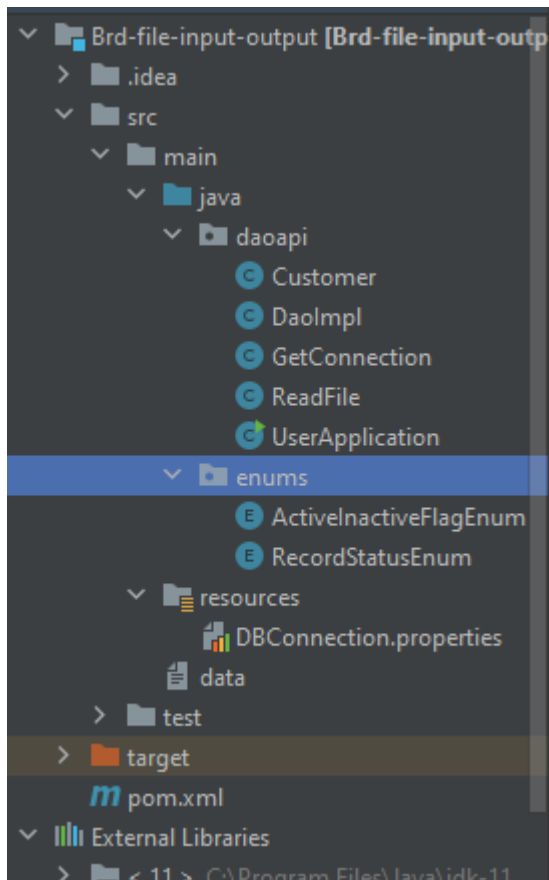


ASSIGNMENT BRD-Input-Output

Harshit Kushmakar | 16896



```
package daoapi;
import enums.ActiveInactiveFlagEnum;
import enums.RecordStatusEnum;

import java.time.LocalDate;
public class Customer {
    private long customerId;
    private String customerCode;
    private String customerName;
    private String customerAddress1;
    private String customerAddress2;
    private long customerPinCode;
    private String emailAddress;
    private long contactNumber;
    private String primaryContactPerson;
    private ActiveInactiveFlagEnum activeInactiveFlag;
    private LocalDate createDate;
    private String createdBy;
    private LocalDate modifiedDate;
    private String modifiedBy;
    private LocalDate authorizedDate;
    private String authorizedBy;

    private RecordStatusEnum recordStatusEnums;
```

```

public RecordStatusEnum getRecordStatusEnums() {
    return recordStatusEnums;
}

public void setRecordStatusEnums(RecordStatusEnum recordStatusEnums) {
    this.recordStatusEnums = recordStatusEnums;
}

public ActiveInactiveFlagEnum getActiveInactiveFlagEnum() {
    return activeInactiveFlagEnum;
}

public void setActiveInactiveFlagEnum(ActiveInactiveFlagEnum
activeInactiveFlagEnum) {
    this.activeInactiveFlagEnum = activeInactiveFlagEnum;
}

private ActiveInactiveFlagEnum activeInactiveFlagEnum;
private static long incrementId;
public Customer(){ }
public Customer(String customerName) {
    incrementId += 1;
    this.customerId = incrementId;
    this.customerName = customerName;
}

public long getCustomerId() {
    return customerId;
}

public String getCustomerCode() {
    return customerCode;
}

public void setCustomerCode(String customerCode) {
    this.customerCode = customerCode;
}

public String getCustomerName() {
    return customerName;
}

public void setCustomerName(String customerName) {
    this.customerName = customerName;
}

public String getCustomerAddress1() {
    return customerAddress1;
}

public void setCustomerAddress1(String customerAddress1) {
    this.customerAddress1 = customerAddress1;
}

public String getCustomerAddress2() {
    return customerAddress2;
}

public void setCustomerAddress2(String customerAddress2) {
    this.customerAddress2 = customerAddress2;
}

public long getCustomerPinCode() {
    return customerPinCode;
}

public void setCustomerPinCode(long customerPinCode) {
    this.customerPinCode = customerPinCode;
}

