**Home exercise 7**

**Course name: Object oriented programming and design for engineering**

**Course number: 157109**

**Subject: Design Patterns - Visitor, Decorator, Composite**

**This assignment contains two unrelated parts. Below you can find an execution output that complements the requirements.**

**Part 1: Art painting system**

employing the Composite and Visitor design patterns. Your task involves constructing a tree structure that represents a painting. The structure will be constructed by reading data from an input file, which contains information about the components that constitute the painting you are to create. Each row in the input file corresponds to the details of a single element. For instance, consider the contents of the "elements.txt" file as an example:

island name: Madagascar, diameter: 8  
lake name: Grand, diameter: 15  
Grand/boat material: Timber, width: 10, length: 10  
Grand/flag Color: Red, poleHeight: 5, width: 3, length: 3  
Grand/island name: Glover, diameter: 5  
Grand/Glover/tree leaves: 2500, width: 12, height: 15  
Grand/Glover/kid birth year: 2013, hair color: Black, width: 12, height: 15  
Grand/Glover/kite color: Red, width: 12, height: 15

We learn that the painting consists of two elements: the first is an island, and the second is a lake. The lake encompasses a boat, a flag, and another island within it. This latter island, in turn, comprises three elements: a tree, a kid, and a kite.

The Painting type serves as the top-level element in the hierarchy, containing a collection of elements. A Painting instance can include any other element, but no element can contain a Painting instance, including lake, island, or the painting itself.

Use the Composite pattern to represent the painting structure. You may use any fitting data structure such as an array or a set to implement the pattern.

The element types that need to be created are kite, boat, flag, tree, kid, lake, island.

Here are some details on the initialization and structure of each element:

Each element has attributes for both length and width. It's important to note that an element's shape may not necessarily be rectangular. Therefore, calculating its area would require an understanding of the specific shapes it comprises (find the details below). Every element is classified as one of three types: terrestrial (capable of residing solely on an island), aquatic (suitable for placement only on a lake), or amphibious (capable of residing on either island or lake). In the example given above, the "flag" is categorized as terrestrial. Therefore, when creating the corresponding painting, an appropriate error message will be generated, and the flag will not be added to the lake in practice.

kite (terrestrial) – the kite has an attribute that indicates its color. The shape of the kite is Dalton.

boat (aquatic) – the boat has an attribute indicating the material from which it is made (Material). The shape of the boat is a rectangle adjacent to a semicircle. The diameter of the circle is equal to the width of the rectangle.

flag (terrestrial) – the flag has attributes that represent its color and the height of its mast. The shape of the flag is a rectangle.

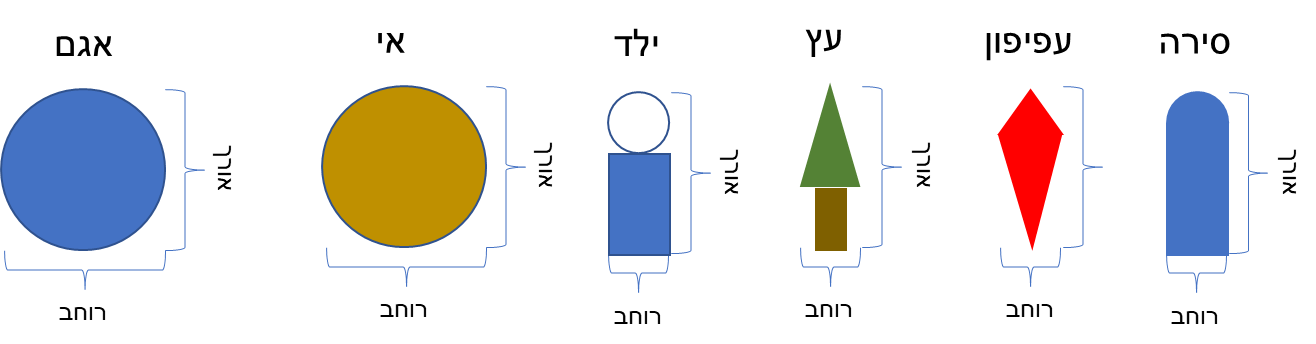
tree (terrestrial) – the tree has an attribute that represents the amount of its leaves. The shape of the tree is a rectangle adjacent to a triangle. The width of the tree is the width of the base of the triangle. The width of the rectangle is equal to half the width of the base of the triangle. The height of the triangle/rectangle separately is not given (the area can be calculated even without this information)

kid (amphibious) – the kid has attributes that represent his hair color and year of birth. The child's shape is a rectangle adjacent to a circle. The diameter of the circle is equal to the width of the rectangle.

lake (terrestrial) – a composite element, meaning it can contain other elements. The lake can only contain aquatic/amphibious elements. The shape of the lake is round (the width and length are equal to the diameter)

Island (marine) – a composite element, meaning it can contain other elements. The island can only contain terrestrial/amphibious elements. The shape of the island is round (the width and length are equal to the diameter)

Below is an illustration of the shape of the various elements in the painting:



Community Verified icon

**In addition, all the above constructors receive the size of the file in bytes as well as its path and name.**

After the data structure construction, the following operations should be available:

1. **elements count** – calculate the total number of elements.
2. **Area calculator** – print the sum of areas of all elements, including the composite elements. The sum should be rounded to the nearest integer
3. **Short print –** print all names of the elements in the painting in pre-order.
4. **Long print –** print a detailed report about the painting in pre-order (see examples below)

Each of the above-mentioned operations should be implemented in a separate Visitor. This means that the classes representing the actual elements will not contain the implementation details, only the accept function enabling the Visitor to operate as required.

