

19011519-015

19011519-027

19011519-037

Zunair Ahmad

Muhammad Saqib

Arooj Saghir

By

Fast Food Management

Data Structures and Algorithms

University of Gujrat

System

CS-209

Faculty of Computing and Information Technology

Ms. Ayesha Rashid

Algorithms

MainClass():

* + 1. Start
    2. Print “Enter Username”
    3. Get String input.
    4. Print”Enter Password”
    5. Get String input.
    6. If (password==null) then
    7. Print “no user with this password”
    8. Else if(tempPass == Password)
    9. Print “welcome to main menu”
    10. Menu.mainMenu();
    11. Else
    12. Print “In correct Password”
    13. End if Else statement
    14. End

Admin Package

AdminMenu():

* + - 1. Start
      2. Using try and catch for exception
      3. While(True)
      4. Get Character
      5. Do
      6. Print All of Your Menu Choices
      7. Print”Enter Your Choice”
      8. Enter the char type input.
      9. Switch (ch)
      10. Case statements to determine the choice selection
      11. Default: Print “Invalid Choice Enter again”
      12. End of Switch
      13. While(! Defined choices)
      14. End While
      15. Catch (exception e)
      16. End

AdminSecurity():

1. Start
2. Private int admin ID
3. Private String adminUserName
4. Private String adminPassword
5. Create function string toString()
6. Return AdminSecurity
7. AdminID= “ +adminID+”,
8. adminUserName= “+adminUserName+”,
9. adminPassword = “+adminPassword+”
10. Generate Getter and Setters
11. End

AdminSecurityDao():

1. Start
2. Database Connection
3. Static String Url= “Database Url”
4. Static String username= “root”
5. Static String password= “7350”
6. Creating static admin method()
7. Insert value to Admin Security
8. Using try and catch
9. Get admin id to set in the 1 column of the database
10. Get adminUserName to set in the 2 column of the database
11. Get adminPassword to set in the 3 column of the database.
12. End try and catch
13. Return adminSecurity;
14. Create a UpdateAdmin method()
15. Boolean success = false
16. String queryUpdate = “Update Admin Security Set”
17. + “adminUserName =” + get adminUserName()
18. + “adminPassword=” + get adminPassword()
19. + AdminID=” + get adminID
20. Using Try and catch
21. Get connection (url,username,password)
22. Int st = s.executeUpdate(queryupdate)
23. Success=st(!=o ?true :false)
24. Return Success;
25. Create a static Boolean Delete Admin Method()
26. Boolean success = false
27. queryDelete = “Delete from AdminSecurity where Admin ID” + adminID;
28. Using Try and catch
29. Get connection (url,username,password)
30. Int st = s.executeUpdate(queryupdate)
31. Success=st(!=o ?true :false)
32. Return Success;
33. Create a getAllAdmins() method
34. Make a new linkedlist<>
35. Using try and catch
36. Get connection (url,username,password)
37. Using while loop
38. Set adminID(rs.getInt(1))
39. Set adminUserName(rs.getString(2))
40. Set adminPassword(rs.getString(3))
41. Return admins;
42. End

Customer Package

Customer():

* + 1. Start
    2. Declaring Private Customer Attributes
    3. Generating Getter an Setter
    4. Creating a String toString() function
    5. End

CustomerDao():

1. Start
2. Database Connection
3. Static String Url= “Database Url”
4. Static String username= “root”
5. Static String password= “7350”
6. Creating add record () method
7. Insert into customer value
8. Using Try and catch
9. Get connection (url,username,password)
10. Get customer Entry number to set in the column 1 of database
11. Get customer name() to set in the column 2 of the database
12. Get customer bill() to set in the column 3 of the database
13. Get customer foodItemsID() to set in the column 4 of the database
14. Get customer foodPurchaseTime() to set in the column 5 of the database
15. Get Customer FoodItemName() to set in the column 6 of the database
16. Get Customer Memberships() to set in the column 7 of the Database
17. Get OrderID to set in the column 8 of the database
18. Excute() it and then close()
19. Return Customer;
20. Create GetCustomer() method
21. Select Customer from their Customer entry Number
22. Get connection (url,username,password)
23. If (resultSet.next()) then
24. It wil go through the table and display the data accordingly
25. Else
26. Customer = null;
27. End if else
28. Return customer
29. Create a getAllHistory() method
30. create new linkedlist customers
31. get query Select \* from Customer
32. using try and catch
33. Get connection (url,username,password)
34. While(resultSet.next())
35. Set the customer details in the tables accordingly
36. End while
37. Return customer;
38. Create UpdateCustomer() method
39. Boolean success = false;
40. Queryupdate = Update customer set
41. Get all customer details accordingly
42. Using Try and catch
43. Get connection (url,username,password)
44. Int st = s.executeUpdate(queryupdate)
45. Success=st(!=o ?true :false)
46. Return Success;
47. Create a deletCustomer() method
48. Boolean success = false;
49. Queryupdate = Delete from customer by getting customerEntityNumber
50. Using Try and catch
51. Get connection (url,username,password)
52. Int st = s.executeUpdate(queryupdate)
53. Success=st(!=o ?true :false)
54. Return Success;
55. End

