



Enumerações

Prog Orientada a Objetos

Prof. João Paulo A. Almeida

Baseado em slides do Prof. Vítor Souza
(usado com permissão)

- Uso mais comum: constantes
- Porém, servem para qualquer classe com instâncias fixas em tempo de projeto

Constantes (ainda sem enumerações)

```
public class TamanhosDeCamiseta
{
    public static final int PP=0;
    public static final int M=1;
    public static final int G=2;
    public static final int GG=3;
}
```

// TamanhosDeCamisa.PP, TamanhosDeCamisa.M, etc.

Constantes (ainda sem enumerações)

```
public class Estacoes
```

```
{
```

```
    public static final int PRIMAVERA=0;
```

```
    public static final int VERAO=1;
```

```
    public static final int OUTUNO=2;
```

```
    public static final int INVERNO=3;
```

```
}
```

```
// Estacoes.PRIMAVERA, TamanhosDeCamisa.VERAO, etc.
```

Constantes (ainda sem enumerações)

```
public class Estacoes  
{
```

```
    public static final int PRIMAVERA=0;
```

```
    public static final int VERA0=1;
```

```
    public static final int OUTUNO=2;
```

```
    public static final int INVERNO=3;
```

```
}
```

// Estacoes.PRIMAVERA, TamanhosDeCamisa.VERAO, etc.

Desvantagem principal:

- Não há tipagem... valores são inteiros, podendo user usado como inteiros sem distinção.

Exemplos de constantes em Java

- [DateFormat](#)
- FULL, LONG, MEDIUM, SHORT, DEFAULT
- <https://docs.oracle.com/javase/8/docs/api/java/text/DateFormat.html#FULL>

Vejam todas as constantes em:

- <https://docs.oracle.com/javase/8/docs/api/constant-values.html#java.text.DateFormat.FULL>

- Tipos enumerados são aqueles que possuem um conjunto finito de valores que as variáveis podem assumir:
- Ex.: estações do ano, naipes ou cartas do baralho, planetas do sistema solar, etc.
- A palavra-chave **enum** define um tipo enumerado:

```
enum Estacao { PRIMAVERA, VERA0, OUTONO, INVERNO };
```

Sempre em maiúsculas (convenção)

Exemplo de Enumeração

```
public enum Planet {  
    MERCURY (3.303e+23, 2.4397e6),  
    VENUS    (4.869e+24, 6.0518e6),  
    EARTH    (5.976e+24, 6.37814e6),  
    MARS     (6.421e+23, 3.3972e6),  
    JUPITER  (1.9e+27,    7.1492e7),  
    SATURN   (5.688e+26, 6.0268e7),  
    URANUS   (8.686e+25, 2.5559e7),  
    NEPTUNE  (1.024e+26, 2.4746e7);  
  
    private final double mass;    // in kilograms  
    private final double radius; // in meters  
    Planet(double mass, double radius) {  
        // no caso de enum, default é private  
        this.mass = mass;  
        this.radius = radius;  
    }  
    public double mass() { return mass; }  
    public double radius() { return radius; }  
}
```


Built-ins: public static enum-type[] values()

