P09 Art Gallery

Overview

This assignment involves the implementation of an online art gallery using Binary Search Trees (BST). An art gallery is a place in which visual arts are displayed. It represents a meeting point for art lovers where one can explore and sometimes purchase artwork by emerging, established or master artists.

Our BST will store a set of visual artwork objects. Art lovers can visit and explore the art gallery and lookup for a specific artwork. They can also purchase a specific artwork. The lookup key information for the retrieval or search operation combines the year of creation of the artwork and its name. You are going to learn how BSTs can be used to facilitate insertion and retrieval operations to and from a collection of ordered elements in an easy and elegant way.

Grading Rubric

5 points	**Pre-Assignment Quiz:** The P09 pre-assignment quiz is accessible through Canvas before having access to this specification by **11:59PM on Monday 04/25/2022**.
15 points	**Immediate Automated Tests:** Upon submission of your assignment to Gradescope, you will receive feedback from automated grading tests about whether specific parts of your submission conform to this write-up specification. If these tests detect problems in your code, they will attempt to give you some feedback about the kind of defect that they noticed. Note that passing all of these tests does NOT mean your program is otherwise correct. To become more confident of this, you should run additional tests of your own.
20 points	**Additional Automated Tests:** When your manual grading feedback appears on Gradescope, you will also see the feedback from these additional automated grading tests. These tests are similar to the Immediate Automated Tests, but may test different parts of your submission in different ways.
10 points	**Manual Grading Feedback:** After the deadline for an assignment has passed, the course staff will begin manually grading your submission with respect to the commenting and style requirements defined in the CS300 Course Style Guide. This grading usually takes about a week from the hard deadline, after which you will find feedback on Gradescope.

Learning Objectives

The goals of this assignment include:

- Implement common Binary Search Tree (BST) operations.
- Further developing your experience in recursive problem-solving.
- Improve your experience in developing unit tests.

Assignment Requirements and Notes

- Pair programming is **ALLOWED** for this assignment. If you decide to work with a partner on this assignment, REGISTER your partnership NO LATER than **11:59PM on Monday 04/25/2022** and MAKE SURE that you have read and understood the CS300 Pair Programming Policy(CS300 Pair Programming Policy (L1)). Your partner must be enrolled in one of the cs300 spring 2022 lectures (L1, L2, L3, or L4).
- The ONLY import statements that you may include in your classes are the ones related to relevant exceptions.
- You CAN define local variables that you may need to implement the methods defined in this program.
- You CAN define private methods to help implement the different public methods defined in this write-up, if needed.
- You MUST NOT add any additional fields either instance or static to your program, and any **public** methods either static or instance to your program, other than those defined in this write-up.
- Any source code provided in this specification may be included verbatim in your program without attribution.
- All the String comparisons in this assignment should be CASE-SENSITIVE.
- ALL your test methods MUST be implemented in your `ArtGalleryTester` class.
- In addition to the required test methods, we HIGHLY recommend (not require) that you develop your additional own unit tests (**public static methods that return a boolean**).
- Ensure that your code for every assignment is styled in conformance to CS300 Course Style Guide(CS300 Course Style Guide (L1)).

- You can submit your work in progress multiple times on gradescope. Your submission may include methods not implemented or with partial implementation or with a default return statement. But avoid submitting a code which does not compile. A submission which contains compile errors won't pass any of the automated tests on gradescope.
- You MUST adhere to the Academic Conduct Expectations and Advice(Academic Conduct Expectations and Advice (L1)).

1 Getting Started

1. To get started, let's first create a new Java11 project within Eclipse. You can name this project whatever you like, but "P09 Art Gallery" is a descriptive choice. You have to ensure that your new project uses Java 11, by setting the "Use an execution environment JRE:" drop down setting to "JavaSE-11" within the new Java Project dialog box. Do not create a module associated to your java project. All your source java files must be included within the default src package of your p09 project.
2. Then, download the four following source files and add them to the src folder of your project.
   - Artwork.java,
   - BSTNode.java,
   - ArtGallery.java,
   - ArtGalleryTester.java
3. You ARE NOT going to make any changes to the implementation details of the `BSTNode.java` and you WILL NOT submit it to gradescope.
4. Read carefully through the implementation details of the provided `BSTNode` class. Notice that only a getter method is defined for the `data` data field of a binary search tree node. No setter is defined. This means that once a BSTNode object is created and assigned a value for its data field, there is no way to change that data.
5. You are going to implement the missing code in the provided three other source files with respect to the comments provided in their javadocs. A few details are provided in the next sections of this write-up.

2 Artwork class

The `Artwork` class models artwork objects stored in our artwork gallery.

