fusberti@ic.unicamp.br

Enumerações

prof. Fábio Luiz Usberti

MC322 - Programação Orientada a Objetos

Instituto de Computação - UNICAMP - 2014

Sumário

Enumerações

Tipos enumerados

- Um tipo enumerado (ou enumeração) consiste em um conjunto de constantes representadas por identificadores únicos.
- Por serem constantes, os identificadores de uma enumeração são descritos em maiúsculas.
- Diferentemente de outras linguagens, as constantes de uma enumeração em Java não estão associadas a números.
- Uma enumeração pode ser declarada com a palavra-chave `enum` da seguinte forma:

```
<modificador de acesso> enum MyEnum {
    CONSTANTE1, ..., CONSTANTEN
}
```

- Exemplo:

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

Enumerações

Tipos enumerados

- A enumeração é um tipo especial de dados que permite uma variável (que referencia a enumeração) ser um dos elementos do conjunto de constantes pré-definidas.
- Uma variável de tipo enumerado deve ser igual a uma das constantes declaradas na enumeração.
- As constantes de uma enumeração são implicitamente finais e estáticas e podem ser usadas como rótulos de uma instrução `switch`.

Enumerações

```
// EnumTest.java
// Classe que testa o uso de uma enumeração em um switch
public class EnumTest {
    public enum Day { // enumeração dos dias da semana
        SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
    } // fim enum

    private Day day; // atributo que armazena um dia da semana

    // construtor
    public EnumTest(Day day) {
        this.day = day;
    } // fim construtor

    // imprime uma mensagem dependendo do dia da semana
    public void tellItLikeItIs () {
        switch (day) {
        case MONDAY: System.out.println("Mondays are bad."); break;
        case FRIDAY: System.out.println("Fridays are better."); break;
        case SATURDAY: case SUNDAY: System.out.println("Weekends are best."); break;
        default: System.out.println("Midweek days are so-so."); break;
        } // fim switch
    } // fim método tellItLikeItIs

    public static void main(String[] args) {
        EnumTest firstDay = new EnumTest(Day.MONDAY);
        firstDay. tellItLikeItIs ();
        EnumTest thirdDay = new EnumTest(Day.WEDNESDAY);
        thirdDay. tellItLikeItIs ();
        EnumTest fifthDay = new EnumTest(Day.FRIDAY);
        fifthDay. tellItLikeItIs ();
        EnumTest seventhDay = new EnumTest(Day.SUNDAY);
        seventhDay. tellItLikeItIs ();
    } // fim método main
} // fim classe EnumTest
```

Enumerações

Tipos enumerados

- Toda enumeração possui um método estático `values()` que retorna um array com as constantes enumeradas na mesma ordem em que elas foram declaradas.
- É comum combinar o método `values()` com o laço `for` aprimorado.
- Quando uma constante enumerada é convertida para `String`, seu identificador é usado na representação da `String`.

Classes `enum`

Tipos enumerados e classes

- Em Java, um tipo enumerado consiste em uma classe, portanto pode incluir componentes como construtores, atributos e métodos.
- A declaração de uma enumeração consiste de duas partes: declaração das constantes e declaração dos membros de classe (atributos, construtores e métodos).
- As constantes devem ser declaradas antes dos atributos e métodos.
- Quando métodos e atributos forem declarados, a lista de constantes da `enum` deve terminar com um ponto-e-vírgula (`;`)

Classes enum

Tipos enumerados e classes

- Uma constante de enumeração pode ser considerada como uma variável (estática e final) que referencia um objeto do tipo enumerado.
- A declaração de uma constante de enumeração corresponde à instanciação de um objeto, logo argumentos podem ser passados para o construtor.
- O objeto referenciado por uma constante mantém suas próprias cópias das variáveis de instância.
- O construtor chamado é escolhido de acordo com as mesmas regras de métodos sobrecarregados, ou seja, depende da lista de argumentos passados na chamada.

Classes enum

Exemplo

```
// Coin.java
// Classe que mostra uma enumeração como uma classe
public enum Coin {
    // declaração das constantes
    PENNY(1), NICKEL(5), DIME(10), QUARTER(25);

    // método construtor de uma constante enumerada
    Coin(int value) {
        this.value = value;
    }

    // atributo
    private final int value;

    // método acessor
    public int getValue() {
        return value;
    }

    public static void main(String[] args) {
        for (Coin s : Coin.values())
            System.out.println(s + " = " + s.getValue());
    } // fim método main
} // fim enum Coin
```

Classes enum

Impressão no terminal:

```
PENNY = 1
NICKEL = 5
DIME = 10
QUARTER = 25
```

Classes `enum`

Tipos enumerados e classes

- Os construtores de um tipo enumerado são implicitamente privados. É um erro de compilação declarar um construtor de uma enumeração com os modificadores de acesso `public` ou `protected`.
- Um tipo enumerado não contém outras instâncias fora aquelas definidas por constantes.
- Tentar instanciar explicitamente um objeto de tipo enumerado (usando a palavra-chave `new`) consiste em um erro de compilação.
- Atributos estáticos (com exceção de constantes primitivas) de um tipo enumerado não podem ser referenciados pelos construtores da enumeração.

