

PHMT Developer - Applicant Test

PHM Technology (PHMT) develops advanced engineering analysis software to optimise the design and sustainment of complex engineering systems in mission / safety critical industries – the Maintenance Aware Design environment (MADe).

After investment from Siemens, PHMT is growing its development team in Melbourne in order to keep MADe on the bleeding edge of technology. When you join us you will be working with the 'best of the best' in the engineering world, our clients currently operate in aerospace, automotive and defence globally.

The development team is responsible for developing innovative advanced modelling concepts, mathematical simulations and decision support tools that are integrated with ERP and PLM systems.

The Applicant Test is provided to all candidates who are applying for a development role at PHM Technology. All applicant's must complete and submit the applicant test in a zip format by archiving all Java files and documents required to complete the test and email the archive to the following address:

Cameron Weick - Technical Lead
cpwei@phmtechnology.com

PHM Technology Pty. Ltd.
www.phmtechnology.com

PHMT Developer - Applicant Test

Core Java

For the following requirements, create the required Java class (or classes) to represent the data within an application (You may assume that classes already exist for the Component related information, just focus on the Maintenance Action):

Users require the ability to create Maintenance Activities which are used to represent different types of Maintenance that may be performed on a component. Each Maintenance Action must record:

- 1) Action ID, a unique identifier for the Maintenance Action.
- 2) Name, a descriptive name of the Maintenance Action.
- 3) Maintenance Type, the type of maintenance being carried out (either Repair, Replace, Service or Lubrication).
- 4) Mean time between maintenance (MTBM), the average time (in hours) between the execution of the Maintenance Action.
- 5) Maintenance Action Tasks, a Maintenance Action may contain one or more Tasks which represent the steps required to complete the Maintenance Action.

Each task must include the following variables:

- A) Task ID: A unique (within the Maintenance Action) identifier of the task.
- B) Name: The descriptive name of the task.
- C) Duration: The duration that the task is expected to take.
- D) Personnel Type: The type of personnel required to perform the task.
- E) Hourly rate: The hourly rate charged for the personnel.
- F) Number of personnel: The number of personnel required (on average) for the task.

PHMT Developer - Applicant Test

File IO

1) Write a Java program that reads a text file and prints the contents to the console. Place the following text in a text file for reading:

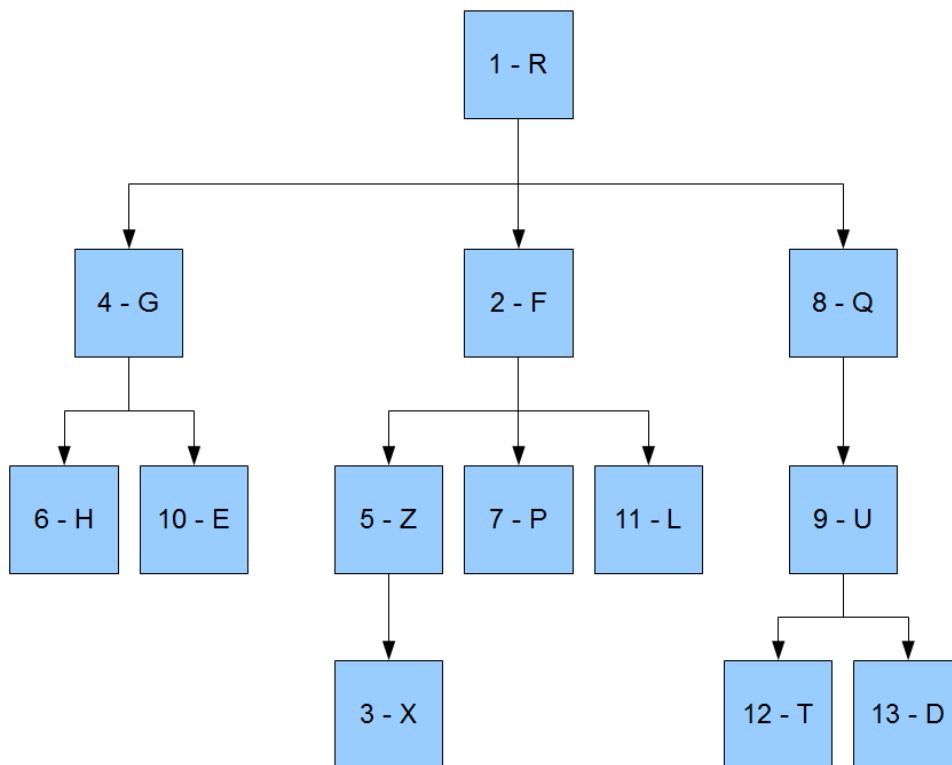
```
A SQL query goes into a bar, walks up to two tables and asks...  
    "Can I join you?"
```

2) Write a Java GUI program that takes a user's text input and writes it to a text file on the disk, the destination where the user saves the file should be user selectable using a GUI component. Once the file has been written provide a user message to tell the user the write process is complete and successful or not.

PHMT Developer - Applicant Test

Create a Tree Data structure

1) Write a Java program that creates the tree data structure drawn below, each tree leaf will contain an integer and a character variable. Using a recursive algorithm write code that will accept a user's input (integer) and output to the console the corresponding letter assigned to that number in the tree (see below).



PHMT Developer - Applicant Test

Math & Problem Solving

1) For the equation shown below, fill in the missing section of the equation where the user has entered 4 input values of: 1, 2, 3 and 4.

$$\begin{array}{r} (1) + (2) + (3) + (4) - \\ (1 \times 2) - (1 \times 3) - (1 \times 4) - (2 \times 3) - (2 \times 4) - (3 \times 4) + \\ (\dots\dots\dots) - \\ (1 \times 2 \times 3 \times 4) \end{array}$$

2) Write a Java program that accepts an input collection of type double with a maximum input size of 1 trillion inputs. Using the equation above as an example input, write an algorithm to formulate the equation using the collection of double inputs and calculate the mathematical result.

Write to the console the calculated result and write to a log file the full expressed equation for validation. Avoid the program running out of memory whilst calculating the result.

Note: The equation uses set theory to perform a union and intersection on the product of the input values.

Example:

Input: 2, 4 and 6.

Output:

$$\begin{array}{r} (2.0) + (4.0) + (6.0) - \\ (2.0 * 4.0) - (2.0 * 6.0) - (4.0 * 6.0) + \\ (2.0 * 4.0 * 6.0) \end{array}$$

Calculated Result: 16

Submit your test result to cpwei@phmtechnology.com, if you have any questions regarding any of the programming sections please address them to Cameron Weick.