Object Oriented Programming.

LAB 3 REPORT

SIMPLE XML DBMS

Prepared by

Omar Ahmed Mohamed

Mostafa Ahmed Abo Shabana

Ali Ramadan El Sayed

Mohamed Said Ibrahem

Contents.

Classes.

COLUMN :

CLASS DESCRIPTION :

This class is used as a setter and a getter for the column :Name , Data type and Elements.
Elements appear as column value in each row.

CLASS METHODS:

```
Column(): it defines the variables and the
data types.
getName(): it returns the column name.
setName(String n): it sets the column name.
getType(): it returns the column type.
setType(String t): it sets the column type.
```

CONTROLLER :

CLASS DESCRIPTION :

This class is used as a link between the parser and the database.
The class connects the user input with the different method which is connected by the databases .

CLASS METHODS:

createDB():it creates new database.

createTable():it creates new table of data.

insertIntoTable():it inserts into the table

printFormatedTable(ArrayList<Column> list):

it is used to format the table data.

selectFromTable():it is used to select the data from the table.

useDataBase():it changes the current used dataBase.

updateTable():it changes data in the table in specific row and column.

drop():it deletes the entire database(directory).

delete():it deletes the database file

main(String[]args):the main function in the program

DTD :

CLASS DESCRIPTION :

This class is used to create the dtd files which make a validation for the xml files .

CLASS METHODS:

this class contains no methods.

DATABASE :

CLASS DESCRIPTION :

This class is used to make the different operations on the database.

CLASS METHODS:

DataBase(): it defines the variables and the data types.

createDB(String dbName):it creates a new database in the given path(directory).

addTable(Idbms t):add the new table to it is database directory.

useDB(String dataBaseName):it changes the current used databases.

useTable(String tableName):it changes the current used tabl.

dropTable():delete the database file.

dropDB():delete the entire database directory.

IDBMS & IDBMS ADAPTER :

DESCRIPTION :

Those two classes are the interfaces for the program.

TE STING DBMS:

CLASS DESCRIPTION :

It is a J-UNIT test for the project .

TABLE :

CLASS DESCRIPTION :

This class is used to make the different operation on the database table such as create, update , load ,insert and drop.

CLASS METHODS:

setName():it sets the table name.

gettColumns():it gets the table name.

settColumns():it sets a table column into the table.

createTable():it creates a new table into the databases.

insert(ArrayList<String> values):it inserts the values inside the table row.

insert(ArrayList<String> columns, ArrayList<String> values):it inserts the value in the chosen raw.

select(ArrayList<String> columnsName,
ArrayList<Integer> rowNumber):
it selects an element for the intersection between a row and a column in the table.

select(ArrayList<String> columnsName):
it selects the entire column from the table.

delete(ArrayList<Integer> rowNumber):it deletes the entire row in the table.

update(ArrayList<String> columnsName,
ArrayList<String> Values):
it updates the entire column in the table by the sent values.

update(ArrayList<String> columnsName,
ArrayList<String> Values,
ArrayList<Integer> rowNumber):
it updates the value from the intersection of a specific column and a specific row.

loadTable(String TName):it reads the XML file and load it to a new database table.

PARSER :

CLASS DESCRIPTION :

This class is used to make the parsing operations for the different user inputs to access or to modify the database.

CLASS METHODS:

initialize():it defines the variables and the data types.

getRowsNumber():it returns the row number.

tableMode():it returns if the table will be saved ot not.

getDataBaseName():it returns the database name.

getTableName():it returns the database table name.

getDataType():it returns the data type.

getValues():it returns the database values.

getTableColumnsName():it returns the database column name.

typeOfQuery():it returns the type of the query.

validate(String inputTxt):it returns if the input syntax valid or not valid.

startState(int startPoint):the state chooses from the different operations to start with the case which is selected by the input.

state0(int startPoint):it is for the select statement so it get * or columns names from -The last state should be select state.

state1(int startPoint, ArrayList<String> list):it saves the datatype which comes after "from" word.

state2(int startPoint, int prevState, boolean saveTable):

state5(int startPoint):it is for the update statement and it checks if "set" is in the statement so it can get to the next state.

state6(int startPoint):it gets the column and value after the set in update.

finalState3(int startPoint):it gets "where" or finish.

finalState4(int startPoint):it gets the condition statement.

getMatchedRow(String columnName, Object value):it gets the row which matches the condition.

compare(char comparisonOP, String element, Object Value):it checks if the condition statement is true or false.

