Due Date: 15.06.2023, 23:55

CENG112 - Data Structures
Homework 4: Traversals that use an iterator

This homework will cover the topics given below:

- Binary Search Tree
- Binary Tree
- Stack
- Queue
- Iterator

Please read the whole document carefully.

In this homework, you will explore the concept of traversing a binary search tree using different methods. You will use a family tree as an example to understand how to traverse a binary search tree using preorder, postorder, and level order methods.

To provide more flexibility to the client, it is recommended to define the traversals in the form of iterators instead of just displaying data during the traversal. This approach allows the client to control when each visit takes place and perform additional operations on the data retrieved from each node.

In Java, the interface Iterator provides hasNext and next methods, which allow the client to retrieve data from the current node in the traversal at any time. This means that the client can retrieve a node's data, perform operations on it, and then retrieve data from the next node in the iteration. In this homework,

- You are given the required interfaces and classes for the binary search tree, node, stack, and queue in the project **CENG112_HW4.zip.**
- The class **BinaryTree** implements the methods in the interface TreeIteratorInterface. You are expected to implement the necessary methods of TreeIteratorInterface in **BinaryTree class.**
- You can find the **InorderIterator** class as an example of how to implement an iterator for inorder traversal. You are expected to implement **PreorderIterator**, **PostorderIterator** and **LevelOrderIterator** classes given in the homework project.
- The expected output is given on the next page. You should get this output when you run Main.java class.
- You can compare your iterators' output and/or functionality using recursive implementations of preorder and postorder traversal methods given in BinaryTree class.
- You can import the project by following the path File->Import->Existing Projects into Workspace->Select archive file->Browse-> CENG112_HW4.zip in Eclipse.
- Please, update the project name by using your group ID. For example, CENG112_HW4_G05.

Expected output: (the comments in RED are for guiding you, do not expect them in output)
"INORDER
(Emily 1830) (Lily 1840) (David 1845) (Jane 1860) (Sarah 1870) (Olivia 1880) (Jack 1890) (Peter 1900) (Sophie 1910) (Ethan 1915) (Chloe 1920) (Mia 1930) (Noah 1935) (John 1945) (Mary 1960) (Liam 1970) (Rachel 1978) (Tom 1985) (Alex 1987) (Kate 1995) (Ben 2005) (Zoe 2015) (Ava 2022) → bst.inorderTraverse();

(Emily 1830)(Lily 1840)(David 1845)(Jane 1860)(Sarah 1870)(Olivia 1880)(Jack 1890)(Peter 1900)(Sophie 1910)(Ethan 1915)(Chloe 1920)(Mia 1930)(Noah 1935)(John 1945)(Mary 1960)(Liam 1970)(Rachel 1978)(Tom 1985)(Alex 1987)(Kate 1995)(Ben 2005)(Zoe 2015)(Ava 2022) → *printWithIterator*(inorderIterator);

PREORDER
(John 1945) (Noah 1935) (Jane 1860) (David 1845) (Emily 1830) (Lily 1840) (Peter 1900) (Sarah 1870) (Jack 1890) (Olivia 1880) (Sophie 1910) (Chloe 1920) (Ethan 1915) (Mia 1930) (Mary 1960) (Tom 1985) (Rachel 1978) (Liam 1970) (Kate 1995) (Alex 1987) (Ben 2005) (Zoe 2015) (Ava 2022) → bst.preorderTraverse();

(John 1945)(Noah 1935)(Jane 1860)(David 1845)(Emily 1830)(Lily 1840)(Peter 1900)(Sarah 1870)(Jack 1890)(Olivia 1880)(Sophie 1910)(Chloe 1920)(Ethan 1915)(Mia 1930)(Mary 1960)(Tom 1985)(Rachel 1978)(Liam 1970)(Kate 1995)(Alex 1987)(Ben 2005)(Zoe 2015)(Ava 2022) → *printWithIterator*(preorderIterator);

POSTORDER
(Lily 1840) (Emily 1830) (David 1845) (Olivia 1880) (Jack 1890) (Sarah 1870) (Ethan 1915) (Mia 1930) (Chloe 1920) (Sophie 1910) (Peter 1900) (Jane 1860) (Noah 1935) (Liam 1970) (Rachel 1978) (Alex 1987) (Ava 2022) (Zoe 2015) (Ben 2005) (Kate 1995) (Tom 1985) (Mary 1960) (John 1945) → bst.postorderTraverse();

(Lily 1840)(Emily 1830)(David 1845)(Olivia 1880)(Jack 1890)(Sarah 1870)(Ethan 1915)(Mia 1930)(Chloe 1920)(Sophie 1910)(Peter 1900)(Jane 1860)(Noah 1935)(Liam 1970)(Rachel 1978)(Alex 1987)(Ava 2022)(Zoe 2015)(Ben 2005)(Kate 1995)(Tom 1985)(Mary 1960) (John 1945) → *printWithIterator*(postorderIterator);

LEVELORDER
(John 1945)(Noah 1935)(Mary 1960)(Jane 1860)(Tom 1985)(David 1845)(Peter 1900)(Rachel 1978)(Kate 1995)(Emily 1830)(Sarah 1870)(Sophie 1910)(Liam 1970)(Alex 1987)(Ben 2005)(Lily 1840)(Jack 1890)(Chloe 1920)(Zoe 2015)(Olivia 1880)(Ethan 1915)(Mia 1930)(Ava 2022) → *printWithIterator*(levelorderIterator);

- **Please submit the project as a whole, not only three classes you've implemented.**

- This is a 2-person group assignment. However, inter-group collaboration is not allowed!
- All assignments are subject to plagiarism detection and the suspected solutions (derived from or inspired by the solution of other groups) will be graded as zero. **Since only three classes will be graded we recommend you to be more cautious, no excuses will be accepted!**
- It is not allowed to use Java Collections Framework.
- Your code should be easy to read and test: **Keep your code clean**. **Avoid duplication and redundancy. Follow Java Naming Conventions**. Use ***relative paths*** instead of absolute ones. 🔗

Submission Rules

All submissions must:

- be performed via **Microsoft Teams** by only one of the group members,
- be exported as an Eclipse Project and saved in ZIP format,
- follow a specific naming convention such that CENG112_HW4_*groupID*. Submissions that do not comply with the rules above are penalized.

Eclipse Project: CENG112_HW4_*G05*
Exported Archive File: CENG112_HW4_*G05*.zip