Assignment 2

oosc@swc.rwth-aachen.de

Issued: 08.11.2021   Submission: 22.11.2021   Discussion: 25.11.2021

IMPORTANT NOTE: A template maven project is provided on Moodle (together with the other assignment documents) for answering the programming tasks in 2.3 until 2.7. Please place the implementation classes under the corresponding Maven sub-modules. If needed, please feel free to extend the pom files.
Upon submitting, please upload your maven project as a zip file.

2.1. Assignment: Inheritance

Given is the following class hierarchy in Java:

```java
class Material {
    private void writeName() {
        System.out.println("Material");
    }
} // class

class Aurochs extends Material {
    public void writeName () {
        System.out.println("Aurochs");
    }
    public static void writeLatin() {
        System.out.print("Bos primigenius");
    }
    public void writeDescription() {
        this.writeName();
    }
} // class

class Cow extends Aurochs {
    public void writeName () {
        System.out.println("Cow");
    }
    public static void writeLatin() {
        Aurochs.writeLatin();
        System.out.println("taurus");
    }
} // class
```

a) Depict the structure modeled by the source code by means of UML.

b) Imagine the following variables are given:

```java
Cow cow = new Cow();
Aurochs aurochs = cow;
Material material = new Material();
```

Please, go through the statements in the table below and describe the output. In case there is an error, please explain it in more detail and weather this is a compile or a runtime error. Do not forget to include the reason for the error.

Note: Try to solve this task without implementing it! The learning effect of copy-pasting stdout is small. ;-)

Statement	Output or Description of Failure
`cow.writeName();`	
`aurochs.writeName();`	
`((Material)aurochs).writeName();`	
`((Aurochs)cow).writeName();`	
`((Cow)aurochs).writeName();`	
`aurochs.writeLatin();`	
`((Aurochs)material).writeName();`	

2.2. Assignment: Inheritance and Associations

The following UML class diagram is given:
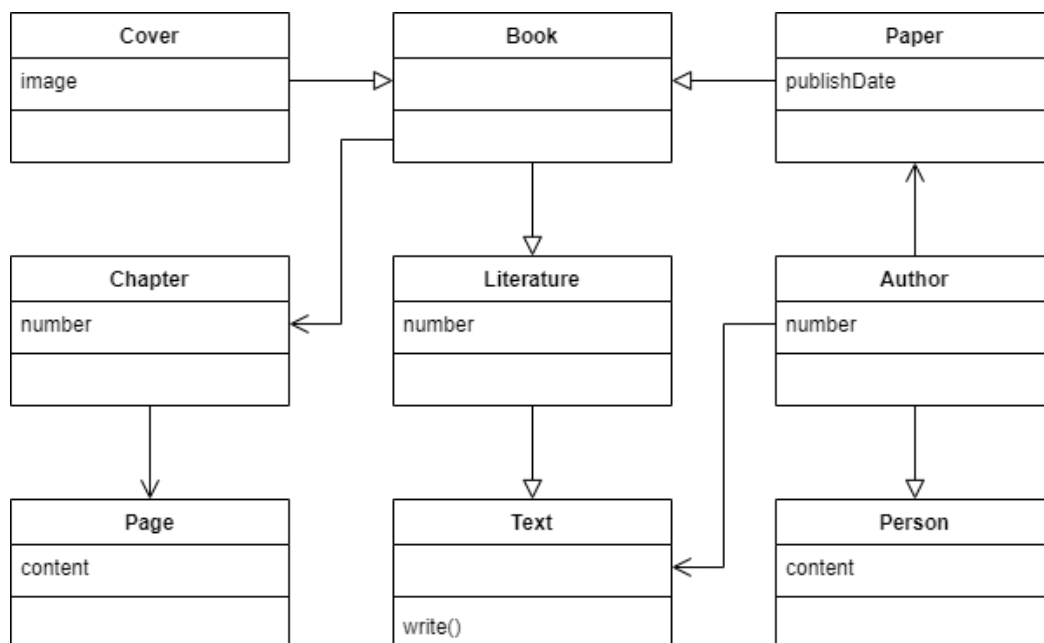
a) Why can this class diagram be considered as poorly modeled? Are the inheritance and association relationships chosen well? Justify your opinion.

b) Restructure the model to improve its quality. Please provide the reasons for why you chose to apply these specific transformations.

2.3. Assignment: Meyer's inheritance taxonomy
The source code below depicts another class hierarchy. Look into the details of this hierarchy and identify kinds of Meyer's inheritance taxonomy; again explain your rationales. Moreover, change the code to a more reasonably class structure; explain your rationales.
NOTE: Please use the template Maven project to provide your implementation for this task.

```java
public class GeoObject {
    public String color;
    public Integer identifier;

    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    public Integer getIdentifier() {
        return identifier;
    }

    public void setIdentifier(Integer identifier) {
        this.identifier = (identifier>0) ? identifier : new Integer(0);
    }
}

public class Circle extends GeoObject {
    public Integer diameter;

    public Integer getDiameter() {
        return diameter;
    }

    public void setDiameter(Integer diameter) {
        this.diameter = (diameter > 0) ? diameter : new Integer(0);
    }
}

public class Ellipse extends GeoObject {
    public Integer diameter;
    public Integer conjugateDiameter;

    public Integer getDiameter() {
        return diameter;
    }

    public Integer getConjugateDiameter() {
        return conjugateDiameter;
    }

    public void setDiameter(Integer diameter) {
        this.diameter = (diameter > 0) ? diameter : new Integer(0);

    }

    public void setConjugateDiameter(Integer conjugateDiameter) {
        this.conjugateDiameter = (conjugateDiameter > 0) ?
                            conjugateDiameter : new Integer(0);
    }
}
```

2.4. Assignment: View inheritance

Try to find an example of view inheritance. Draw a sketch of your example as an UML class diagram. Write the source code as well. Show how your example works by implementing JUnit tests or a small Java application.

