Java report for DB assignment

The aim of this DB assignment is to design and implement a database system, and this report including mainly 6 stages, the last one is the final version.

Stage 1: Records

This stage create a Record.java in order to store individual records, I choose to store the record string into an ArrayList, which can be easier add or remove the records.

As for the functions in this Record.java, most of them should be public in order to be used in the next Stages.

The test part of this stage is use the static function claim () instead of assert (), which will throw error when the test fails. And write a main function only for test the result, it will print "Tests pass." When all the test is alright.

This stage is not so complex, only used for store the record string.

Stage 2: Tables

The table stage is used for tables which is the collection of some records. In this stage I add a Table.java which including the insert, delete, update and select function. And also can print the records and dividers to clearly show in the database table:

Insert () function first to check whether the size is enable to add into the ArrayList

Delete () function is to remove the record from the ArrayList after check the index number.

Update () function is to restore the records by removing the old one and adding a new record.

Select () function is to show the select record with table.

This table.java also have the self-test part similar with the record part and can test every basic function of this stage.

In this stage also add a DB.java which only contains main function in order to test and show the result of the tables.

This stage mainly includes the functions based on the string or list level to edit records.

Stage 3: Files&Printing

This stage update the Table.java as the basis of Stage 2, add the function of save and load files and add a animal.txt for test the load record from file to table, (in the latest version I final decided write a new class to achieved the files and printing function).

Table (String fileName) for loads data from the file with its name, and use loadtable() function to get the records data, and can also save in the files by using saveTable() function.

In the final version, the load function achieved by Files.java, in order to easy call different kinds of function instead of just writing all of the functions in only Table.java one class. Files.java has self-test part, with a test file animals.txt saved in the tableFiles folder.

As for the save table, use the saveFile () function to save the records line by line into the files with the asked file name (in the self-test part called test.txt) and finally use the delete function to remove this txt file at the end of test part.

This part of stage aimed on the files level input or output, when the system cannot find the file path or with no records in the Arraylist will throw a NULLPointerException.

Stage 4: Keys

This stage add a key as a unique identify and at first I want to add a new integer statements in Record.java, and also update the Table.java add a autoKey () function to let the int key automatically generated add 1 each time. This autoKey as a unique key in the table and will not be resigned after deleted records. The Key similar like the records id stored in the ArrayList and can also list when print out the whole table.

As for the equals () function added into the Record.java and the Table.java, update it and add check whether those two keys also be equals in order to determine the relations between two tables.

This stage mainly add the unique identify key like ID in the table and record, which help users to insert or delete records without change the key.

Stage 5: Databases

This stage is a general classify and improve the division among those classes, in this stage I began to use Main.java to be the main controller for all the classes instead of the DB.java in previous stages. And as for the DB.java, it used for storage the database, each database can hold a collection of tables. As for the files input and output part in the stage 3, I divided them into two new classes Files.java and Print.java in order to reduce the length of Table.java (which is so long and not easy for me to test or improve). In the DB.java add the getTableRecords (), getTableList() etc functions in order to achieve the operation on the databases level. The depend relationship of those main three parts is DB -> Table ->Record.

Stage 6: FINAL and Extension

This stage is also the final version of this assignment, this part main achieved a simple textual interface like the catalogs function. Basis on the stage 5, add a Catalogs.java to achieved textual interface. This class choose use stack to store the catalogs user command, since the Stack including the push (), pop (), peek () etc. functions which is similar with this catalogs required and the Stack is like grow able array of objects can manage the user command easily. Classify different users input and call the relative functions. The Catalogs.java can call the Table.java and also be called by DB.java and Main.java.

The mainly functions have: load tables with enter the table name, and can also create a new table with input the table name and attribute. Save the table with command save table name, and clear Stack or edit on the level of record etc.

User manual：

“tableName” for load this table with this name;

“create tableAttributes” for create a temp table in the Stack;

“save tableName” for save this table in the database;

“delete tableName” for delete this table from database;

“show tableName” for list the table records without save in the Stack;

“rename colmnName newName” for rename the colmn name;

“add attributesValue” for add new records;

“list” for show all the tables in the current database;

“clear” for clear the Stack value

“quit” for quit from the current level.