

Exercicio

Suponha agora que o projeto atual seja uma loja virtual. Essa loja virtual possui um carrinho de compras, que guarda uma lista de itens comprados. Um item possui a descrição de um produto, a quantidade, o valor unitário e o valor total desse item. Veja o código que representa esse carrinho:

```
public class CarrinhoDeCompras {  
    private List<Item> itens;  
    public CarrinhoDeCompras()  
    {this.itens = new ArrayList<Item>();}  
    public void adiciona(Item item)  
    {this.itens.add(item);}  
    public List<Item> getItens()  
    {return Collections.unmodifiableList(itens);}  
}
```

```
public class Item {  
    private String descricao;  
    private int quantidade;  
    private double valorUnitario;  
    public Item(String descricao, int quantidade, double valorUnitario) {  
        this.descricao = descricao;  
        this.quantidade = quantidade;  
        this.valorUnitario = valorUnitario;  
    }  
    public double getValorTotal()  
    {  
        return this.valorUnitario * this.quantidade;    }  
    // getters para os atributos  
}
```

Agora imagine que o programador deva implementar uma funcionalidade que devolva o valor do item de maior valor dentro desse carrinho de compras.

Pensando já nos testes, temos os seguintes cenários:

- Se o carrinho só tiver um item, ele mesmo será o item de maior valor.
- Se o carrinho tiver muitos itens, o item de maior valor é o que deve ser retornado.
- Um carrinho sem nenhum item deve retornar zero.

Crie um projecto com o nome TestesJUnit e inclua as classes conversorTemperatura e TemperaturaCelsius.

```
public class ConversorTemperatura {  
    public static double converteCelsiusParaFahrenheit(double celsius) {  
        double fahrenheit = 1.8*celsius+32; // f =1,8*c+32  
        return fahrenheit;  
    }  
    public static double converteFahrenheitParaCelsius(double fahrenheit) {  
        double celsius = (5/9)*(fahrenheit-32); // c =(5/9)*(f-32)  
        return celsius;  
    }  
}
```

```
public class TemperaturaCelsius {  
    private double celsius;  
    public TemperaturaCelsius (double celsius) {  
        this.celsius = celsius;  
    }  
    public double converteCelsiusParaFahrenheit() {  
        double fahrenheit = 1.8*celsius+32; // f =1,8*c+32  
        return fahrenheit;  
    }  
}
```

A. Crie testes unitários para a classe ConversorTemperatura:

B. Crie testes unitários para a classe TemperaturaCelsius:

Crie uma aplicação para criptografar e descriptografar usando a Cifra de César. Esta é uma forma simples de criptografia, que consiste em mover as letras do alfabeto. Dado um número n , vamos mover as letras no alfabeto, por exemplo:

Dado $n = 3$ e o texto "Rafael Guimaraes Sakurai", vamos aumentar em 3 cada caractere, assim a versão criptografada ficará "Udidho Jzlpdudhv Vdnzudi", e para descriptografar basta diminuir em 3 cada caractere.

Crie uma aplicação para validação de CPF. Em um CPF, os dois últimos dígitos, são os dígitos verificadores, que são obtidos a partir de um calculo sobre os nove primeiros dígitos, a formula de geração dos dígitos verificadores é a seguinte:

1. Calcular o primeiro digito verificador.

Para calcular o primeiro dígito verificador, precisamos multiplicar sequencialmente os primeiros nove números do CPF por 10, 9, 8, 7, 6, 5, 4, 3 e 2 respectivamente, e somar o resultado, exemplo:

CPF 3 3 3 1 2 4 9 8 7

Multiplicador 10 9 8 7 6 5 4 3 2

Resultado 30 27 24 7 12 20 36 24 14

A soma total de cada resultado é 194.

Agora, divida o resultado total por 11, se o resto da divisão for menor que 2, então o primeiro digito verificador é zero, caso contrário subtraia o total de 11 para obter o primeiro dígito.

Neste exemplo o resto da divisão de 194 por 11 é 7, então precisamos subtrair 11 - 7 para obter 4 como o primeiro dígito verificador.

2. Calcular o segundo dígito verificador.

Para calcular o segundo dígito verificador vamos fazer um cálculo similar ao do primeiro dígito, só que agora temos o primeiro dígito verificador, então iremos multiplicar os primeiros dez números do CPF por 11, 10, 9, 8, 7, 6, 5, 4, 3 e 2 respectivamente, e somar o resultado, exemplo:

CPF 3 3 3 1 2 4 9 8 7 4

Multiplicador 11 10 9 8 7 6 5 4 3 2

Resultado 33 30 27 8 14 24 45 32 21 8

A soma total de cada resultado é 242.

Agora, divida o resultado total por 11, se o resto da divisão for menor que 2, então o segundo dígito verificador é zero, caso contrário subtraia o total de 11 para obter o segundo dígito.

Neste exemplo o resto da divisão de 242 por 11 é 0, então usamos zero como segundo dígito verificador.

Então, para este exemplo de CPF: 333.124.987, temos o resultado 40 como dígito verificador.