

Analisando o processo de conversão de um número inteiro para o seu correspondente texto por extenso percebemos que existe muita coisa em comum quando são adicionados novos dígitos.

1. Crie uma família de classes (utilize herança) para transformar um número inteiro em seu correspondente texto por extenso sendo possível executar o código abaixo.

```
ConverterExtenso ce = null;
ce = new ConverterExtenso4Digitos(9999);
System.out.println(ce.toString());
// -> Nove Mil Novecentos e Noventa e Nove
ce = new ConverterExtenso3Digitos(999);
System.out.println(ce.toString());
// -> Novecentos e Noventa e Nove
ce = new ConverterExtenso2Digitos(99);
System.out.println(ce.toString());
// -> Noventa e Nove
ce = new ConverterExtenso1Digito(9);
System.out.println(ce.toString());
// -> Nove
```

Segue exemplo:

```
package br.edu.ifba.saj.ads.poo.extenso;

... class ConverterExtenso1Digito {
    ... int numero;
    ... String[] porExtenso = { "Zero", "Um", "Dois", "Três", "Quatro", "Cinco", "Seis", "
    ... ConverterExtenso1Digito ...

    ...

    @Override
    ... String toString() {
        ... return porExtenso[this.numero%10];
    }
}
```

```
... class ConverterExtenso2Digitos ... {
    String[] dezenasBaixas = { "Dez", "Onze", "Doze", "Treze", "Quatorze", "Quinze", "Dezes
        "Dezenove" };
    String[] porExtenso = { "", "", "Vinte", "Trinta", "Quarenta", "Cinquenta", "Sessenta"
        "Oitenta", "Noventa" };

    ... ConverterExtenso2Digitos ...
```

```
...

@Override
... String toString() {
    if(... < 10){
        return ...(passa a bola para ConverterExtenso1Digito);
    }else{
        if( ... < 20 ){
            return dezenasBaixas[this.numero%10];
        }
        String retorno = porExtenso[... / 10];

        if (... % 10 != 0) {
            retorno+=" e " ...(passa a bola para ConverterExtenso1Digito);
        }
        return retorno;
    }
}
}
```

2. Utilizou Polimorfismo? Onde, e porque?
3. É possível utilizar o `super` para resolver o problema?
4. Utilizou Encapsulamento (métodos de acesso e modificadores de acesso)? Onde, e porque?
5. Utilizou (Poderia ter utilizado) Métodos/Atributos de Classe? Caso sim, de que forma?