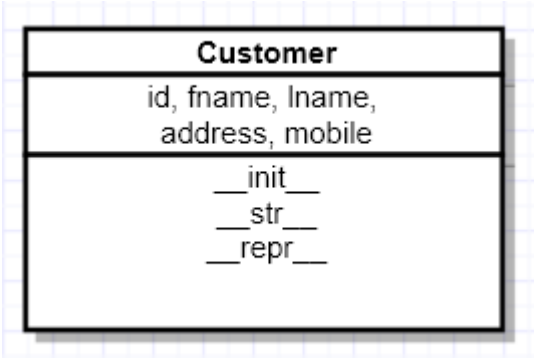
SQLITE Hanuka Project

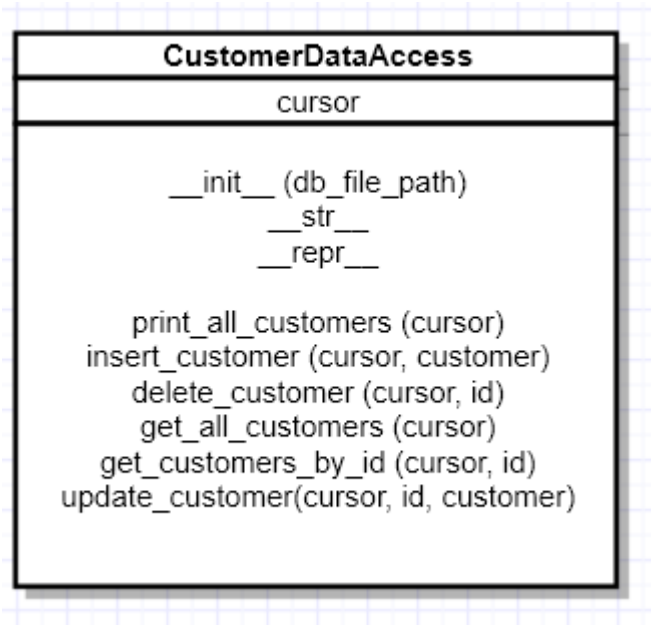
Step 1-



- Create a *Customer* table in ***sqlite***
- Create ***Customer.py*** file with the customer class (of the above UML)
- Create ***program.py*** file which ***imports*** the *Customer* module
- In program.py- create ***def print_all_customers (cursor):*** this function prints all of the customers
- In program.py- create ***def insert_customer (cursor, customer)***: this function inserts the given customer parameter object into the table
- In program.py- create ***def delete_customer (cursor, id)***: this function deleted the customer by its id from the table
- In program.py- create ***def get_all_customers (cursor):*** this function ***returns*** a list of all of the customers in the table (hint: see class code)
- In program.py- create ***def get_customers_by_id (cursor, id):*** this function ***returns*** an instance of the customer which has this id from the table (hint: sql select using where)
- In program.py- create ***def update_customer(cursor, id, customer)***: this function updates the customer details into the table. you need to update the customer in the table that his id is like the parameter (hint: use where), you need to set all of its field using the: *update ... set* sql query (you need the id parameter, and not use the id from the customer object- since you might want to modify the id of the customer)

-- Continue in next page…

Step 2-



- Create ***CustomerDataAccess.py*** file with the CustomerDataAccess class (of the above UML)
- the __init__ method gets as parameter the file path of the db file and creates the cursor as a member.

hint:

```
con = sqlite3.connect(db_file_path)
self.cursor = con.cursor()
```

- copy all the functions from ***program.py*** into **this class** (and make it work)
- now, in ***program.py*** import CustomerDataAccess module

-- Continue in next page…

Step 3-

Display a menu for the user (use print to display the menu)

1. Get all customers
2. Get customer by id
3. Insert customer
4. Delete customer
5. Update customer
6. Exit

Input a selection from the user (user will enter a number between 1-5)

For each option, do the following-

Get all customers- will print all of the customer in the db table

Get customer by id- will input a customer id from the user and then get this customer from the db and print it, if no customer exist with this id- then print f"customer with id: {id} does not exist"

Insert customer- will input all customer details from the user and then insert a customer (with these details) into the db

Delete customer- will input a customer id from the user and then delete the customer from the db

Update customer- will input all customer details from the user + the id (in case the user wants to modify the customer id) and then will update the customer record in the db

*Bonus:

- in the insert method- if customer with the same id already exist, print an error message and do NOT insert
- in the update method, if a customer with the specified id does not exist, print an error message and do NOT update
- in the insert method- if there is already a customer with the same exact first name and last name– print "are you sure?" question before the insertion. and only of the user inputs "yes", then insert the customer into the table

Hag sameah 😊