Java enum examples/tutorial

Submitted by alvin on January 13, 2012 - 2:58pm

Java enum FAQ: Can you share some Java *enum* examples, such as how to declare a Java enum, and how to use a Java enum in a *for loop*, if then statement, and Java *switch*statement?

As described in Sun's Java documentation, a Java enum "is a type whose fields consist of a fixed set of constants ... you should use enum types any time you need to represent a fixed set of constants." Let's take a look at some Java enum examples to see how this works.

Java enum - constant examples

Because a Java enum type is good for defining a fixed set of constants, it's often used to represent constants like these in Java programs:

- Days of the week
- Months in the year
- Cards in a deck
- Directions (north, south, east, west)
- Menu item options
- Defining arguments for a command-line program
- States in the United States (such as for a drop-down list)

In your code you'll probably also have other business-specific constants that you'll want to declare in your own enum.

In this *Java enum tutorial*, we'll share a number of enum examples so you can see (a) how to define and then (b) how to use Java enum types in your programs.

Java enum examples - How to declare a simple Java enum type

You can declare simple Java enum type with a syntax that is similar to the Java class declaration syntax. Let's look at several short Java enum examples to get your started.

First, here's an example Java enum type that declares the days of the week in an enum named `Day`:

```
public enum Day
{
```

```
  SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
}
```

Similarly, here's a simple Java enum named `Month` that declares the months in a year:

```
public enum Month
{
  JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE,
  JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER
}
```

And here's an example Java enum type that declares the margin types that you can use when working with CSS:

```
/**
  * Corresponds to CSS margins.
  */
public enum Margin
{
  TOP, RIGHT, BOTTOM, LEFT
}
```

As you can see in these simple examples, to define a Java enum, all you have to do is declare the constants that the enum contains. It's worth nothing that because these are constants, we name them with all capital letters, which is consistent with Java naming conventions.

How to use a Java enum type

Now that you've seen how to declare simple Java enum types, let's take a look at how to use them.

You can use a Java enum type in a variety of situations, including in a Java 5 `for` loop, in a`switch` statement, in an if/else statement, and more. Let's take a look at how to use our simple enum types with each of these Java constructs.

A Java enum for loop example

If you need to iterate through all the constant fields in a Java enum, you can do this pretty easily in a Java 5 for loop. You just need to use the `values` method of the enum type in the`for` loop, as shown in this Java enum for loop example:

```
/**
 * Purpose: A Java enum for loop example.
 * @author alvin alexander, devdaily.com
```

```
 *
 */
public class JavaEnumForLoopExample
{
  public enum Margin
  {
    TOP, RIGHT, BOTTOM, LEFT
  }

  public static void main(String[] args)
  {
    int count = 1;
    // loop through our enum
    for (Margin m: Margin.values())
    {
      System.out.printf("Margin %s = %s\n", count++, m);
    }
  }
}
```

As you can see, we use the `values` method in this line of code:

```
for (Margin m: Margin.values())
```

Here's what the output from this program looks like:

```
Margin 1 = TOP
Margin 2 = RIGHT
Margin 3 = BOTTOM
Margin 4 = LEFT
```

A Java enum switch statement example

You can also use a Java enum in a `switch` statement. Here's the source code for a complete Java enum switch statement example:

```
/**
 * A Java enum switch statement example (a switch/case statement
example).
 * @author alvin alexander, devdaily.com
 *
 */
public class JavaEnumSwitchCaseExample
{

  enum Margin
  {
    TOP, RIGHT, BOTTOM, LEFT
```

```
  }

  public static void main(String[] args)
  {
    System.out.println(getMarginValue(Margin.TOP));
  }

  /**
   * @param A valid Margin value.
   * @return A String representing the value for the given Margin,
   *         or null if the Margin is not valid.
   */
  public static String getMarginValue(Margin margin)
  {
    // use the enum values in our switch statement here
    switch (margin)
    {
      case TOP:     return "1em";
      case RIGHT:   return "12px";
      case BOTTOM:  return "1.5em";
      case LEFT:    return "6px";
      default:      return null;
    }
  }

}
```

As you might guess, the output from this program is:

```
1em
```

A Java enum if/then example

You can also use a Java enum type in an if/then statement. While the example below might be better implemented as a switch/case statement, it does provide an enum if/then example:

```
/**
 * Demonstrates the use of a Java enum type in an if/else
statement.
 * @author alvin alexander, devdaily.com
 */
public class JavaEnumIfThenExample
{

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

```
  }

  public static void main(String[] args)
  {
    Day theDay = Day.THURSDAY;
    printDayGreeting(theDay);
  }

  public static void printDayGreeting(Day day)
  {
    if (day == Day.FRIDAY)
      System.out.println("TGIF");
    else
      System.out.println("Some other day");
  }

}
```

As you can see from that code, an *enum reference* is tested against *an enum constant* in this line of code:

```
if (day == Day.FRIDAY)
```

Java enum idioms

While digging around through the Sun Java enum documentation I found the following "enum idioms", and thought it might help to share them here:

- Because they are constants, the names of an `enum` fields are created in uppercase (like other Java constants).
- You can use an `enum` any time you need to represent a fixed set of constants.

More complicated Java enum examples

In this *enum tutorial* I've only shown the basic ways of declaring and using Java enums. When the need arises, Java enum types can also be much more complicated than what I've shown in these examples. The Planet enum type at the bottom of this Sun enum pageshows how to define an enum that includes fields, methods, and even a `main` method.