Exercise

Alexandre Bergel
Nancy Hitschfeld
07/10/2020

Friday 16 October

At 10:00am, we will upload on u-cursos a .pdf file that contains MiniTarea 1

You will have 26 hours to do the MiniTarea 1

You code has to be uploaded on u-cursos on Saturday 17 octubre, at 12:00pm (medio día)

The time to complete the minitarea is between 20 and 45 minutos

Monday 26 October at 12:00, we will see a solution to the mini tarea

MiniTarea are *individual* (even if we do breakout group to do exercises in the classroom)

No atraso will be allowed

Exercise (from a previous control)

Un inch es una métrica para medir distancias.
1 inch = 0.0254 meter.

El objetivo de esta pregunta es de definir al menor 2 clases, Meter y Inch, para sumar valores de diferentes unidades.

Por ejemplo:
1 metro + 39 inches = 1.9906 metros
10 inches + 2 metros = 88.74 inches

Se recuerda que 1 metro = 39.37 inches y 1 inch = 0.0254 metros.

- Escriba el código necesario para poder crear y sumar valores de diferentes unidades.

No estamos interesados en tener una solución que funcione, sino que apliquen el conocimiento adquirido en clase

Solution

Diseño

This exercise is very similar to the Money/MoneyBag and Cachipun exercises

The idea is to use a double dispatch to avoid having to check the unit of the distance

```
public interface Length {
    float value();
    Length add(Length l);
    Length addInch(InchLength inchLength);
    Length addMeter(MeterLength meterLength);
}
```

```
public class MeterLength implements Length {
    private float value;

    public MeterLength(float i) {
        value = i;
    }

    @Override
    public Length add(Length l) {
        return l.addMeter(this);
    }

    @Override
    public float value() {
        return value;
    }

    @Override
    public Length addInch(InchLength inchLength) {
        return new InchLength ((float) (value * 39.37 + inchLength.value()));
    }

    @Override
    public Length addMeter(MeterLength meterLength) {
        return new MeterLength ((float) (value + meterLength.value()));
    }
}
```

```
public class InchLength implements Length {
    private float value;

    public InchLength(float i) {
        value = i;
    }

    @Override
    public Length add(Length l) {
        return l.addInch(this);
    }

    @Override
    public float value() {
        return value;
    }

    @Override
    public Length addInch(InchLength inchLength) {
        return new InchLength (value + inchLength.value());

    }

    @Override
    public Length addMeter(MeterLength meterLength) {
        return new MeterLength ((float) (value * 0.0254 + meterLength.value()));
    }
}
```

License

dcc

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

www.dcc.uchile.cl

/ DCCUCHILE