

Table Component

1) Brief Introduction

Create a class or group of classes that takes the “data.csv” data file (a part of the test) as an input and load the data to the memory, ensure that it is easy to read and sort, and enable it to be exported to HTML. This class or group must enable the data to be exported easily in other formats (e.g., CSV and Excel). The exported output should be in the format of the following table. (Please note that the data in this table does not correspond to the input file.)

Table 1, PC Quarterly Market Share, the Czech Republic, 4Q10

Vendor	Units	Share
IBM	15,000	13.3%
Dell	12,030	10.6%
HP	11,000	9.7%
Others	75,000	66.0%
Total	113,030	100%

2) Description of the Table

- The table illustrates how many computers have been sold by each vendor during a specific quarter.
- The first column, **Vendor**, provides the vendor names.
- The second column, **Units**, provides the number of PCs the respective vendor has sold during the given period.
- The third column, **Share**, provides the percentage of the total number of PCs sold during the quarter for each of the respective vendors.
- The last row contains the total number of PCs sold in second column and always 100% in the third column.
- The input file contains more data than the table requires.

3) Specific Tasks

Create a group of classes that enables another programmer to perform the following easily:

- a) Load the input file and transform it into an object from which you can create a table. You may use the OpenCSV framework for this purpose. See <http://opencsv.sourceforge.net>.
- b) Ascertain the units and share values for a given vendor. Example: I want to know how many units Dell sold during the given quarter and what its percentage share is. Answers: 12,030 and 10.6%.
- c) Ascertain which row contains information about a given vendor. Example: I want to know which row contains information about Dell. Answer: the second row.
- d) Sort the rows alphabetically (by vendor).
- e) Sort the rows by unit values.
- f) Export the object structure to HTML; the format used in Table 1 should be applied.
- g) Export to additional formats (Excel and CSV). You do **NOT** need to implement them. Outline only how you would do so. Create empty classes or methods for these exports, and describe them in JavaDoc.

4) Instructions

- a) This test should **mainly** demonstrate your abilities in terms of **object-oriented design**.
- b) You should not use any technologies other than Java SE and OpenCSV.
- c) Another programmer should be able to understand and reuse your code easily.