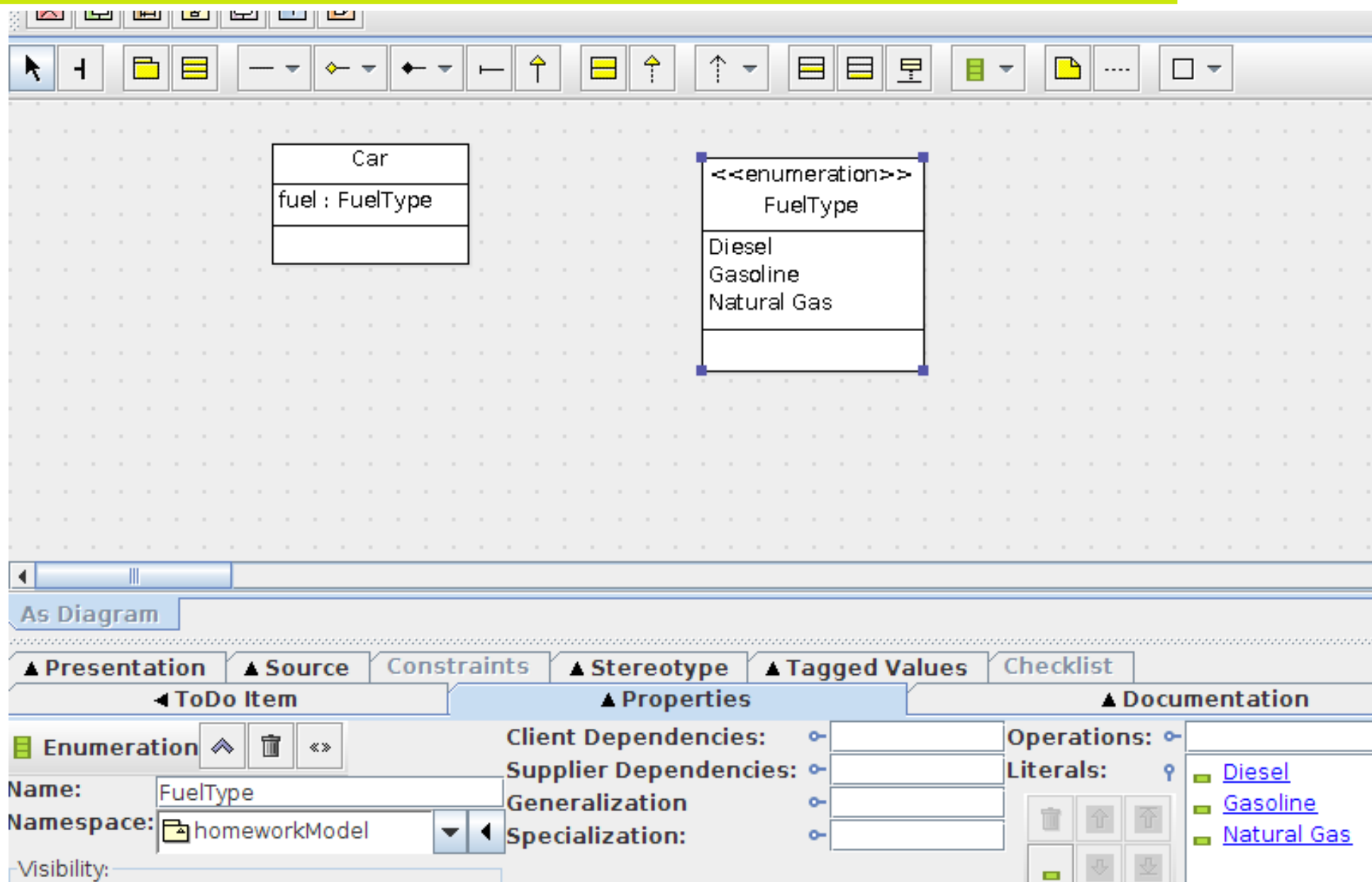
```
enum DayOfWeek {  
    MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY,  
    SUNDAY  
}
```

```
public class MainClass {  
    public static void main(String args[]) {  
        System.out.println("Here are all Week constants");  
  
        // use values()  
        DayOfWeek allDays[] = DayOfWeek.values();  
        for (DayOfWeek aday : allDays) {  
            System.out.println(aday);  
        }  
    }  
}
```

```
enum Weekday {  
    MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY,  
    SATURDAY, SUNDAY  
}
```

```
public class MainClass {  
    public static void main(String args[]) {  
        Weekday day;  
  
        //  
        day = Weekday.valueOf("MONDAY");  
        System.out.println("day contains " + day);  
    }  
}
```

Enums em UML



<http://stackoverflow.com/questions/412944/uml-class-diagram-enum>

Enums possuem características de classe

```
public enum Comando {  
    AJUDA("?", "Mostra esta lista de comandos."),  
    ADICIONAR("adic", "Adiciona um novo contato."),  
    LISTAR("list", "Lista os contatos."),  
    SAIR("sair", "Sai do programa."),  
    DESCONHECIDO("", "");  
  
    private final String nome;  
    private final String descricao;  
  
    private Comando(String nome, String descricao) {  
        this.nome = nome;  
        this.descricao = descricao;  
    }  
  
    public String getNome() {  
        return nome;  
    }  
  
    /* Continua... */
```

Enums possuem características de classe

```
public String toString() {
    if (this == DESCONHECIDO) return "";
    return "- " + nome + ": " + descricao;
}

public static Comando obterComando(String linha) {
    // trim
    int idx = linha.indexOf(' ');
    if (idx != -1) linha = linha.substring(0, idx);
    linha = linha.toLowerCase();

    for (Comando comando : Comando.values())
        if (comando.nome.equals(linha))
            return comando;

    return DESCONHECIDO;
}
```

Enums possuem características de classe

```
/* No método main() ... */
try (Scanner scanner = new Scanner(System.in)) {
    String linha = scanner.nextLine();
    Comando comando = Comando.obtemComando(linha);

    while (comando != Comando.SAIR) {
        switch (comando) {
            case AJUDA:
                System.out.printf("Comandos disponíveis:\n\n");
                for (Comando cmd : Comando.values())
                    System.out.printf("%s\n", cmd);
                break;

            case ADICIONAR:
                // etc...
                break;
        }
    }
}
```