Ollscoil na Gaillimhe
University of Galway

CT326 Programming III

LECTURE 8

ENUMS

DR ADRIAN CLEAR
SCHOOL OF COMPUTER SCIENCE

Ollscoil na Gaillimhe
University of Galway

Objectives for today

- Understand and demonstrate how enum types work in Java

Enumerated Types

- An enumerated type is a type whose legal values consist of a fixed set of constants. Common examples include compass directions, which take the values North, South, East and West and days of the week, etc.
- In the Java programming language, you define an enumerated type by using the enum keyword. For example, you would specify a days of the week enumerated type as:

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

- Notice that by convention the names of an enumerated type's values are spelled in uppercase letters.
- You should use enumerated types any time you need to represent a fixed set of constants.

Enumerated Types

- Java programming language enumerated types are much more powerful than their counterparts in other languages, which are just glorified integers.
- The enum declaration defines a class (called an enum type).
- These are the most important properties of enum types:
  - Printed values are informative.
  - They are typesafe and they exist in their own namespace.
  - You can `switch` on an enumeration constant.
  - They have a static `values()` method that returns an array containing all of the values of the enum type in the order they are declared. This method is commonly used in combination with the for-each construct to iterate over the values of an enumerated type.
  - You can provide methods and fields, implement interfaces, and more.

Enumerated Types

- In the following example, `Planet` is an enumerated type that represents the planets in the solar system.
- A `Planet` has constant mass and radius properties. Each enum constant is declared with values for the mass and radius parameters that are passed to the constructor when it is created.
- Note that the constructor for an enum type is implicitly private. If you attempt to create a public constructor for an enum type, the compiler displays an error message.

Enum example

```
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),
    PLUTO   (1.27e+22,  1.137e6);

    private final double mass;   //in kilograms
    private final double radius; //in meters
    Planet(double mass, double radius) {
        this.mass = mass;
        this.radius = radius;
    }
    public double mass()   { return mass; }
    public double radius() { return radius; }

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

    public double surfaceGravity() {
        return G * mass / (radius * radius);
    }
    public double surfaceWeight(double otherMass) {
        return otherMass * surfaceGravity();
    }
}
```

Enum example

- In addition to its properties, `Planet` has methods that allow you to retrieve the surface gravity and weight of an object on each planet.
- Here is a sample program that takes your weight on earth (in any unit) and calculates and prints your weight on all of the planets (in the same unit):

```
public static void main(String[] args) {
    double earthWeight = Double.parseDouble(args[0]);
    double mass = earthWeight/EARTH.surfaceGravity();
    for (Planet p : Planet.values()) {
        System.out.printf("Your weight on %s is %f%n",
                          p, p.surfaceWeight(mass));
    }
}
```

Enum example

- Here's the output:

```
$ java Planet 175
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
Your weight on JUPITER is 442.847567
Your weight on SATURN is 186.552719
Your weight on URANUS is 158.397260
Your weight on NEPTUNE is 199.207413
Your weight on PLUTO is 11.703031
```

- There's one limitation of enum types: although enum types are classes, you cannot define a hierarchy of enums. In other words, it's not possible for one enum type to extend another enum type.

Exercise

- Create a top trumps enum class for a topic of your choice (e.g. cars, dinosaurs), each with four attributes
- Include a compareCards method that takes a category and a variable number of cards and returns the card that wins for that category.

Next time…

- Javadoc and lambda expressions