Tutorial 5 – Meta programming

Pre-requisites

You must have finished the OOP tutorial exercises.

Instructions

In this tutorial, you will learn to apply Java's meta programming to the `PizzaDemo` application. In particular, you will use the reflection and annotation features to annotate and write input validation code for the `Pizza` classes of this application.

1. Copy the content of `tutes.oop2` package to another package named `tutes.meta` and complete the following tasks.

   Use the latest version of your `tutes.oop2` package for this tutorial.

2. Copy the classes provided in the attached `metalib.zip` file into your package.

3. Use the `DomainConstraint` annotation class to define some suitable domain constraints for the `Pizza` attributes. You also need to apply constraints to the `Topping` class attributes as well.

   For example, a pizza's size must be a valid string and the quantities and costs of these toppings must not be negative.

4. The provided `DataManager` class comes with a method to create a new instance of a class from an array of arguments that are for a constructor method of that class. You do not need to implement this method, but you should take a look to see what it does.

   The method definition is:

   ```
   public static Object newInstance(Class c, Object[] attributeVals) throws
   NotPossibleException
   ```

   The parameter `attributeVals` is an array of attribute values whose types match the parameter types of a constructor of `c`. This method validates the input value of each attribute by invoking the `validateDomainAtttribute` method. If a validation fails then this method throws a `NotPossibleException` immediately, otherwise it returns a new object of `c` that is created by invoking the constructor method which takes the arguments specified in the `attributeVals` array.

5. Implement the `validateDomainAttribute` method from the lecture in the `DataManager` class. The header of the method must be the following:

   ```
   public static validateDomainAttribute(Field f, Object value) throws
   NotPossibleException
   ```

   Note that this method throws a `NotPossibleException` immediately after a failed validation.

6. Implement another input validation method which you will use later in the `PizzaDemo` application. It is used as a short-cut to the method in the previous step. The method header is as follows:

   ```
   public static void validateDomainAttribute(Class c, String name,
   Object value) throws NotPossibleException
   ```

   This method should find the attribute of the class `c` that has the given name and invoke the method in the previous step to validate its value.

7. Update the `PizzaDemo` class so that the code that generate new toppings and pizzas must validate the data values before creating a new object. Further, object creation must be performed using the `DataManager.newInstance` method.

   Note: it is suggested that you create several static helper methods to create a new topping and a new instance of each type of pizza.