- Read carefully through the provided source codes details.
- Every `Artwork` object is defined by three instance data fields (the name of the artwork, the year of creation of the artwork, and its cost).
- the `Artwork` class implements the `java.util.Comparable` interface. Two Artwork objects are compared with respect to their three instance fields.
- Notice that the constructor of the class `Artwork` does not allow the creation of Artwork objects finished before the year 1000.
- Implement both `compareTo()` and `equals()` method.
- Two Artwork objects are first compared with respect to their years of creation. The oldest is the smallest. If there is a tie of years, the less expensive one is the smallest. If there is still a tie in terms of costs, comparison decision will be made based on the alphabetical order of their names. You can rely on the String.compareTo() method to make this decision.
- Make sure to add your name to the @author annotation in the javadoc style method headers of the `equals` and `compareTo` methods in the `Artwork` class. All the other methods have been completely implemented by our TA whose name is in the javadoc style class header.
- Implement the tester method `testArtworkCompareToEquals()` defined in the `ArtGalleryTester` class to ensure the correctness of your implementation.
- Keep in mind that your tester method should be able to detect defects in any broken implementation not necessarily yours.
- Recall that you MUST add a complete file header to all your submitted files to gradescope.

3 ArtGallery and ArtGalleryTester classes

The class `ArtGallery` models an artwork gallery implemented as a binary search tree. Read carefully through the provided implementation details in both `ArtGallery` and `ArtGalleryTester` classes.

- You are going to complete the implementation of all the methods defined in these classes and including the `TODO` tag with respect to the details provided in their javadoc method headers.

- **Make sure to remove the TODO tags once you complete the implementation of each method.**
- Notice that we added default return statements to the incomplete methods to simply let the code compile.
- Duplicate Artworks with exactly the same names, years and costs are not allowed in this application.
- In our `ArtGallery` binary search tree, the increasing order of artworks goes from the smallest to the greatest artwork with accordance to the comparison result returned by `Artwork.compareTo` method. The compareTo method returns negative, zero, or positive integers. It does not specify specific positive or negative values.
- Recall that you ARE NOT allowed to add any additional fields either instance or static fields to the `ArtGallery` class.
- NO additional public methods must be added to the `ArtGallery` class.
- If a method is described to be a *recursive helper* method, you are not allowed to implement it using iteration (for or while loops). It MUST be designed and implemented using recursion.
- Make sure to update the size of the art gallery each time an artwork is successfully added or removed to or from the gallery. Avoid updating the size in the helper methods.
- Make sure to add implementation level comments highlighting the major steps in the complex argorithms to implement.
- You are responsible for testing thoroughly your implementation of the `ArtGallery` public methods. Implement all the test methods defined in the `ArtGalleryTester` class. Make sure to test every method with at least 3 different scenarios. Hints are provided in the provided javadocs method headers.
- You can refer to the following online gallery if you are looking for examples of artworks to use in your tester scenarios. You can also use fake artwork names, years, and costs based on your imagination.