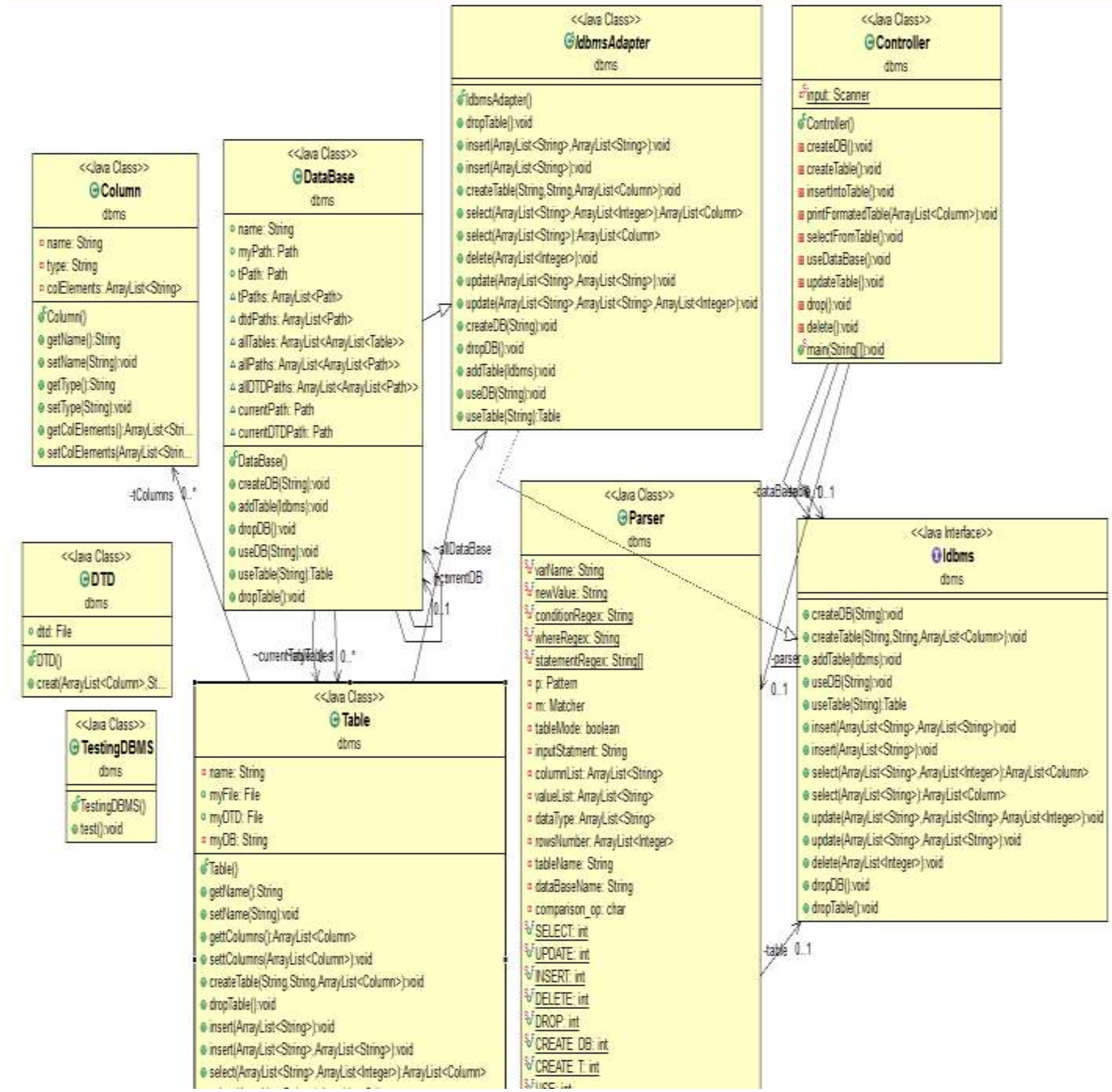
state7(int startPoint, int prevState):it gets "table" or "database" so it can get to the next state.

state8(int startPoint):it is for the create statement ,it takes the table data and finish.

state9(int startPoint, int stringStartPoint):it gets the columns of values.

state10(int stringStartPoint):it gets columns values.

