Tipos Enumeráveis e Registros

Prof. Pedro Pongelupe

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Curso de Engenharia de Software

Sumário

Tipos Enumeráveis

O quê são?

Os tipos enumeráveis, ou enums, são um tipo especial de dados que permitem que uma variável seja parte de um conjunto predefinido de constantes.

- Por ser constante, o nome de um enum deve ser escrito em letras maiúsculas.

Exemplo

```
public enum DiaSemana {
    DOMINGO, SEGUNDA, TERCA, QUARTA, QUINTA, SEXTA, SABADO;
}
```

Tipos Enumeráveis

Quando usar?

É recomendado utilizar enuns quando precisamos representar um conjunto **fixo** de constantes. Por exemplo:

- Planetas do Sistema Solar
- Itens de um menu
- Comandos possíveis de um sistema

Tipos Enumeráveis

Propriedades de Enums

- Podem possuir atributos
- Podem possuir métodos
- Todos estendem *implicitamente* de java.lang.Enum, logo, um enum não pode estender outra classe em Java.
- Os construtores de um enum devem ser **private** ou **package-private (default)**.
- Podemos acessar todos os elementos de um enum estaticamente como: <Enum>.values(). ex. DiaSemana.values()

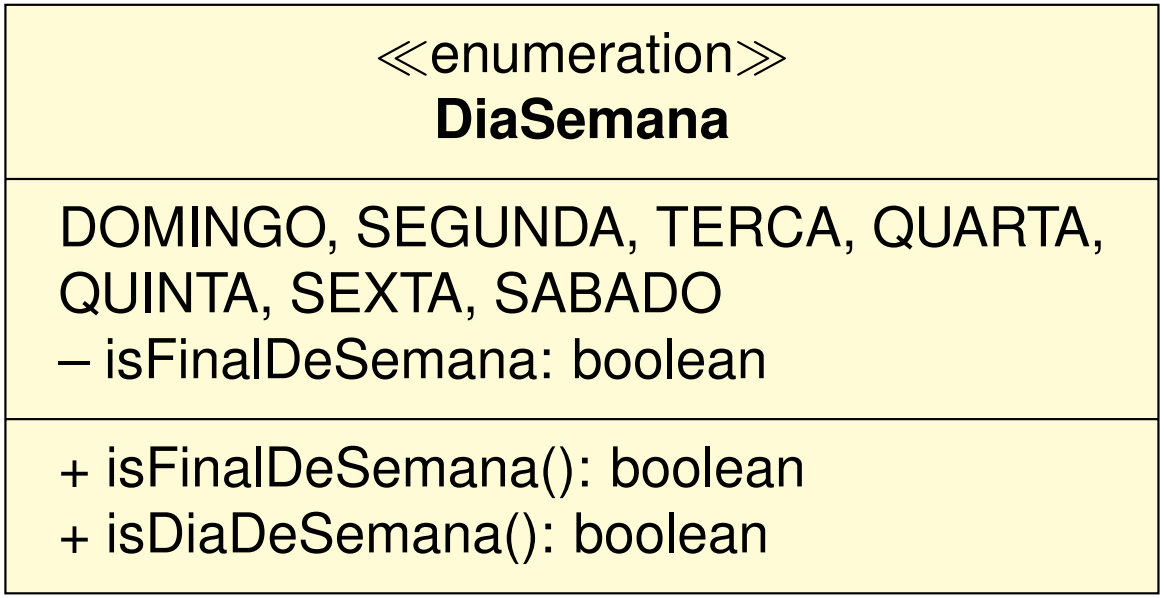
Exemplo Dia Semana

```
public enum DiaSemana {
    DOMINGO(true), SEGUNDA, TERCA, QUARTA,
    QUINTA, SEXTA, SABADO(true);

    private final boolean isFinalDeSemana;

    private DiaSemana(boolean isFinalDeSemana) {
        this.isFinalDeSemana = isFinalDeSemana;
    }
    private DiaSemana() {
        this(false);
    }

    public boolean isFinalDeSemana() { return isFinalDeSemana; }
    public boolean isDiaDeSemana() { return !isFinalDeSemana; }

}
```

Exemplo Dia Semana

```
public class Driver {
    public static void main(String[] args) {
        List.of(DiaSemana.values())
            .stream()
            .filter(DiaSemana::isFinalDeSemana)
            .forEach(System.out::println);
    }
}
```

UML Enum


≪enumeration≫
DiaSemana
DOMINGO, SEGUNDA, TERCA, QUARTA,
QUINTA, SEXTA, SABADO
– isFinalDeSemana: boolean
+ isFinalDeSemana(): boolean
+ isDiaDeSemana(): boolean

Registros

O quê são?

Os registros, ou records, são um tipo especial de dados que funcionam como transportadores transparentes de dados imutáveis. Descrito pela JEP-395 da release 16 do Java.

Um registro em Java define um cabeçalho com as seguintes informações:

- Métodos de acesso apropriados (getter)
- Construtor
- equals
- hashCode
- toString

Java Class

```
public final class Rectangle {
    private final double length;
    private final double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    double length() { return this.length; }
    double width()  { return this.width; }

    @Override
    public boolean equals(Object obj) {...}

    @Override
    public int hashCode() {...}

    @Override
    public String toString() {...}
}
```

Record

```
record Rectangle(double length, double width) { }
```

Instanciando um novo Record

Os records representam um tipo especial de classes em Java. Para criar um nova instância de um record, utilizamos a palavra chave new.

```
record Rectangle(double length, double width) { }
...
Rectangle r = new Rectangle(4,5);
```

Registros

Restrições

- Não é possível uma classe explicitamente estender um registro.

Possibilidades

- É possível criar um registro genérico

  record Triangulo<C **extends** Coordenada> (C topo, C esquerda, C direita) { }
- Um registro pode implementar uma ou mais interfaces

  record Aluno (...) **implements** Matriculavel { }
- Um registro pode ser definido como um membro interno de uma classe

Record como membro de uma classe

```
public class Driver {
        private record Suco(String fruta, int quantidadeAgua) {}

        public static void main(String[] args) {
            var sucoMelancia = new Suco("melancia", 500);
            var sucoLimao = new Suco("Limao", 1500);
            System.out.println("Total Suco: "
                + (sucoMelancia.quantidadeAgua()
                 + sucoLimao.quantidadeAgua()));
        }
}
```

Obrigado!!

Muito obrigado pela atenção! Alguma dúvida? Bora praticar!!!

"*Nada de desgosto, nem de desânimo; se acabas de fracassar, recomeça.* "