```

```
public String getEmailAddress() {
    return emailAddress;
}
public void setEmailAddress(String emailAddress) {
    this.emailAddress = emailAddress;
}
public long getContactNumber() {
    return contactNumber;
}
public void setContactNumber(long contactNumber) {
    this.contactNumber = contactNumber;
}
public String getPrimaryContactPerson() {
    return primaryContactPerson;
}
public void setPrimaryContactPerson(String primaryContactPerson) {
    this.primaryContactPerson = primaryContactPerson;
}
public RecordStatusEnum getRecordStatus() {
    return recordStatusEnums;
}
public void setRecordStatus(RecordStatusEnum recordStatus) {
    this.recordStatusEnums = recordStatus;
}
public ActiveInactiveFlagEnum getActiveInactiveFlag() {
    return activeInactiveFlag;
}
public void setActiveInactiveFlag(ActiveInactiveFlagEnum
activeInactiveFlag) {
    this.activeInactiveFlag = activeInactiveFlag;
}
public LocalDate getCreateDate() {
    return createDate;
}
public void setCreateDate(LocalDate createDate) {
    this.createDate = createDate;
}
public String getCreatedBy() {
    return createdBy;
}
public void setCreatedBy(String createdBy) {
    this.createdBy = createdBy;
}
public LocalDate getModifiedDate() {
    return modifiedDate;
}
public void setModifiedDate(LocalDate modifiedDate) {
    this.modifiedDate = modifiedDate;
}
public String getModifiedBy() {
    return modifiedBy;
}
public void setModifiedBy(String modifiedBy) {
    this.modifiedBy = modifiedBy;
}
public LocalDate getAuthorizedDate() {
    return authorizedDate;
}
public void setAuthorizedDate(LocalDate authorizedDate) {
    this.authorizedDate = authorizedDate;
}
}
```

```

public String getAuthorizedBy() {
    return authorizedBy;
}

public void setAuthorizedBy(String authorizedBy) {
    this.authorizedBy = authorizedBy;
}

@Override
public String toString() {
    return "\nCustomer{" +
        "customerId=" + customerId +
        ", customerCode=" + customerCode + '\'' +
        ", customerName=" + customerName + '\'' +
        ", customerAddress1=" + customerAddress1 + '\'' +
        ", customerAddress2=" + customerAddress2 + '\'' +
        ", customerPinCode=" + customerPinCode +
        ", emailAddress=" + emailAddress + '\'' +
        ", contactNumber=" + contactNumber +
        ", primaryContactPerson=" + primaryContactPerson + '\'' +
        ", activeInactiveFlag=" + activeInactiveFlag +
        ", createDate=" + createDate +
        ", createdBy=" + createdBy + '\'' +
        ", modifiedDate=" + modifiedDate +
        ", modifiedBy=" + modifiedBy + '\'' +
        ", authorizedDate=" + authorizedDate +
        ", authorizedBy=" + authorizedBy + '\'' +
        ", recordStatusEnums=" + recordStatusEnums +
        ", activeInactiveFlagEnum=" + activeInactiveFlagEnum +
        '}';
}
}

```

```

package daoapi;

import java.sql.*;

public class DaoImpl {
    private static Connection connect = GetConnection.getConnection();

    public boolean insert(Customer customer) {
        int rowCount = 0;
        String query = "insert into customer_master_table(CUSTOMER_CODE, CUSTOMER_NAME, CUSTOMER_ADDRESS1, " + "CUSTOMER_ADDRESS2, CUSTOMER_PINCODE, EMAIL_ADDRESS, CONTACT_NUMBER, PRIMARAY_CONTACT_PERSON, RECORD_STATUS," + "ACTIVE_INACTIVE_FLAG, CREATE_DATE, MODIFIED_DATE,MODIFIED_BY, AUTHORIZED_DATE, AUTHORIZED_BY) " + "values(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
        try {
            System.out.println("Running Statement");
            PreparedStatement preparedStatement = connect.prepareStatement(query);
            preparedStatement.setString(1, customer.getCustomerCode());
            preparedStatement.setString(2, customer.getCustomerName());
            preparedStatement.setString(3, customer.getCustomerAddress1());
            preparedStatement.setString(4, customer.getCustomerAddress2());
            preparedStatement.setLong(5, customer.getCustomerPinCode());
            preparedStatement.setString(6, customer.getEmailAddress());
            preparedStatement.setLong(7, customer.getContactNumber());

```

```

        preparedStatement.setString(8,
customer.getPrimaryContactPerson());
        preparedStatement.setString(9,
customer.getRecordStatus().toString());
        preparedStatement.setString(10,
customer.getActiveInactiveFlag().toString());
        preparedStatement.setDate(11,
Date.valueOf(customer.getCreateDate()));
        preparedStatement.setString(12, customer.getCreatedBy());
        preparedStatement.setDate(13,
Date.valueOf(customer.getModifiedDate()));
        preparedStatement.setString(14, customer.getModifiedBy());
        preparedStatement.setDate(15,
Date.valueOf(customer.getAuthorizedDate()));
        preparedStatement.setString(16, customer.getAuthorizedBy());
        rowCount = preparedStatement.executeUpdate();
        preparedStatement.close();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        commit();
    }
    if (rowCount > 0) {
        System.out.println("1 row inserted");
    } else {
        System.out.println("Customer code = " + " ");
    }
    return rowCount > 0;
}

public void commit() {
    String query = "COMMIT";
    try {
        Statement statement = connect.createStatement();
        statement.execute(query);
        statement.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
}