Draw a class diagram for the system and attach it to the submission file.

**General notes**

You must use Visitor and Composite design patterns.

The way in which you construct the system is up to you!

You are provided with the class ElementDetailsFactory, creating a simple/composite element given a String. In addition, you are provided with the file elements.txt for your tests. Do not change these files and do not submit them.

You are also provided with partial implementations of the classes representing simple/composite elements. You should complete these files for supporting Visitor and Composite. You should also implement the Visitor classes and submit them.

Note: beside the files that are explicitly required by the automatic checker submission system you will need to submit additional classes, required for fulfillment of the assignment requirements (Using the “any file” option).

**Part 2 (short): Document decoration system**

In this part you will implement a system for composing the structure of a document using the Decorator and Factory design patterns.

Below are the classes which implement the different document types. **Do not change the given classes.** There are 4 different types of documents. The client must choose the type of desired documents (one of the options). After that the client can add various attributes as he sees fit (in terms of type, quantity and order) to the selected document:

Below are the markings of the various document types (all in lowercase):

* ac – academic paper
* cn – contract
* jr – journal article
* bk – book

Below are the markings of the various document attributes (all in lowercase):

* tb – table
* eq – equation
* d – diagram
* nt – note

The document construction will be done using a Factory Class.

The document attributes should be presented as the output in accordance with the order they were mentioned.

Draw a class diagram for the system and attach it to the submission file.

**General notes**

You must use Decorator and Factory design patterns.

You are provided with the classes representing basic documents. Do not change these files and do not submit them.

You are also provided with a skeleton of PaperFactory. You should complete this file. You should also implement the classes for the additional attributes (using the decorator pattern) and the factory creating them and submit them. You are provided with a partial implementation of the Main class. You should complete this class so that it activates the Composite, Visitor and Decorator patterns.

The automatic checking program is an integral part of the requirements. You can understand the requirements better using the automatic checker and the attached running sample below.

**Good luck!**

Assume that the input file contains:

island name: Madagascar, diameter: 8  
lake name: Grand, diameter: 15  
Grand/boat material: Timber, width: 10, length: 10  
Grand/flag Color: Red, poleHeight: 5, width: 3, length: 3  
Grand/island name: Glover, diameter: 5  
Grand/Glover/tree leaves: 2500, width: 12, height: 15  
Grand/Glover/kid birth year: 2013, hair color: Black, width: 12, height: 15  
Grand/Glover/kite color: Red, width: 12, height: 15

**Running example 1:**

Choose from the following options:

a: Art

p: Papers

a

Enter the path of the painting description

elements.txt

Grand cannot contain flag

Choose from the following options:

q: quit

c: count elements

lp: long print

sh: short print

ta: total area

c

7

lp

An empty island named Madagascar. A lake named Grand containing: A boat made of timber material. An island named Glover containing: A tree with an amount of 2500 leaves. A 10 year old kid with black hair. A kite of color: red.

sh

Madagascar

Grand

Grand/boat

Grand/Glover

Grand/Glover/tree

Grand/Glover/kid

Grand/Glover/kite

ta

665

q

**Running example 2:**

Choose from the following options:

a: Art

p: Papers

p

Choose from the following paper:

ac: academic paper

cn: contract

jr: journal article

bk: book

ac

Choose from the following options:

a: add element

s: submit

a

Choose from the following elements:

tb: table

eq: equation

d: diagram

nt: note

tb

Choose from the following options:

a: add element

s: submit

a

Choose from the following elements:

tb: table

eq: equation

d: diagram

nt: note

d

Choose from the following options:

a: add element

s: submit

a

Choose from the following elements:

tb: table

eq: equation

d: diagram

nt: note

eq

Choose from the following options:

a: add element

s: submit

a

Choose from the following elements:

tb: table

eq: equation

d: diagram

nt: note

nt

Choose from the following options:

a: add element

s: submit

a

Choose from the following elements:

tb: table

eq: equation

d: diagram

nt: note

d

Choose from the following options:

a: add element

s: submit

a

Choose from the following elements:

tb: table

eq: equation

d: diagram

nt: note

eq

Choose from the following options:

a: add element

s: submit

s

Academic Paper Table Diagram Equation Note Diagram Equation