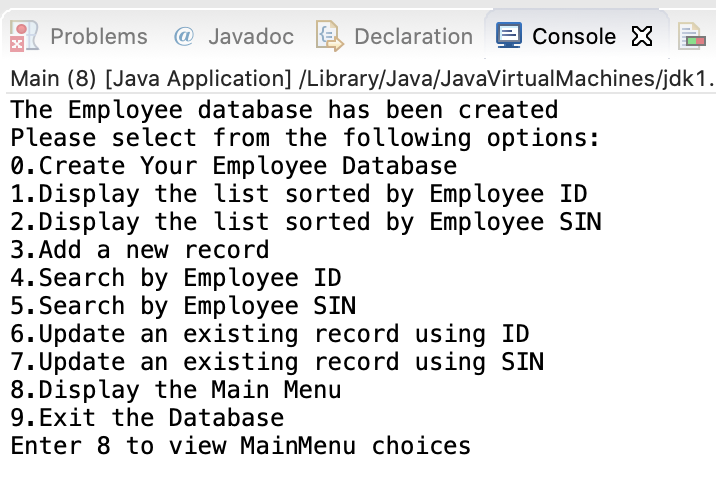
The Main class allows a user to create, sort and update a database of employees. The database includes 2100 randomly generated records. Each record has the following data:

**Employee ID (unique), Employee SIN (unique), Name, Department, Address, Salary**

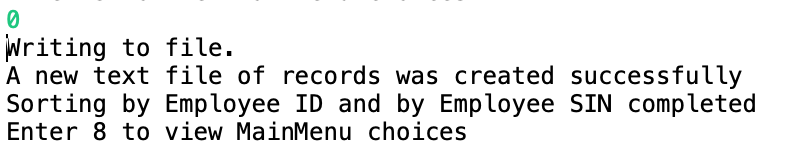
The database is sorted by Employee ID using Heapsort and by Employee SIN using Binary Search Tree data structure.

In the Main class, the user is prompted to select one of the following 10 options. After each selection, the user can go back to the main menu.



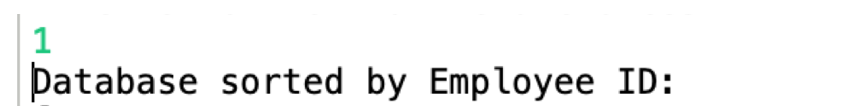
1. **Create Your Employee Database**

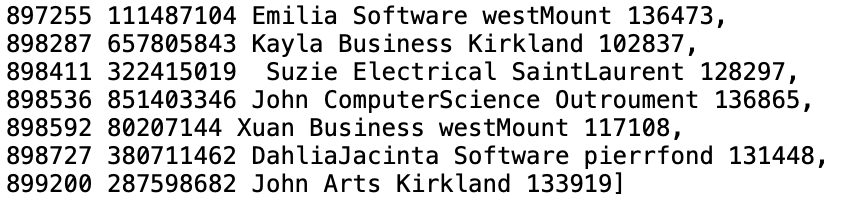
It creates the database using Employee and RandomFile classes and sorts it by Employee ID using HeapSort class and by Employee SIN using BinarySearchTree class.



