --- | --- | --- |
| Consumo | 110 V | 220 V | 240 V |
| A | ✔ | ✔ | ✔ |
| B | ✔ | ✔ | ✔ |
| C | ✔ | ✔ | ✔ |
| D | X | ✔ | ✔ |
| E | X | ✔ | ✔ |
| F | X | X | ✔ |

El cambio es que ahora el consumo E – 220V esta permitido para Electrodomestico

Ademas, tambien se permite el color BEIGE, entre la lista posible de colores de los electrodomésticos

a) Realizar el cambio correspondiente a la clase Electrodomestico

b)Agrega la HornoElectrico como sub-clase del Electrodomestico ya creado anteriormente en la Actividad 3.Se debe agregar un atributo *double calorias*, que da una idea de cuanto calienta el horno.

Debe tener diferentes constructores, al igual que el electrodoméstico.  
  
-El constructor por defecto (precio base: $800, color BEIGE, consumo E, 5 kg, voltaje 220 V, 1000 calorías).  
-Un constructor que solo pida el precio y calorías, el resto por defecto.

Metodos:

int getCalorias(): para conocer la cantidad de carga del HornitoElectrico.

boolean puedoCalentar(int calorias): si la cantidad de calorías supera las del hornito, devuelve false, caso contrario true.