UML diagram.



<<Java Class>>
Column
dbms
name: String
type: String
colElements: ArrayList<String>
Column()
getName():String
setName(String):void
getType():String
setType(String):void
getColElements():ArrayList<Stri...
setColElements(ArrayList<Strin...
<<Java Class>>
DataBase
dbms
name: String
myPath: Path
tPath: Path
tPaths: ArrayList<Path>
dtdPaths: ArrayList<Path>
allTables: ArrayList<ArrayList<Table>>
allPaths: ArrayList<ArrayList<Path>>
allDTDPaths: ArrayList<ArrayList<Path>>
currentPath: Path
currentDTDPath: Path
DataBase()
createDB(String):void
addTable(Idbms):void
dropDB():void
useDB(String):void
useTable(String):Table
dropTable():void
<<Java Class>>
IdbmsAdapter
dbms
IdbmsAdapter()
dropTable():void
insert(ArrayList<String>,ArrayList<String>):void
insert(ArrayList<String>):void
createTable(String,String,ArrayList<Column>):void
select(ArrayList<String>,ArrayList<Integer>):ArrayList<Column>
select(ArrayList<String>):ArrayList<Column>
delete(ArrayList<Integer>):void
update(ArrayList<String>,ArrayList<String>):void
update(ArrayList<String>,ArrayList<String>,ArrayList<Integer>):void
createDB(String):void
dropDB():void
addTable(Idbms):void
useDB(String):void
useTable(String):Table
<<Java Class>>
Controller
dbms
input: Scanner
Controller()
createDB():void
createTable():void
insertIntoTable():void
printFormatedTable(ArrayList<Column>):void
selectFromTable():void
useDataBase():void
updateTable():void
drop():void
delete():void
main(String[]):void
-tColumns 0..*
<<Java Class>>
DTD
dbms
dtd: File
DTD()
creat(ArrayList<Column>,St...
<<Java Class>>
TestingDBMS
dbms
TestingDBMS()
test():void
~allDataBase
currentDB
0..1
~currentTable
allTables 0..*
<<Java Class>>
Table
dbms
name: String
myFile: File
myDTD: File
myDB: String
Table()
getName():String
setName(String):void
gettColumns():ArrayList<Column>
settColumns(ArrayList<Column>):void
createTable(String,String,ArrayList<Column>):void
dropTable():void
insert(ArrayList<String>):void
insert(ArrayList<String>,ArrayList<String>):void
select(ArrayList<String>,ArrayList<Integer>):ArrayList<Column>
<<Java Class>>
Parser
dbms
varName: String
newValue: String
conditionRegex: String
whereRegex: String
statementRegex: String[]
p: Pattern
m: Matcher
tableMode: boolean
inputStatment: String
columnList: ArrayList<String>
valueList: ArrayList<String>
dataType: ArrayList<String>
rowsNumber: ArrayList<Integer>
tableName: String
dataBaseName: String
comparison_op: char
SELECT: int
UPDATE: int
INSERT: int
DELETE: int
DROP: int
CREATE_DB: int
CREATE_T: int
USE: int
-dataBase
-table 0..1
-parser
0..1
-table 0..1
<<Java Interface>>
Idbms
dbms
createDB(String):void
createTable(String,String,ArrayList<Column>):void
addTable(Idbms):void
useDB(String):void
useTable(String):Table
insert(ArrayList<String>,ArrayList<String>):void
insert(ArrayList<String>):void
select(ArrayList<String>,ArrayList<Integer>):ArrayList<Column>
select(ArrayList<String>):ArrayList<Column>
update(ArrayList<String>,ArrayList<String>,ArrayList<Integer>):void
update(ArrayList<String>,ArrayList<String>):void
delete(ArrayList<Integer>):void
dropDB():void
dropTable():void

User guide:

The application takes the user input into the input area such as the sql statement and after it validate the statement it returns the correct value for the input statement