CustomerMembership():

1. Start
2. Declaring private customer Membership attributes
3. Generating getter and setter
4. Creating a string ToString() function
5. End

CustomerMembershipType():

1. Start
2. BRONZE(“bronze”)
3. SILVER(“Silver”)
4. GOLD(“Gold”)
5. Create a string type method
6. END

CustomerMembershipDao():

1. Start
2. Database Connection
3. Static String Url= “Database Url”
4. Static String username= “root”
5. Static String password= “7350”
6. Create addMemeberShip() method
7. queryInsert = “insert into Customer\_membership\_Value”
8. Get connection (url,username,password)
9. get all membership details accordingly and place in tables
10. return CustomerMembership;
11. create a getAllMembership() method
12. queryGet = “Select \* from customer\_Member\_ship”
13. using try and catch
14. Get connection (url,username,password)
15. While(resultSet.next())
16. Set all memebership details accordingly
17. End while
18. Return CustomerMemebership;
19. Create a UpdateCustomerMembership () method
20. Boolean success = false;
21. Queryupdate = Update customer\_Membership set
22. Get all customer details accordingly
23. Using Try and catch
24. Get connection (url,username,password)
25. Int st = s.executeUpdate(queryupdate)
26. Success=st(!=o ?true :false)
27. Return Success;
28. Create a deleteCustomerMembership () method
29. Boolean success = false;
30. QueryDelete = delete from customer\_Membership set
31. Using Try and catch
32. Get connection (url,username,password)
33. Int st = s.executeUpdate(queryupdate)
34. Success=st(!=o ?true :false)
35. Return Success;

Employee Package

Employee():

1. Start
2. Declaring private Employee attributes
3. Generating getter and setter
4. Creating String toSting() function
5. End

EmployeeType():

1. Start
2. SINGLE(“Single”)
3. MARRIED(“Married”)
4. STUDENT(“Student”)
5. Create a string type EmployeeStatus
6. End

EmployeeDao():

1. Start
2. Database Connection
3. Static String Url= “Database Url”
4. Static String username= “root”
5. Static String password= “7350”
6. Using Try and catch
7. Get connection (url,username,password)
8. queryTableCheck = select \* from Information
9. create a Boolean type exist variable to check the table
10. if (! exist)
11. query = “Create a table employee”
12. insert all the employee respective attributes
13. create a Boolean type success variable to excute the table
14. success == true ? “table created” : “table not created”;
15. else
16. print “employee table”
17. Create addEmployee()
18. queryInset = “Insert into Employee value”
19. using try and catch
20. Get connection (url,username,password)
21. get all employee details accordingly
22. return employee
23. create a getAllEmployee()
24. make new linkedlist of Employees
25. queryGet = “Select \* from employee”
26. using try and catch
27. Get connection (url,username,password)
28. While(resultSet.next())
29. Set all employee details accordingly
30. End while
31. Return employees
32. Create a getEmployee()
33. queryGet = “Select \* from employee ”
34. using try and catch
35. Get connection (url,username,password)
36. If (resultSet.next())
37. Set all employees values in the table accordingly
38. Else
39. Employee = null
40. Return employee
41. Create a getWorkingEmployee()
42. queryGet = “Select \* from employee”
43. using try and catch
44. While(resultSet.next())
45. Set all employee details accordingly
46. End while
47. Retun employees;
48. Create a updateEmployee()
49. Boolean success= false
50. QueryUpdate = “Update Employee set”
51. Get all employees deatails accordingly
52. Using try and catch
53. Get connection(url,username,password)
54. Int st=s.executeUpdate(queryUpdate)
55. Success = st(!0? true:false)
56. Return success
57. Create a deleteEmployee()
58. Boolean success= false
59. QueryUpdate = “delete from Employee code”
60. Using try and catch
61. Get connection(url,username,password)
62. Int st=s.executeUpdate(queryUpdate)
63. Success = st(!0? true:false)
64. Return success

FoodItem Package

FoodItem():

1. Start
2. Declaring private foodItem Attributes
3. Generate getter and setter
4. Create string toString()
5. End

FoodItemCategory():

1. Start
2. FASTFOOD(“FastFood”)
3. DESSERT(“Dessert”)
4. DRINK(“Drink”)
5. Create a string type FoodItemCategory
6. End

FoodItemDao():