Illustrative Example

In order to provide you with a better understanding on how to use the implemented classes, demo.txt contains an example of source code. You can find in the following its expected output. Do not use this source code to test your methods. Rely on the implementation of your own test scenarios.

```
Size: 0 Height: 0
Gallery:

==============================================================
Size: 2 Height: 2
Gallery:
[(Name: Starry Night, Van Gogh) (Year: 1889) (Cost: $2000.0)]
[(Name: Guernica, Picasso) (Year: 1937) (Cost: $3000.0)]
==============================================================
Size: 6 Height: 4
Gallery:
[(Name: Mona Lisa, DaVinci) (Year: 1503) (Cost: $1000.0)]
[(Name: Whistler, Abbott) (Year: 1871) (Cost: $5000.0)]
[(Name: Starry Night, Van Gogh) (Year: 1889) (Cost: $2000.0)]
[(Name: Guernica, Picasso) (Year: 1937) (Cost: $3000.0)]
[(Name: NightHawks, Hopper) (Year: 1942) (Cost: $4000.0)]
[(Name: Amazone, Tsalapatanis) (Year: 2021) (Cost: $6080.0)]
==============================================================
Size: 11 Height: 6
Gallery:
[(Name: Mona Lisa, DaVinci) (Year: 1503) (Cost: $1000.0)]
[(Name: Whistler, Abbott) (Year: 1871) (Cost: $5000.0)]
[(Name: Starry Night, Van Gogh) (Year: 1889) (Cost: $2000.0)]
[(Name: Gothic, Wood) (Year: 1930) (Cost: $6000.0)]
[(Name: Persistence of Memory, Dali) (Year: 1931) (Cost: $7000.0)]
[(Name: Guernica, Picasso) (Year: 1937) (Cost: $3000.0)]
[(Name: NightHawks, Hopper) (Year: 1942) (Cost: $4000.0)]
[(Name: Der Schrei, Silber) (Year: 2019) (Cost: $12160.0)]
[(Name: For gourmets, Tuzhilkina) (Year: 2021) (Cost: $1280.0)]
[(Name: Cantabrico, Torices) (Year: 2021) (Cost: $3870.0)]
[(Name: Amazone, Tsalapatanis) (Year: 2021) (Cost: $6080.0)]
This gallery contains (Mona Lisa, DaVinci, 1503, 1000): true
This gallery contains (Whistler, Abbott, 1871, 5000): true

This gallery contains (Chaplin, Brainwash", 2020, 9090): false

Best (greatest) artwork: [(Name: Amazone, Tsalapatanis) (Year: 2021) (Cost: $6080.0)]

Lookup query: search for the artworks of 2021 whose costs do not exceed $5000.00:
[[(Name: For gourmets, Tuzhilkina) (Year: 2021) (Cost: $1280.0)],
[(Name: Cantabrico, Torices) (Year: 2021) (Cost: $3870.0)]]

Lookup query: search for the artworks of 2021 whose costs do not exceed $10000.00:
```

```
[[(Name: Amazone, Tsalapatanis) (Year: 2021) (Cost: $6080.0)],
[(Name: For gourmets, Tuzhilkina) (Year: 2021) (Cost: $1280.0)],
[(Name: Cantabrico, Torices) (Year: 2021) (Cost: $3870.0)]]

Buy "Der Schrei, Silber", 2019, 12160:
Size: 10 Height: 5
Gallery:
[(Name: Mona Lisa, DaVinci) (Year: 1503) (Cost: $1000.0)]
[(Name: Whistler, Abbott) (Year: 1871) (Cost: $5000.0)]
[(Name: Starry Night, Van Gogh) (Year: 1889) (Cost: $2000.0)]
[(Name: Gothic, Wood) (Year: 1930) (Cost: $6000.0)]
[(Name: Persistence of Memory, Dali) (Year: 1931) (Cost: $7000.0)]
[(Name: Guernica, Picasso) (Year: 1937) (Cost: $3000.0)]
[(Name: NightHawks, Hopper) (Year: 1942) (Cost: $4000.0)]
[(Name: For gourmets, Tuzhilkina) (Year: 2021) (Cost: $1280.0)]
[(Name: Cantabrico, Torices) (Year: 2021) (Cost: $3870.0)]
[(Name: Amazone, Tsalapatanis) (Year: 2021) (Cost: $6080.0)]

Buy "Mona Lisa, DaVinci", 1503, 1000:
Size: 9 Height: 5
Gallery:
[(Name: Whistler, Abbott) (Year: 1871) (Cost: $5000.0)]
[(Name: Starry Night, Van Gogh) (Year: 1889) (Cost: $2000.0)]
[(Name: Gothic, Wood) (Year: 1930) (Cost: $6000.0)]
[(Name: Persistence of Memory, Dali) (Year: 1931) (Cost: $7000.0)]
[(Name: Guernica, Picasso) (Year: 1937) (Cost: $3000.0)]
[(Name: NightHawks, Hopper) (Year: 1942) (Cost: $4000.0)]
[(Name: For gourmets, Tuzhilkina) (Year: 2021) (Cost: $1280.0)]
[(Name: Cantabrico, Torices) (Year: 2021) (Cost: $3870.0)]
[(Name: Amazone, Tsalapatanis) (Year: 2021) (Cost: $6080.0)]
Buy "Guernica, Picasso", 1937, 3000:
Size: 8 Height: 4
Gallery:
[(Name: Whistler, Abbott) (Year: 1871) (Cost: $5000.0)]
[(Name: Starry Night, Van Gogh) (Year: 1889) (Cost: $2000.0)]
[(Name: Gothic, Wood) (Year: 1930) (Cost: $6000.0)]
[(Name: Persistence of Memory, Dali) (Year: 1931) (Cost: $7000.0)]
[(Name: NightHawks, Hopper) (Year: 1942) (Cost: $4000.0)]
[(Name: For gourmets, Tuzhilkina) (Year: 2021) (Cost: $1280.0)]
[(Name: Cantabrico, Torices) (Year: 2021) (Cost: $3870.0)]
[(Name: Amazone, Tsalapatanis) (Year: 2021) (Cost: $6080.0)]
Buy "Mona Lisa, DaVinci", 1503, 1000:
[(Name: Mona Lisa, DaVinci) (Year: 1503) (Cost: $1000.0)] not found in this gallery.
```

4 Assignment Submission

Congratulations on finishing this CS300 assignment! After verifying that your work is correct, and written clearly in a style that is consistent with the CS300 Course Style Guide, you should submit your final work through Gradescope. The only THREE files that you must submit is `Artwork.java`, `ArtGallery.java`, and `ArtGalleryTester.java`. Your score for this assignment will be based on your **"active"** submission made prior to the hard deadline of **9:59PM on April 28**th. The second portion of your grade for this assignment will be determined by running that same submission against additional offline automated grading tests after the submission deadline.