CENG 202 Data Structure

Lab Assignment 2 (20/03/2021)

Due date: (23/03/2021 at 23.55 on AYBUZEM)

Algorithm Analysis

You are given a **Product** class last week and you implemented **ProductArray** and **TestProductArray** classes.

This week, **Products.txt** and **ProductList.txt** files are given to you. You need to add or update some functions inside the **ProductArray** class:

- Add **sortedInsert** function to insert items in alphabetically sorted according to the attribute **name**
- Add **bubbleSort** function to sort your array according to the attribute **name**
- Add **insertionSort** function to sort your array according to the attribute **name**
- Add **selectionSort** function to sort your array according to the attribute **name**

  (**Hint**: you can use **compareTo()** function in java for comparing strings)

- You can update **search** function to return the index of found item
- You need to make sure your **delete** function does not make your list unsorted.

- Finally, define a global variable for **ProductArray** class named **sortCost** which starts from **0.** This variable will calculate the cost of sort functions as shown in second page.

Here you have a bubble sort algorithm for integer arrays. Your **sortCost** variable will count your **loop** count and **operation with the array that you sort**. Just like the bubble sort you will use **sortCost** variables in **insertionSort, selectionSort** and **sortedInsert** functions.

```
void bubbleSort(int arr[])
{
    int n = arr.length;
    for (int i = 0; i < n-1; i++) {
        for (int j = 0; j < n-i-1; j++){
            sortCost++;  // Counts the loop
            if (arr[j] > arr[j+1])
            {
                // swap arr[j+1] and arr[j]
                int temp = arr[j];
                sortCost++;  // Counts the operation of the array
                arr[j] = arr[j+1];
                sortCost++; // Counts the operation of the array
                arr[j+1] = temp;
                sortCost++;  // Counts the operation of the array
            }
        }
     }
 }
```

In your **TestProductArray** class:

- You will read Products from txt files. The attributes are seperated by comma "," (Hint: you can use **split** function of strings)
- You will create **4** different array object. (**30000** length)
- When you are reading products from the file you will insert same item for the first 3 array **unsorted** and sort them with **bubbleSort, insertionSort** and **selectionSort** respectively. For the last array you will use **sortedInsert** function and finally you will display the results as shown in next page:

It is **suggested** to work with the **smaller** file first and after you make sure that everything works correctly, you can try the other one.

```
Output for Products.txt file:
BUBBLE SORT
Total Sort Operations: 501
-----------------------
INSERTION SORT
Total Sort Operations: 234
-----------------------
SELECTION SORT
Total Sort Operations: 270
-----------------------
SORTED INSERT
Total Sort Operations: 346
```

```
Output for ProductList.txt file:
BUBBLE SORT
Total Sort Operations: 937118715
-----------------------
INSERTION SORT
Total Sort Operations: 374684242
-----------------------
SELECTION SORT
Total Sort Operations: 375256704
-----------------------
SORTED INSERT
Total Sort Operations: 562544027
```

----------------------- -----------------------

Sorted Version of Product File (you can check your solutions):

```
Product name: ASUS X415JA, Description: CORE I5 8GB RAM, Price: 6550.0

Product name: ASUS X515JF, Description: CORE I5, Price: 5699.0

Product name: ASUS X543UA, Description: CORE I3 6100U, Price: 3799.0

Product name: EVERPAD SC-730, Description: PANDA A50, Price: 427.0

Product name: EXPER XCELLERATOR, Description: XC600, Price: 7258.0

Product name: HOMETECH ALFA, Description: 420C INTEL CELERON, Price: 1999.0

Product name: HP 14S-FQ0011NT, Description: AMD ATHLON 3050U, Price: 3299.0

Product name: HP 14S-FQ0014NT, Description: AMD RYZEN, Price: 3699.0

Product name: HP 250 G7, Description: CORE I3, Price: 4959.0

Product name: HUAWEI MATEBOOK D15, Description: AMD RYZEN 7 3700, Price: 6799.0

Product name: IPAD AIR, Description: 64-GB WIFI, Price: 5799.0

Product name: IPAD PRO, Description: 128GB GRAY, Price: 7499.0

Product name: IPAD-32GB 8.NESIL, Description: SPACEGRAY-BLUETOOTH, Price: 3199.0

Product name: LENOVO IDEACENTRE 3, Description: CORE I5, Price: 5999.0

Product name: LENOVO IDEAPAD, Description: FLEX 5 CORE I5, Price: 7899.0

Product name: LENOVO TABM10, Description: MTK HELIO, Price: 1799.0

Product name: MACBOOK AIR, Description: 8GB-256GBSSD, Price: 9899.0

Product name: MSI GE76, Description: RAIDER CORE I7, Price: 27762.0

Product name: MSI GL75, Description: CORE I7, Price: 17725.0

Product name: SAMSUNG GALAXY TAB A7, Description: SM-T500NZAATUR GRI, Price: 1499.0

Product name: SAMSUNG GALAXY TAB S6, Description: LITE SM GRI, Price: 2599.0
```