NOTE: Please use the template Maven project to provide your implementation for this task.

2.5. Assignment: Implementation inheritance

Draw and implement an example of implementation inheritance. Explain why this is implementation inheritance and no other type of Meyer Inheritance. Finally, discuss how to avoid implementation inheritance in your example and in general. Show how your example works by implementing JUnit tests or a small Java application.

NOTE: Please use the template Maven project to provide your implementation for this task.

2.6. Assignment: Variation inheritance

Try to find an example of functional variation inheritance. Draw a sketch of your example as an UML class diagram. Write the source code as well. Show how your example works by implementing JUnit tests or a small Java application.

NOTE: Please use the template Maven project to provide your implementation for this task.

2.7. Assignment: Polymorphism

This task is inspired by a question asked in many job interviews at big IT companies nowadays. The question will proof a wide spectrum of important programming skills: recursion, trees, OO and polymorphism.

In this task you will want to model and implement the classes to build the expression tree from an arithmetic expression (given as a string) with the operators "+", "-", "*", and "/". An example for the input is given by the string "(2 + 2) * 2".

Note: **In all following tasks you are not required to write a parser. Just if you want to. ;-)**

a) Create a UML class diagram of classes that allow to construct arithmetic expressions.

   Hint 1: Consider recursion/trees and polymorphism when modelling the classes.
   Hint 2: As an example, consider the similar problem of representing Boolean expressions. Given is the following expression:
   $$(true \land (\neg true \land false)) \lor (false \land \neg true)$$
   The corresponding expression tree is depicted below (~ is used for $\neg$):
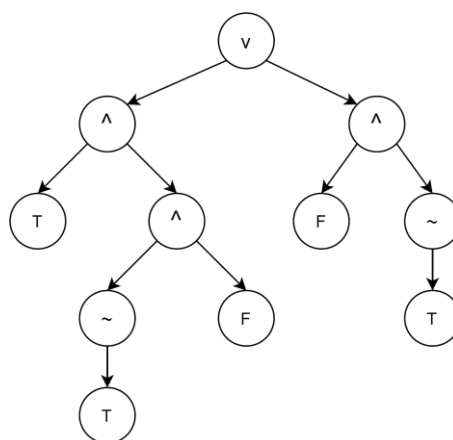
b) Implement the modelled classes in Java.

c) Provide Java code for the construction of the arithmetic expression ((2*(5+10))-9)/3 using the classes you designed.

d) Show with Java code how the constructed arithmetic expression ((2*(5+10))-9)/3 can be evaluated, i.e. how the result 7 can be obtained.

e) Extend your implementation to support the modulo operator: %

f) Was it difficult to extend your implementation? Provide a brief argumentation, why this was or was not the case.

Please provide your complete Eclipse or IntelliJ project with all source files and necessary dependencies.
NOTE: Please use the template Maven project to provide your implementation for this task.

2.8. Assignment: Meyer's inheritance taxonomy
The following class diagram, depicting inherit relationships between classes, is given.

Classify all inheritance relationships according to Meyer's inheritance taxonomy. Since there may be several reasonable classifications for a single relationship, please explain your assumptions and the reasons why you chose a particular classification.
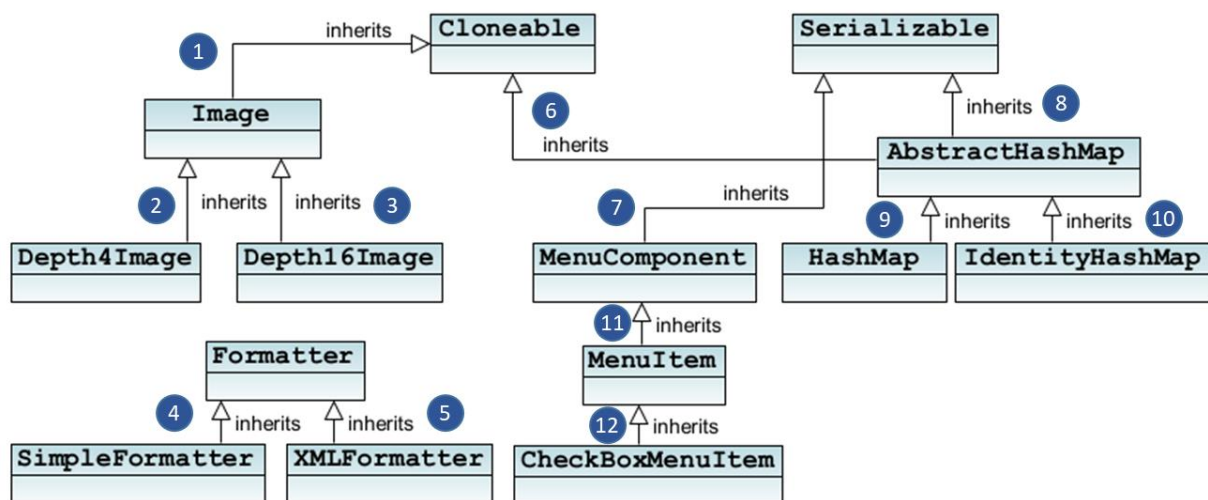
Hints:

Please, your results shall be handed in as a compressed file containing all necessary files and data named „**GroupX_AssignmentX_OOSC2021.zip**"; replace the **X** with your group identifier. Your submission should be submitted via Moodle Learning Room.

Please include the following materials in your submission:
- all documents as PDF
- all UML diagrams as images (jpg, png) and included in the PDF
- source code / dependencies in a zip file. Please use the template maven project provided on Moodle.