Assignment 2, Semester 2: Software Design 4.2 2019

Task: Design & Build a Search application based on Swing using both BST and HashTable

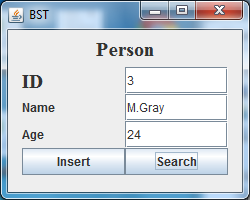
There is no limit to what you can do but it must involve some type of Search/insert application using both techniques.

Can use/modify supplied Search Application Sample Code.

Functionality include,

* Insert into BSTree and Hash Table
* Search using key
* Expect a more complex application and better user interface
* Scope to use any java feature covered on course
* Counts for 30% of final mark
* Ideally should not look like the application below
* Should report on performance of operations (e.g. number of steps for a search)
* Should use Iterator Pattern on Hashtable to allow client to iterate over collection using

(.iterator(), .next() & .hasNext() )



See Example in Moodle.

Preliminary (brief) Spec to be stored in Moodle by 10PM Wednesday

27th February, 2019

Please Submit Writeup & Source Code to Moodle as a zipped file, it must include:

* Brief textual description of completed system
* Key classes and some code snippets of key parts of code
* Some references to performances of BST,Hashtable from literature
* Some references to interator pattern from literature
* Screen Shots
* Conclusions, Description of various problems encountered during development
* Source Code (.java files) must also be stored

Date: Final report and code stored on Moodle - Thursday, 21st March, 2019

Marking Scheme:

1. Initial Specification, including how closely final system matches Spec (10%)
2. Final Specification, how well it’s presented (15%)
3. Good Basic Application Functionality (Class Structure, GUI,

search functionality, iterator) (40%)

1. Performance features included in completed app (15%)
2. Extra Functionality beyond basic requirement. No clear definition but means

Going well beyond sample given.

May include one or more of these

or other features

Complex GUI, complexity of underlying functionality,

complexity class structure (inherit, aggr etc), RMI, Searialization etc (20%)

See Marking Matric attached