

Enum

```
45: public void setUnidade(String unidade) {
46:     switch (unidade) {
47:         case "KG": case "LT": case "CX": case "MT": case "UN":
48:             this.unidade = unidade;
49:             break;
50:         default:
51:             this.unidade = "UN";
52:     }
53: }
```

celular.setUnidade("APARELHO"); // => "UN"

```
1: public enum UnidadeMedida {
2:     KG,
3:     LT,
4:     CX,
5:     MT,
6:     UN;
7: }
```

```
8: public abstract class Produto {
9:
10:     private static int ultimoCodigo = 0;
11:
12:     private Integer codigo;
13:     private String descricao;
14:     private UnidadeMedida unidade;
15:     private Double valor;
16:     ...
44:     public void setUnidade(UnidadeMedida unidade) {
45:         this.unidade = unidade;
46:     }
47: }
```

~~celular.setUnidade("UN");~~

celular.setUnidade(UnidadeMedida.UN);
celular.setUnidade(UnidadeMedida.CX);

Enum Complexo

- VIVO: Telefônica Brasil S.A – 8955
- TIM: Tim Brasil – 144
- CLARO: Claro S/A – 552
- OI: Telemar Norte S.A - 1057

```

1: public enum Operadora {
2:
3:     //CONSTANTE(String, int)
4:     VIVO("Telefônica Brasil S.A", 8955),
5:     TIM("Tim Brasil", 144),
6:     CLARO("Claro S/A", 552),
7:     OI("Telemar Norte S.A", 1057),
8:     NEXTEL("Nextel Telecomunicações S/A", 613);
9:
10:     final private String nomeEmpresa;
11:     final private int numeroAtivacao;
12:
13:     private Operadora(String nomeEmpresa, int numeroAtivacao) {
14:         this.nomeEmpresa = nomeEmpresa;
15:         this.numeroAtivacao = numeroAtivacao;
16:     }
17:
18:     public String getNomeEmpresa() {
19:         return nomeEmpresa;
20:     }
21:
22:     public int getNumeroAtivacao() {
23:         return numeroAtivacao;
24:     }
25:
26: }

```

Métodos estáticos e formatação:

```
private static final Locale DEFAULT_LOCALE = new Locale("pt", "BR");
```

```
static public String numero(double valor) {
    return getNumberInstance(DEFAULT_LOCALE).format(valor);
}
```

```
static public String moeda(double valor) {
    return getCurrencyInstance(DEFAULT_LOCALE).format(valor);
}
```

Data:

```

21: static public String data(LocalDate data, FormatStyle formato) {
22:     return data(data, DEFAULT_LOCALE, formato);
23: }
24:
25: static public String data(LocalDate data, Locale locale, FormatStyle formato) {
26:     Locale.setDefault(locale);
27:     return DateTimeFormatter.ofLocalizedDate(formato).format(data);
28: }
29:
30: static public String data(LocalDate data, String padrao) {
31:     return DateTimeFormatter.ofPattern(padrao).format(data);
32: }

```