Classes enum

```
// Book.java
// Classe Book exemplifica tipos enumerados com construtores, atributos e métodos acessores
public enum Book {

    // declarando as constantes da enumeração
    JCP2008("Java Como Programar", "2008"),
    JIAN2005("Java in a Nutshell", "2005"),
    TJPL2005("The Java Programming Language", "2005"),
    TIJ2000("Thinking in Java", "2000");

    // atributos de instância
    private final String title; // título
    private final String copyrightYear; // ano de publicação

    Book(String bookTitle, String year) {
        title = bookTitle;
        copyrightYear = year;
    } // fim construtor

    public String getTitle() {
        return title;
    } // fim método getTitle

    public String getCopyrightYear() {
        return copyrightYear;
    } // fim método getCopyrightYear

    public static void main(String args[]) {
        System.out.println("Lista dos livros:\n");
        // imprime todos os livros da enumeração
        for (Book book : Book.values())
            System.out.printf("%-10s%-45s%s\n", book, book.getTitle(), book.getCopyrightYear());

    } // fim main
} // fim enum Book
```

Classes enum

Exemplo: impressão no terminal

```
Lista dos livros:

JCP2008   Java Como Programar                2008
JIAN2005  Java in a Nutshell                 2005
TJPL2005  The Java Programming Language      2005
TIJ2000   Thinking in Java                   2000
```

Classes enum

```
// Planets.java
// Classe Planets determina os pesos de um objeto na superfície de outros planetas.
public enum Planets {
    MERCURIO(3.303e+23, 2.4397e6),   VENUS(4.869e+24, 6.0518e6),
    TERRA(5.976e+24, 6.37814e6),     MARTE(6.421e+23, 3.3972e6),
    JUPITER(1.9e+27, 7.1492e7),      SATURNO(5.688e+26, 6.0268e7),
    URANO(8.686e+25, 2.5559e7),      NETUNO(1.024e+26, 2.4746e7);

    public final double mass; // in kilograms
    public final double radius; // in meters

    Planets(double mass, double radius) {
        this.mass = mass;
        this.radius = radius;
    } // fim construtor

    // constante universal de gravitação (m3 kg-1 s-2)
    public static final double G = 6.67300E-11;

    double surfaceGravity() {
        return G * mass / (radius * radius);
    } // fim método surfaceGravity

    double surfaceWeight(double otherMass) {
        return otherMass * surfaceGravity();
    } // fim método surfaceWeight

    public static void main(String[] args) {
        double earthWeight = 10; // in Newtons
        double mass = earthWeight / TERRA.surfaceGravity();
        for (Planets p : Planets.values())
            System.out.printf("O peso correspondente no planeta %s é %f N\n", p, p.surfaceWeight(mass));
    } // fim método main
} // fim classe Planets
```

Classes enum

Exemplo: impressão no terminal

```
O peso correspondente no planeta MERCURIO é 3.777576 N
O peso correspondente no planeta VENUS é 9.049991 N
O peso correspondente no planeta TERRA é 10.000000 N
O peso correspondente no planeta MARTE é 3.787372 N
O peso correspondente no planeta JUPITER é 25.305575 N
O peso correspondente no planeta SATURNO é 10.660155 N
O peso correspondente no planeta URANO é 9.051272 N
O peso correspondente no planeta NETUNO é 11.383281 N
```

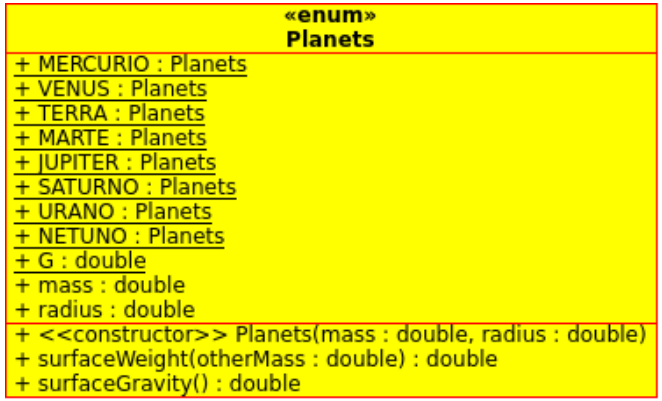
Classes enum

Representação em UML

Enumeração: representação simples de uma enumeração em UML.

Classes enum

Representação em UML



Enumeração: representação detalhada de uma enumeração em UML.

Referências

1. Java: Como Programar, Paul Deitel & Heivey Deitel; Pearson; 7a. Ed. (no. chamada IMECC – 05.133 D368j)
2. Data Structures and Algorithms with Object Oriented Design Patterns in Java, Bruno Preiss; (http://www.brpreiss.com/books/opus6/)
3. The Java Tutorials (Oracle) (http://docs.oracle.com/javase/tutorial/)
4. Guia do Usuário UML, Grady Booch et. al.; Campus(1999)
5. Java Pocket Guide - Robert Liguori & Patricia Liguori; O'Reilley, 2008.