Programming Assignment 2

Student Management System

Problem Description:

In *Emperor Penguin University (EPU)*, every student record contains the following information:

- First Name
- Surname
- Gender
- ID
- Age

Among them, the ID is a unique identifier of the student. The complete student list is stored in "studentList.txt". Please build a student management system for EPU which:

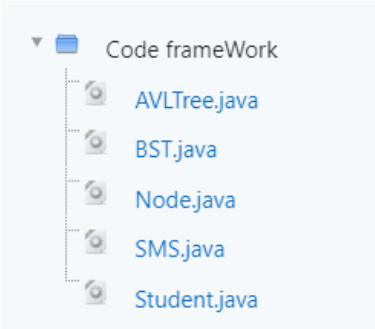
- Stores all the student records on an AVL tree.
    1. There are in total 31537 records
    2. Please define a proper class to describe a student record
- Accepts query commands and prints a message for a certain student.
    1. The command for a query is of format: "q"+<ID>. For example, if the user inputs "q29001", the system should print a message about the student whose ID is 29001.
    2. Follows describes the format of the message.
        a) If the user inputs "q29001", the system should print "Olivia SWANSON (29001) is a female student of age 20."
        b) If the user inputs "q26870", the system should print "Zyn STARK (26870) is a male student of age 21."
        c) If the user inputs "q40000", the system should print "40000 is not found."
- Accepts unenrollment command and removes a certain student from the tree.
    1. The command for unenrollment is "u" + <ID>. For example, if the user inputs "u29001", the system should remove the student record whose ID is 29001.
    2. A message is printed after unenrollment.
        a) If the user inputs "u29001", the system should print "Olivia SWANSON (29001) is unenrolled from EPU."
        b) If the user inputs "40000", the system should print "40000 is not found."
    3. You don't have to write the information back into "studentList.txt".
- Accepts the exit command and exits the program. The exit command is "e".

You need to submit five files on iSpace:

- Student.java: the class for a student record.
- Node.java: the class for an AVL tree node which stores one *Student* object.
- BST.java: the class for a BST where every node is a *Node* object. It is the super class of *AVLTree*.
- AVLTree.java: the class for an AVL tree where every node is a *Node* object. It is a subclass of *BST*.
- SMS.java: the class for the "student management system" which stores the student records using an AVL tree. It contains a main function which runs the main procedure.

Note:

- The code framework (Code framework.zip attached) has already been provided to you. You just need to fill in the code with the framework, and then upload these 5 .java file with changing their names. Doing so will help you smoothly and quickly pass the autograder's compilation. （代码框架已经提供给你，你只需要在框架上面填写好代码即可，无需更改文件的名称，直接上传这5个java文件即可。这么做可以顺利帮你迅速通过autograder的编译。）



-
- Make sure your code is written in Java.
- Make sure that an AVL tree is used for SMS, and every node of the tree stores one student record.
- You may assume that the .txt file is in the same directory as your program does.
- You may assume that the user command is always valid.