













UPskill - JAVA + .NET

Programação Orientada a Objetos - Tipo Enumerado

Adaptado de Donald W. Smith (TechNeTrain)



Enum Types (1)



- Um tipo enumerado é um tipo especial que permite predefinir um conjunto de valores que poderão ser assumidos por uma variável
- A variável assume um dos valores predefinidos
- Uma vez que os diferentes valores são constantes, devem ser representados com letra maiúscula



















Enum Types (2)

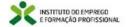


 Exemplo: Tipos enumerados para representar um dos 4 pontos cardeais (norte, sul, este, oeste) e os dias da semana

```
public enum CardinalPoints {
    NORTH, SOUTH, EAST, WEST
}

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



















Enum Type: Exemplo 1



```
public class EnumTest {
    Day day;
    public EnumTest(Day day) {
        this.day = day;
    public void tellItLikeItIs() {
        switch (day) {
            case MONDAY:
                System.out.println("Mondays are bad.");
                break;
            case SATURDAY: case SUNDAY:
                System.out.println("Weekends are best.");
                break;
```

















Enum Type: Exemplo 1 (cont.)



```
public static void main(String[] args) {
        EnumTest firstDay = new EnumTest(Day.MONDAY);
        firstDay.tellItLikeItIs();
        EnumTest seventhDay = new EnumTest(Day.SUNDAY);
        seventhDay.tellItLikeItIs();
    }
}
```

Output:

Mondays are bad. Weekends are best.



















Enum Type: uma Classe



- A declaração enum define uma classe designada por enum type
- O corpo da classe enum pode incluir métodos e outros campos
- As constantes têm que ser definidas antes dos atributos e dos métodos
- Quando existem atributos e métodos, a declaração das constantes tem que terminar com;



















Enum Type: uma Classe



- O compilador adiciona automaticamente alguns métodos especiais quando cria um enum.
- Por exemplo, inclui um método estático que devolve um array contendo todos os valores de um enum na ordem pela qual foram declarados

```
for (Planet p : Planet.values()) {
   System.out.printf("Your weight on %s is %f%n", p, p.surfaceWeight(mass));
}
```

Planet é um tipo enumerado. O método values() pode ser usado num ciclo *for each* para iterar todos os valores de um tipo enumerado



















Enum Type: métodos pré-definidos



Método	Instância / classe	Descrição
ordinal()	instância	Devolve o inteiro correspondente à ordem de uma constante
name()	instância	Converte uma constante numa string equivalente à sua forma literal
valueOf(String s)	classe	Devolve uma constante equivalente à string s, a qual deverá ser uma representação válida de uma das constantes do tipo enumerado
values()	classe	Devolve um <i>array</i> com as constantes da classe à qual foi aplicado o método

toString(), compareTo(), equals()



















Enum Type: Construtor



- Quando as constantes de um tipo enumerado são declaradas com valores, estes valores são passados ao construtor no momento em que a constante é criada
- O construtor de um enum type deve ter acesso package private ou private
- O construtor cria automaticamente as constantes que estão definidas no início do corpo
- Não é possível invocar explicitamente o construtor



















Enum Type: Exemplo 2

```
UPskill
```

```
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);
   private final double mass;
   private final double radius;
   Planet(double mass, double radius) {
       this.mass = mass;
       this.radius = radius;
   private double mass() { return mass; }
   private double radius() { return radius; }
   public static final double G = 6.67300E-11;
   double surfaceGravity() {
       return G * mass / (radius * radius);
   double surfaceWeight(double otherMass) {
       return otherMass * surfaceGravity();
```

```
public static void main(String[] args) {
    double earthWeight = 175;
    double mass = earthWeight/
        EARTH.surfaceGravity();

    for (Planet p : Planet.values())
        System.out.printf("Your weight
        on %s is %f%n",
        p, p.surfaceWeight(mass));
    }
}
```

Output:

Your weight on MERCURY is 66.107583 Your weight on VENUS is 158.374842 Your weight on EARTH is 175.000000 Your weight on MARS is 66.279007