```

```

package daoapi;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class GetConnection {
    public static Connection getConnection() {
        Connection connection = null;
        try {

```

```

        Class.forName("oracle.jdbc.driver.OracleDriver");
        System.out.println("Driver registered");
        // making connection
        connection =
DriverManager.getConnection("=jdbc:oracle:thin:sh@10.1.50.198:1535/nsbt19c"
, "sh", "sh");
        System.out.println("Connection created");

    } catch ( ClassNotFoundException | SQLException e) {
        e.printStackTrace();
    }
    return connection;
}
}

```

```

package daoapi;

import com.validation.kushmakar.Validation;
import enums.ActiveInactiveFlagEnum;
import enums.RecordStatusEnum;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.text.SimpleDateFormat;
import java.util.*;

public class ReadFile {
    public List<Customer> readFile() {
        List<Customer> customers = new ArrayList<>();
        try {
            String fileName =
"C:\\Users\\harshit.kushmakar\\IdeaProjects\\data.txt";

            if (Validation.checkFileExtension(fileName)) {
                File file = new File(fileName);
                BufferedReader br = new BufferedReader(new
FileReader(file));
                Customer customer = new Customer();
                while (br != null){

                    String[] arr = br.readLine().split("#");

//System.out.println(Validation.dataTypeValidation(arr[0], "String"));
                    if (Validation.dataTypeValidation(arr[0], "String")
&&
                        Validation.dataLengthValidation(arr[0], 10)) {
                        //System.out.println("True");
                        customer.setCustomerCode(arr[0]);
                    }
                    if (Validation.dataTypeValidation(arr[1], "String")
&&
                        Validation.dataLengthValidation(arr[1], 30)) {
                        customer.setCustomerName(arr[1]);
                    }
                    if (Validation.dataTypeValidation(arr[2], "String")
&&
                        Validation.dataLengthValidation(arr[2], 100)) {

```

```

        customer.setCustomerAddress1(arr[2]);
    }
    if (Validation.dataTypeValidation(arr[3], "String")
        && Validation.dataLengthValidation(arr[3],
100)) {
        customer.setCustomerAddress2(arr[3]);
    }
    if (Validation.dataTypeValidation(arr[4], "String") &&
        Validation.dataLengthValidation(arr[4], 6)) {

customer.setCustomerPinCode(Integer.parseInt(arr[4]));
    }

    if (Validation.dataTypeValidation(arr[5], "String")
        &&
        Validation.dataLengthValidation(arr[5], 100) &&
        Validation.validateEmail(arr[5])) {
        customer.setEmailAddress(arr[5]);
    }

    customer.setContactNumber(Long.parseLong(arr[6]));

    if (Validation.dataTypeValidation(arr[7], "String")) {
        customer.setPrimaryContactPerson(arr[7]);
    }

customer.setRecordStatus(RecordStatusEnum.valueOf(arr[8]));

customer.setActiveInactiveFlag(ActiveInactiveFlagEnum.valueOf(arr[9]));

        String createDate = arr[10];
        Date creDate = new
SimpleDateFormat("dd/MMM/yyyy").parse(createDate);
        java.sql.Date maindate = new
java.sql.Date(creDate.getTime());
        customer.setCreateDate(maindate.toLocalDate());
        if (Validation.dataTypeValidation(arr[11], "String")
            && Validation.dataLengthValidation(arr[11],
30)) {
            customer.setCreatedBy(arr[11]);
        }
        if (arr.length > 12) {
            if (arr[12].equals("")) {
                customer.setModifiedDate(null);
            } else {
                String ModifiedDate = arr[12];
                Date modDate = new
SimpleDateFormat("dd/MMM/yyyy").parse(ModifiedDate);
                java.sql.Date moddate = new
java.sql.Date(modDate.getTime());
                customer.setModifiedDate(moddate.toLocalDate());
            }
        }
        if (arr.length > 13) {
            if (arr[13].equals(""))
                customer.setModifiedBy(null);
            else if (Validation.dataTypeValidation(arr[13],
"String") &&
                Validation.dataLengthValidation(arr[13],
30)) {

