Student Name: Calvin Moylan

Student ID: 30018702

Date: 28/08/2020

Assessment Title: Portfolio One

## AT 2: Activity 3

## Design Specifications

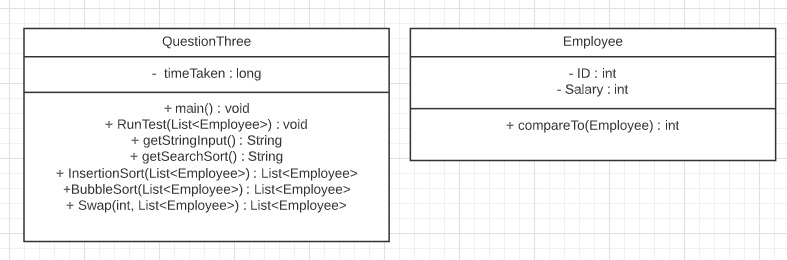
For this question, I was required to make a program that will store employee salaries. The program requires that it have an employee class as well as three different sort methods. The application must sort the employee salaries and then record how long it took to sort them.

## Design Specifications Analysis

I have followed all requirements for the task, which includes being able to store over 100,000 employees with their Salary and ID. There are three different sorting methods, class comparator, bubble sort and insertion sort. The application also comes with a stopwatch to see how long it took to sort the list of employees and their salaries.

The completed application is a success as it follows all task requirements and runs without and errors.

### UML Diagram



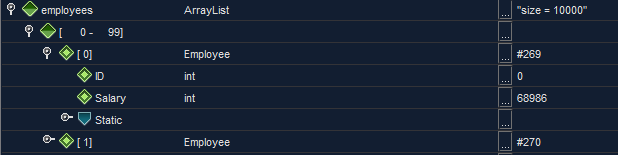
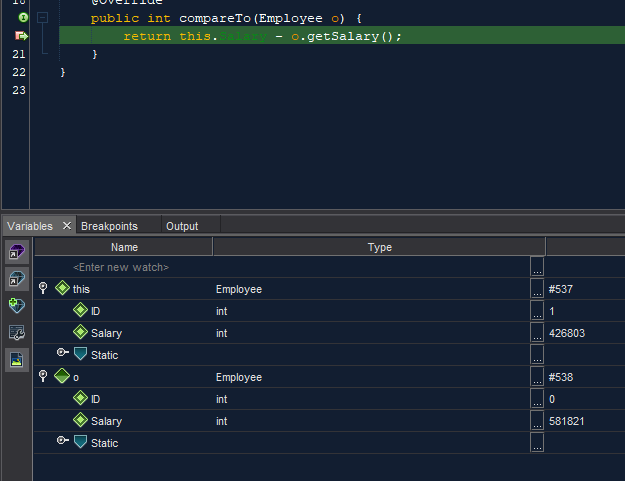
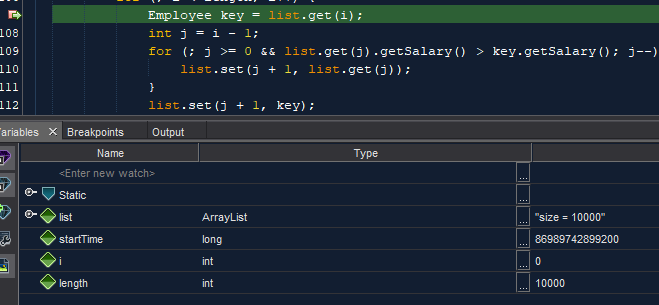
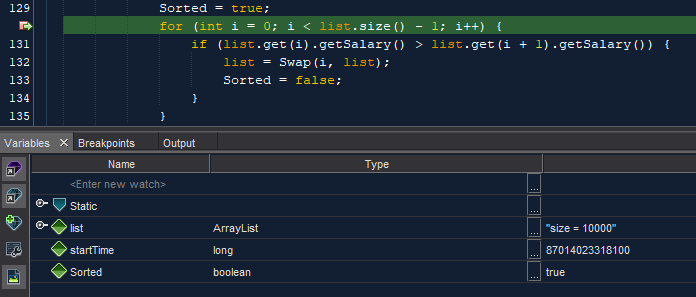
## Sorting Methods

Custom comparator  
**Pros:**   
can only take one line  
very fast  
flexible  
**Cons:**   
not reusable across different classes  
have to write a new comparator for every class

Insertion Sort  
**Pros:**Very easy to implement  
Works well with primitive types  
Simple and easy to understand  
relatively fast speed  
**Cons:**Confusing when working with non-primitive types  
Not very hardware friendly  
Bad with large datasets

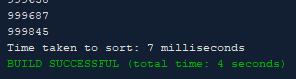
Bubble Sort  
**Pros:**   
Very easy to implement  
Works well with primitive types  
Simple sorting method  
**Cons:**   
Very slow when dealing with large lists  
Doesn’t work well with non-primitive types  
Not very hardware friendly

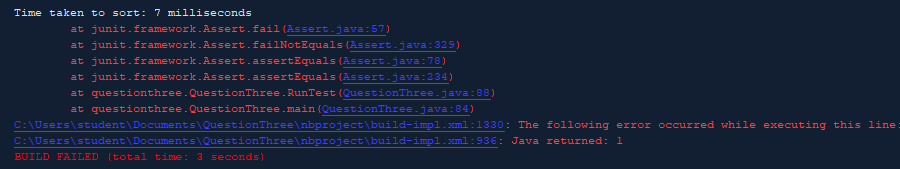
## Debugging

Generating 10,000 employees  
  
Sorting by comparator:  
  
Sorting by insertion:  
  
Sorting by bubble:  


## Testing the Application

### Junit Testing

Successfully sorting 10,000 employees  


Expecting 10,001 employees:  


### Test and validate with sample inputs with screenshots.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Test Case** | **Data** | **Expected Result** | **Actual Result / Comment** |
| *Case 1* | *Adding 10000 employees to list with random salaries* | *Randomly generated list of employees and their salaries* | *10000 employees with random salaries are generated.* | *Results as expected.*  *Ref Figure 1.1* |
| *Case 2* | *Sorting 10000 employees with class comparator* | *Randomly generated list of employees and their salaries* | *10000 employees with random salaries are generated.* | *Results as expected.*  *Ref Figure 1.2* |
| *Case 3* | *Sorting 1000 employees with insertion sort* | *Randomly generated list of employees and their salaries* | *10000 employees with random salaries are generated.* | *Results as expected.*  *Ref Figure 1.3* |
| *Case 4* | *Sorting 1000 employees with bubble sort* | *Randomly generated list of employees and their salaries* | *10000 employees with random salaries are generated.* | *Results as expected.*  *Ref Figure 1.4* |



Figure 1.



Figure 1.2

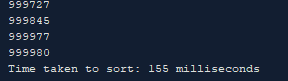


Figure 1.3

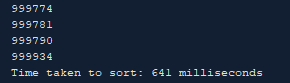


Figure 1.4

END OF TEST TABLE

## Implementation

The application has been uploaded to GitHub under the url: <https://github.com/CalvinMoylanTAFE/JavaThreePortfolio/tree/main/QuestionThree>  
Using GitHub, we will be able to update the repository if any future changes are necessary such as a bug fix or a future feature. The application can be implemented on any computer that supports Java Runtime Environment so deploying the application will require JavaRE to be installed.