

DMIT2015 Exercise 2: JavaBean and JavaDoc

Objective

This exercise will show you how to create use the Eclipse IDE and the Java programming language to create Java classes that conform the JavaBean specifications.

BMI JavaBean

Studies have shown that the risk of developing heart disease increases when a person's body mass index (BMI) is greater than 30. In this exercise, you will create a Java class to calculate the body mass index (BMI) of a person. The class diagram for the BMI class is shown on Figure 1. You can compute your BMI based on your weight (in pounds) and height (in inches) using the formula shown on Figure 2. The BMI category of a person is determine from the table shown in Figure 3.

BMI
+ weight: int
+ height: int
+ BMI() :
+ BMI(weight: int, height: int)
+ bmi() : double
+ bmiCategory() : String

Figure 1 - BMI Class Diagram

$$BMI = \frac{703 \times weight}{height^2}$$

Figure 2 - BMI Formula

BMI Category	BMI Range
underweight	less than 18.5
normal	from 18.5 to 24.9
overweight	from 25 to 29.9
obese	30 and over

Figure 3 - BMI Category

Instructions

1. From Eclipse in the Project Explorer view, navigate to **dmit2015-exercises-yourname | Java Resources | src/main/java** . Right-mouse click on the **src/main/java** folder and from the context menu select **New | Package**
2. The "New Java Package" dialog opens. Enter **ca.nait.dmit.domain** in the Name field then click **Finish**
3. Right-mouse click on the **ca.nait.dmit.domain** package you just created and from the context menu select **New | Class**
4. The "New Java Class" dialog opens. Enter **BMI** in the Name field then click **Finish**
5. The Java Editor opens with the BMI.java class loaded. Type in the following fields inside the class.

```
private int weight;
private int height;
```

6. To get Eclipse to generate default no-argument constructor, select from the menu bar **Source | Generate Constructor from Superclass...**
7. The "Generate Constructor from Superclass" dialog opens. Click **Finish**
8. To get Eclipse to generate an overloaded constructor, select from the menu bar **Source | Generate Constructor using Fields...**
9. The "Generate Constructor using Fields" dialog opens. Select **weight** and **height** for the fields to initialize then click **OK**
10. To get Eclipse to generate the getters and setters, select from the menu bar **Source | Generate Getters and Setters**
11. The "Generate Getters and Setters" dialog opens. Select the option to generate getters and setters for the **weight** field and the **height** field then click **OK**
12. Properties of a class can be generated from the IDE but not the methods of a class. From the Java Editor for the BMI.java class write the code to implement the **bmi** method using the BMI formula shown on Figure 2.
13. From the Java Editor for the BMI.java class write the code to implement the **bmiCategory** method using the table shown on Figure 3. You will need to call the bmi() to get the BMI to calculate the BMI category.
14. Now that you finished creating your BMI class how do you know if it is working as expected?

ChineseZodiac JavaBean

In the Chinese calendar, every year is associated with a particular animal. The 12-year animal cycle is rat, ox, tiger, rabbit, dragon, snake, horse, goat (or ram), monkey, rooster, dog, and pig (or boar). The year 1900 is a year of the rat; thus 1901 is a year of the ox and 1912 is another year of the rat. If you know in what year person was born, you can compute the offset from 1900 and determine the animal associated with that person's year of birth.

In this part of the exercise, you will create a Java class to determine the animal associated with a person's year of birth. The class diagram for the ChineseZodiac class is shown on Figure 4. The animal associated with the offset from the year 1900 is shown in Figure 5.



Figure 4 - ChineseZodiac Class Diagram

Offset of 12 from 1900	Animal
0	Rat
1	Ox
2	Tiger
3	Rabbit
4	Dragon
5	Snake
6	Horse
7	Goat
8	Monkey
9	Rooster
10	Dog

11	Pig
----	-----

Figure 5 - Animal for Offset of 12 from 1900

Instructions

1. In the Java package `ca.nait.dmit.domain`, create a new Java class named **ChineseZodiac**
2. Write the code to implement the **static** method **animal**. The animal can be calculated by subtracting 1900 from the birth year and then calculate the offset from 12. The offset from a number is the remainder from dividing by a number. You can use the modulo operator (%) to get the remainder from a division. The animal for the offset is shown on Figure 5. You can use an if statement or an switch statement or an string array to determine the animal using the offset.

javadoc

You will now add doc comments to your Java source code and use the javadoc command to generate HTML documentation for your source code.

Instructions

1. Open **BMI.java** and add the following javadoc comments before the class declaration.

```
/**
 * This class is use to calculate a person's body mass index (BMI) and their BMI Category.
 * @author yourFirstName yourLastName
 * @version 2015.01.16
 */
```

2. Add the following javadoc comments before the **bmi** method declaration

```
/**
 * Calculate the body mass index (BMI) using the weight and height of the person.
 * The BMI of a person is calculated using the formula: BMI = 700 * weight / (height * height)
 * where weight is in pounds and height is in inches.
 * @return the body mass index (BMI) value of the person
 */
```

3. Add the following javadoc comments before the **bmiCategory** method declaration

```
/**
 * Determines the BMI Category of the person using their BMI value and
 * comparing it against the following table.
 * <table>
 *   <thead>
 *     <tr>
 *       <th>BMI Range</th>
 *       <th>BMI Category</th>
 *     </tr>
 *   </thead>
 *   <tbody>
 *     <tr>
 *       <td>< 18.5</td>
 *       <td>underweight</td>
 *     </tr>
 *     <tr>
 *       <td>>= 18.5 and < 25</td>
 *       <td>normal</td>
 *     </tr>
 *     <tr>
 *       <td>>= 25 and < 30</td>
 *       <td>overweight</td>
 *     </tr>
 *   </tbody>
 * </table>
 */
```

```

*          </tr>
*          <tr>
*                  <td>>= 30</td>
*                  <td>obese</td>
*          </tr>
*      </tbody>
* </table>
*
* @return one of following: underweight, normal, overweight, obese.
*/

```

4. Open ChineseZodiac.java and add javadoc comments before the class declaration with a description of the class, the author of the class, and the current date (in yyyy.mm.dd format) of the version of the class.
5. In ChineseZodiac.java, add the javadoc comments before the **animal** method declaration with a description of the method.
6. From the menu bar select **File | Export..**
7. The "Export" dialog opens. Select **Java | Javadoc** then click **Next**
8. The "Generate Javadoc" dialog opens. In the "Select the type for which Java will be generated" panel, select only **dmit2015-exercises-yourname | src/main/java | ca.nait.dmit.domain**
9. For the "Use standard doclet" field, click on **Browse...** and navigate to **~/workspace/dmit2015-exercises-YourName/src/main/webapp**. Click **Create Folder** and assign **doc** for the new folder. Click **OK** to close the browse dialog.
10. Click **Finish** to generate javadoc HTML API documentation. On the "Update Javadoc Location" dialog, click **Yes To All**
11. From Project Explorer view, navigate to **dmit2015-exercises-yourname | Deployed Resources | webapp | doc**. Expand the doc folder and right-mouse click on **index.html** and from the context menu select **Open With ==> Web Browser**. The web browser opens with your index.html page. Click on the class name links on the web browser to see the comments you added to the class.