1. **Display the list sorted by Employee ID**

It displays the records sorted by Employee ID using HeapSort class

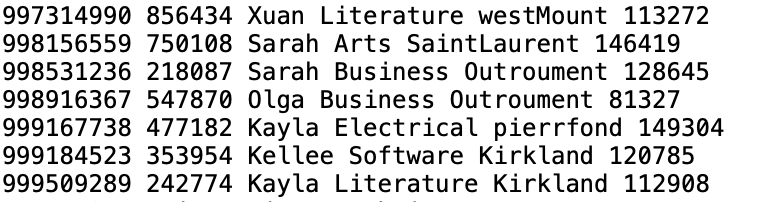
****



1. **Display the list sorted by Employee SIN**

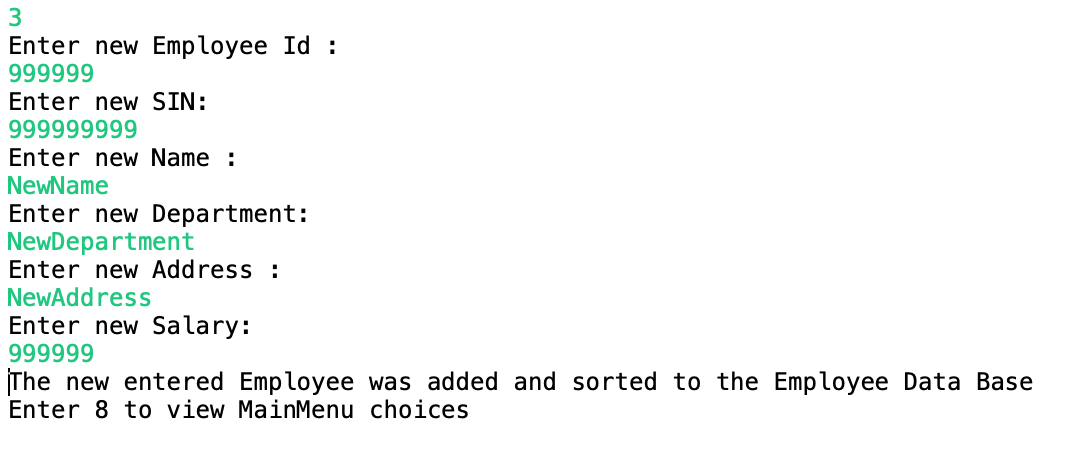
It displays the records sorted by Employee SIN using BinarySearchTree class



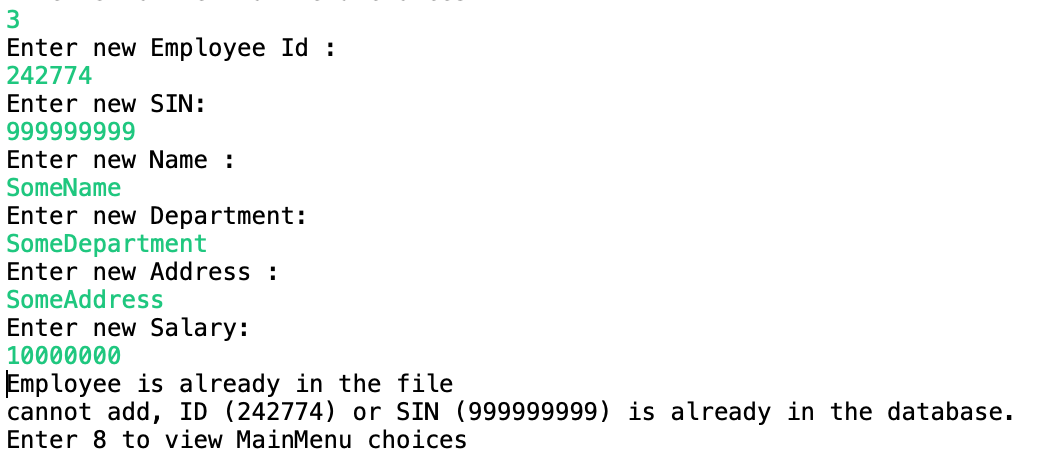


1. **Add a new record**

It adds a record and sorts it in the Database (both using Heapsort and Binary Search Tree); the user can then select options 1 and 2 to see the updated and sorted database.

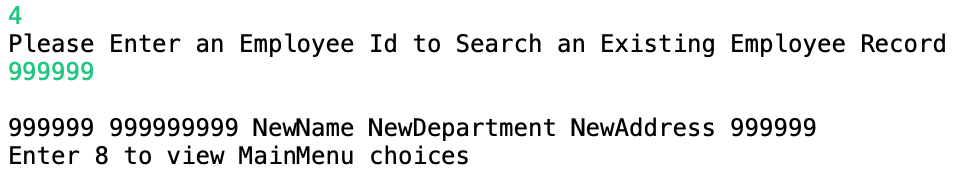


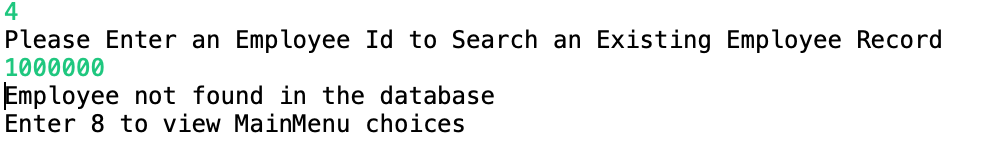
Since ID and SIN are unique, if the new ID or SIN are already in the database, an error message is displayed.



1. **Search by Employee ID**

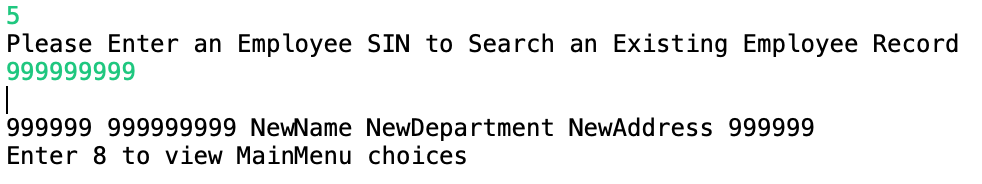
The database is searched by the employee ID; if the record with this ID was found in the database, the full record is displayed; if not, an error message is output.

