

## Atividade de Fixação - Teste Automatizados – JUnit

### Parte II

1. 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 cálculo sobre os nove primeiros dígitos, a formula de geração dos dígitos verificadores é a seguinte:

#### **1. Calcular o primeiro dígito 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 do resultado é 194.

Divida a soma total por 11, se o resto da divisão for menor que 2, então o primeiro dígito verificador é zero, caso contrário subtraia do valor 11 o resto da divisão para obter o primeiro dígito.

Neste exemplo o resto da divisão de 194 por 11 é igual a 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 do resultado é 242.

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 de 11 o resto da divisão para obter o segundo dígito.

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

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

Crie uma classe (VerificaCPF.java) para compor um objeto VerificaCPF, esse objeto terá um método estático que retornará verdadeiro se um CPF for considerado valido.

Para se ter certeza da implementação realizada corretamente, crie uma classe (TestVerifcaCPF.java) para compor todos os casos de teste.

2. Dado o chassi de um veículo, como, por exemplo, 9BP17164GA0000001, será necessário identificar qual o ano de fabricação deste veículo. Todo chassi possui um caractere alfa numérico (letra ou número) que possui esta informação.

Neste exemplo de chassi, o caractere 'A' na 10ª posição informa que o veículo é do ano 2010.

A tabela abaixo possui alguns valores e os anos que eles representam:

Valor	Ano
A	2010
B	2011
C	2012

Crie uma classe (Chassi.java) para compor um objeto Chassi, esse objeto terá um método estático que retornará o ano correto de um chassi.

Para se ter certeza da implementação realizada corretamente, crie uma classe (TestChassi.java) para compor todos os casos de teste:

- Testar se o chassi com o valor A retorna o ano 2010.
- Testar se o chassi com o valor B retorna o ano 2011.
- Testar se o chassi com o valor C retorna o ano 2012.
- Testar se um chassi pode ser escrito com caracteres em maiúsculo ou minúsculo.
- Testar se um chassi incompleto retorna um erro de informação inválida.

Por exemplo, sabemos que, ao informar o chassi 9BP17164GA0000001 e a posição 10, é esperado que a aplicação informe que o ano desse chassi é 2010. Portanto podemos criar uma classe contendo um teste que espera que isso aconteça.

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

Exemplo:

Dado  $n = 3$  e o texto “Rafael Guimaraes Sakurai”, vamos deslocar para esquerda o alfabeto em 3 caractere, assim a versão criptografada ficará “Udidho Jzlpdudhv Vdnzudl”, e para descriptografar basta diminuir em 3 cada caractere.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Boa sorte !