1. Start
2. Database Connection
3. Static String Url= “Database Url”
4. Static String username= “root”
5. Static String password= “7350”
6. Create a AddFoodItem()
7. queryInsert = “Insert into Food item”
8. Get connection(url,username,password)
9. Get all fooditems details accordingly
10. Return FoodItem
11. Create a getFoodItem()
12. queryGet = “Select From FoodItem"
13. using try and catch
14. Get connection(url,username,password)
15. If (rs.next())
16. Set all foodItems details accordingly
17. Else
18. foodItem = null;
19. return FoodItem;
20. create getAllFoodItems()
21. queryGet = “Select from FoodItems”
22. using try and catch
23. Get connection(url,username,password)
24. While (rs.next())
25. Set all foodItem details accordingly
26. Return FoodItems
27. Create UpdateFoodItem()
28. Boolean success= false
29. QueryUpdate = “update from FoodItem set”
30. Using try and catch
31. Get connection(url,username,password)
32. Int st=s.executeUpdate(queryUpdate)
33. Success = st(!0? true:false)
34. Return success
35. Create a deleteFoodItem()
36. Boolean success= false
37. QueryUpdate = “delete from FoodItem”
38. Using try and catch
39. Get connection(url,username,password)
40. Int st=s.executeUpdate(queryUpdate)
41. Success = st(!0? true:false)
42. Return success

FoodItemDeals Package

FoodItemDeals():

1. Start
2. Create a new linkedlist of FoodItemsDeals
3. Declare FoodItemDealsPrice
4. Create a function()
5. Declare Integer[] type ids
6. For int i=0 and i<ids.length
7. Add food item to food items deals
8. FooditemDeals foreach(item-> print item.get(0))
9. End

Input Package

Input():

1. Start
2. Create a integer Input() method
3. While(true)
4. try
5. get input
6. catch
7. Print “Invalid Input”
8. End while
9. Return integer
10. Create a Long InputLong() method
11. While(true)
12. try
13. get input
14. catch
15. Print “Invalid Input”
16. End while
17. Return Long
18. Create Boolean type InputBoolean()
19. Boolean bool=false
20. While(true)
21. Try
22. Get input
23. Catch
24. Print “Invalid Input”
25. Return bool
26. Create String InputString()
27. String String= “”
28. While(true)
29. Try
30. If (string [a-zA-Z]==true)
31. Return string
32. Else if(string [a-zA-Z0-9]== true)
33. Print “Invalid input”
34. Catch
35. Print “Invalid input”
36. Return String

LinkedList Package

Node():

1. Start
2. Declare all the respective variables
3. Generates getter and setter
4. Create Node() with parameters
5. all variables specified with This type
6. End

LinkedList():

1. Start
2. Create Node head and tail
3. Int count
4. Create add()
5. If(head== null)
6. Head = temp
7. Else
8. Make current of node equal to head
9. While(current.getnext!=null)
10. Cur= current.getNext()
11. Setnext(temp)
12. Tail=temp;
13. Count ++
14. Create get(int id)
15. Make cur = head and node= null;
16. While(current.getnext!=null)
17. If (cur.getID() !=null)
18. Current= current.getNext()
19. Else if(cur.getID= id)
20. Return current;
21. Create GetAll()
22. If (head==null)
23. Print “no value”
24. Make current of node to head
25. While(current != null)
26. Current.GetID()
27. Make current = getNext()
28. Create int getCount()
29. Return count++
30. End

Order Package

order():

1. Start
2. Declare all the order attributes
3. Generate getter and setter
4. Create a String ToString() function
5. End

orderDao():

1. Start
2. Database Connection
3. Static String Url= “Database Url”
4. Static String username= “root”
5. Static String password= “7350”
6. Create AddOrder()
7. queryInsert = “Insert into Order Value”
8. Using try and catch
9. Get connection(url,username,password)
10. Get all the order details accordingly
11. Return true
12. Create getAllOrder()
13. queryGet = “Select From Order”
14. using try and catch
15. Get connection(url,username,password)
16. While(rs.next())
17. Set all the order details accordingly
18. Return orders
19. Create getOrder()
20. QueryGet= “Select \* from where orderID”
21. Using try and catch
22. Get connection(url,username,password)
23. Set all the order details accordingly
24. Else
25. Order= null
26. End if else
27. Return order

Menu Package

CustomerMemberShipMenu():

1. Start
2. Try
3. While(true)
4. Get ch
5. Do
6. Print all of your menu choices
7. Print “Enter your choice”
8. Enter the char type input
9. Switch(ch)
10. Case statement to determine the choice selection
11. Default: print “Invalid choice Enter again”
12. End of Switch
13. While(!Defined choices)
14. End while
15. Catch
16. End

CustomerMenu():

1. Start
2. Try
3. While(true)
4. Get ch
5. Do
6. Print all of your menu choices
7. Print “Enter your choice”
8. Enter the char type input
9. Switch(ch)
10. Case statement to determine the choice selection
11. Default: print “Invalid choice Enter again”
12. End of Switch
13. While(!Defined choices)
14. End while
15. Catch
16. End

EmployeeMenu():

1. Start
2. Try
3. While(true)
4. Get ch
5. Do
6. Print all of your menu choices
7. Print “Enter your choice”
8. Enter the char type input
9. Switch(ch)
10. Case statement to determine the choice selection
11. Default: print “Invalid choice Enter again”
12. End of Switch
13. While(!Defined choices)
14. End while
15. Catch
16. End

We will have the same menu classes for FoodItem,FoodMenu,Menu and OrderMenu.