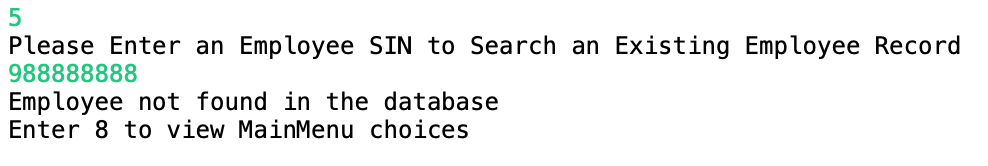




1. **Search by Employee SIN**

The database is searched by the employee SIN; if the record with this SIN was found in the database, the full record is displayed; if not, an error message is output.

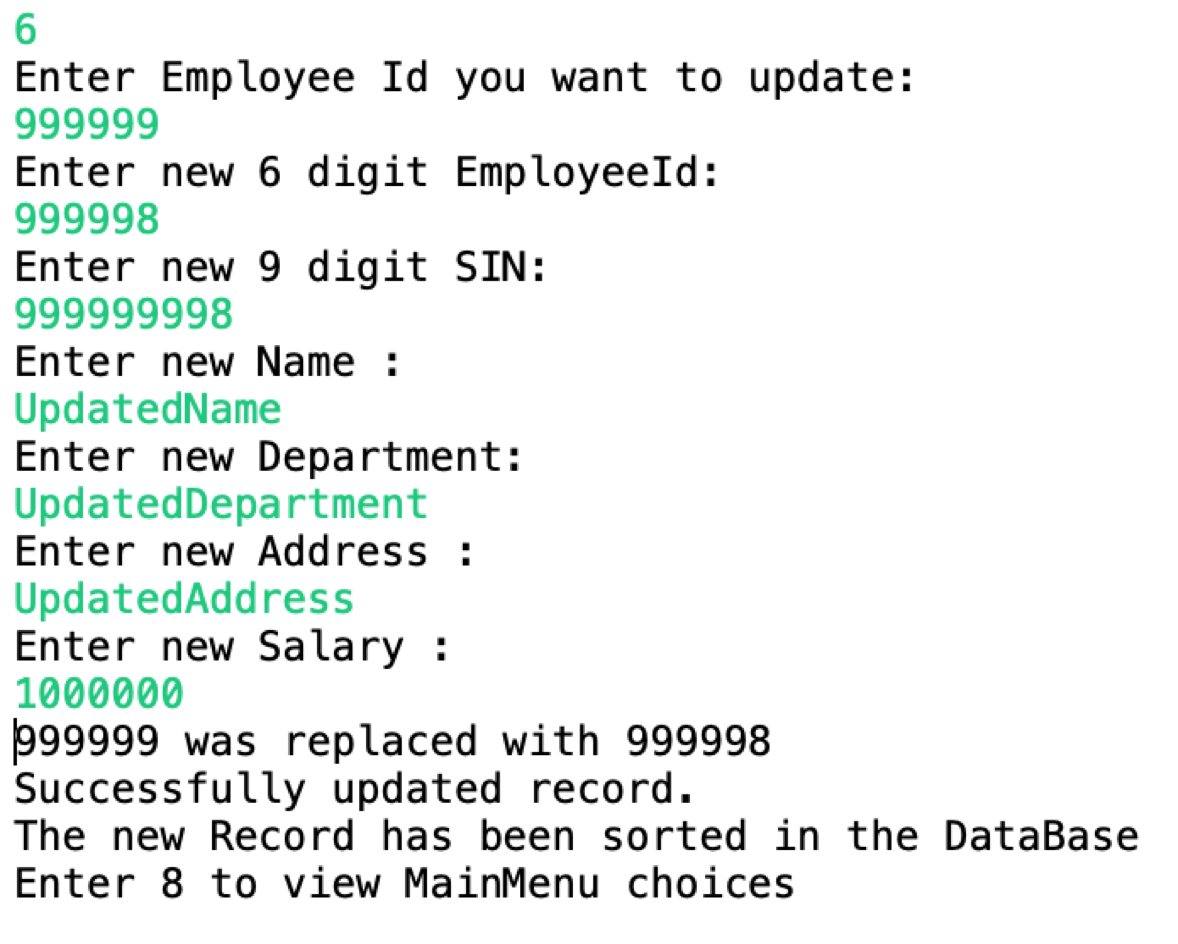




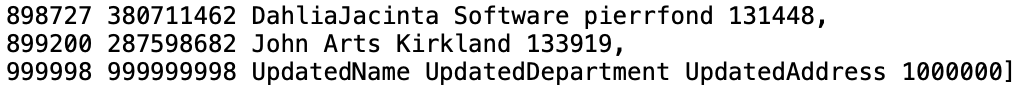
1. **Update an existing record using ID**

The database is searched by the employee ID.

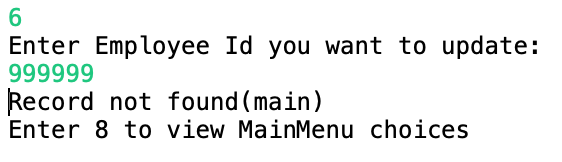
If the record with this ID was found in the database, the user is prompted to enter all the parameters of the new record (new ID, new SIN, new name, new department, new address and new salary).



The new record replaces the old one and is sorted in the database (both using Heapsort and Binary Search Tree); the user can then select options 1 and 2 to see the updated and sorted database.



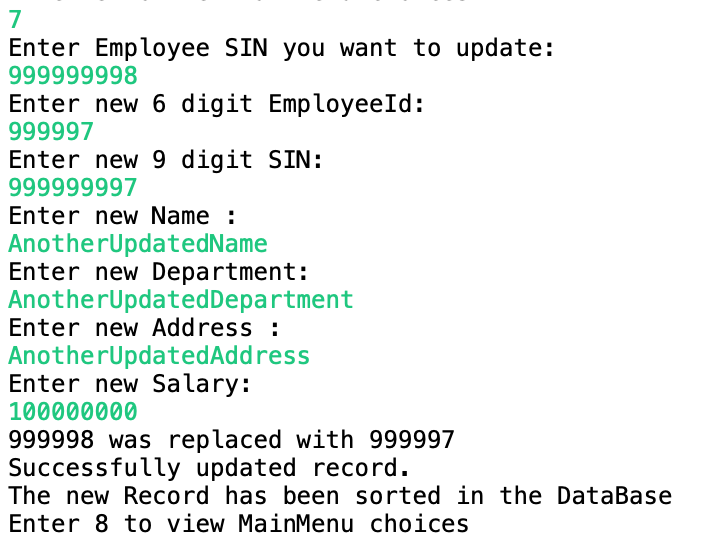
If the record with this ID was not found in the database, an error message is output.



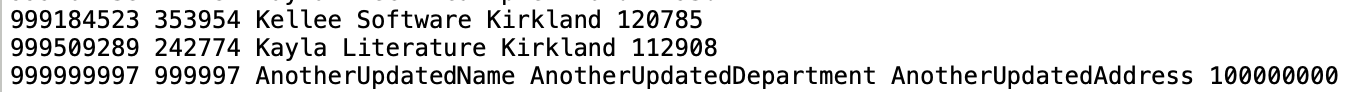
1. **Update an existing record using SIN**

The database is searched by the employee SIN.

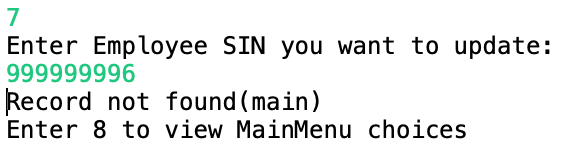
If the record with this SIN was found in the database, the user is prompted to enter all the parameters of the new record (new ID, new SIN, new name, new department, new address and new salary).



The new record replaces the old one and is sorted in the database (both using Heapsort and Binary Search Tree); the user can then select options 1 and 2 to see the updated and sorted database.

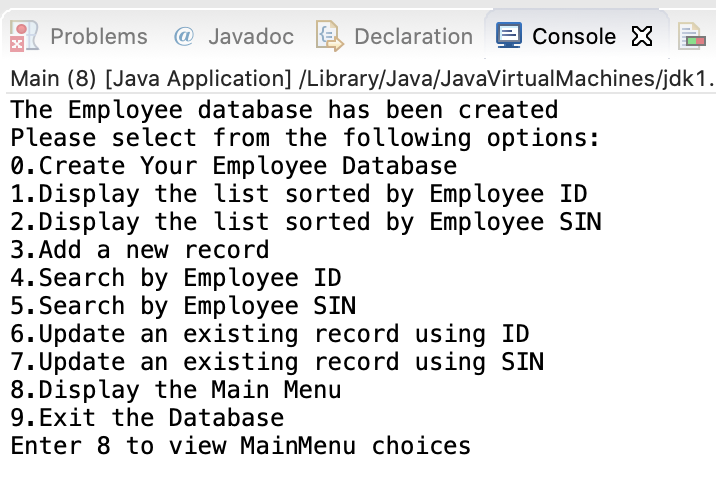


If the record with this ID was not found in the database, an error message is output.



1. **Display the Main Menu**

The following menu is displayed:



1. **Exit the Database**