```

```

        customer.setModifiedBy(arr[13]);
    }
}
if (arr.length > 14) {
    if (arr[14].equals("")) {
        customer.setAuthorizedDate(null);
    } else {
        String authorisedDate = arr[14];
        Date authDate = new
SimpleDateFormat("dd/MMM/yyyy").parse(authorisedDate);
        java.sql.Date audate = new
java.sql.Date(authDate.getTime());

customer.setAuthorizedDate(audate.toLocalDate());
    }
}
if (arr.length > 15) {
    if (arr[15].equals(""))
        customer.setAuthorizedBy(null);
    else if (Validation.dataTypeValidation(arr[15],
"String") &&
Validation.dataLengthValidation(arr[15],
30)) {
        customer.setAuthorizedBy(arr[15]);
    }
}
customers.add(customer);
System.out.println(customer.toString());
}
}
} catch (Exception e) {
}
return customers;
}
}

```

```

package daoapi;

import daoapi.DaoImpl;

import java.util.List;

public class UserApplication {
    public static void main(String[] args) {
        ReadFile obj = new ReadFile();
        List<Customer> customerArray = obj.readFile();
        // for(int i=0; i<customerArray.size(); i++){
        //     Customer obj1 = null;
        //     System.out.println(customerArray.get(i).getCustomerCode());
        // }
    }
}

```



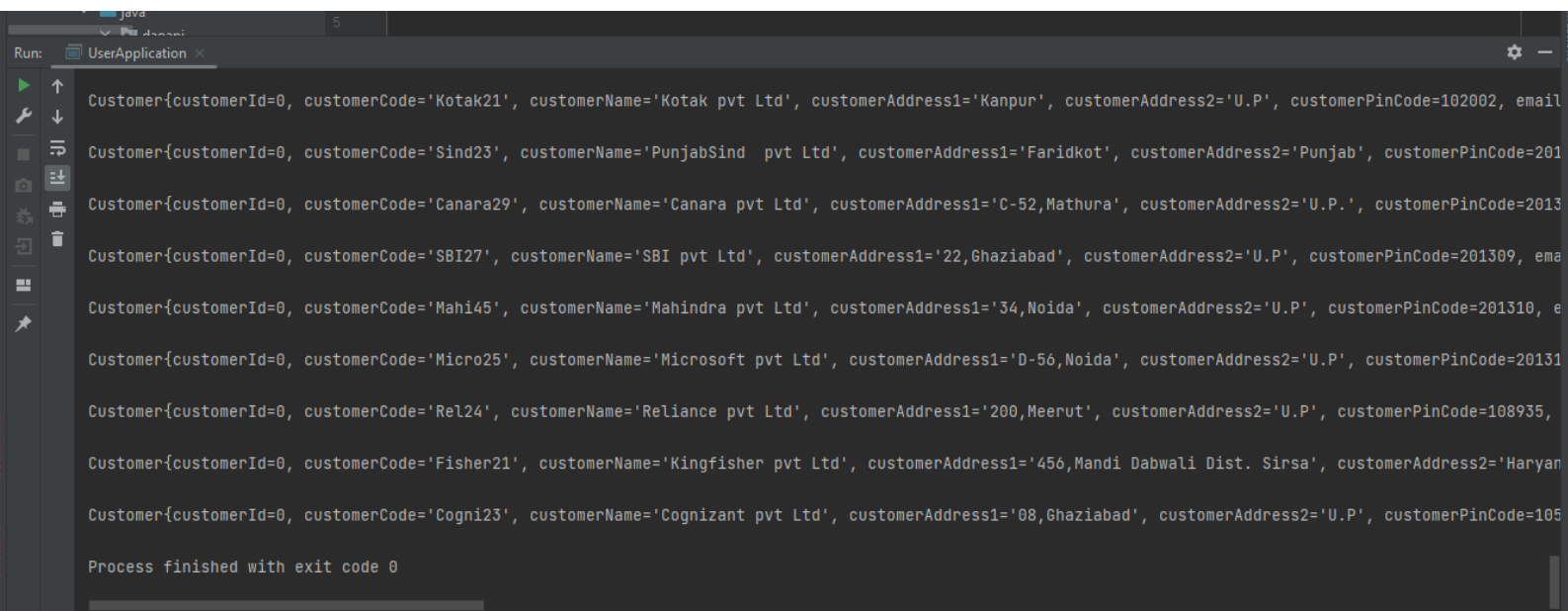
```
package enums;

public enum ActiveInactiveFlagEnum {
    A, I;
}
```

```
package enums;

public enum RecordStatusEnum {
    N, M, D, A, R;
}
```

OUTPUT:



```
Run: UserApplication x
Customer{customerId=0, customerCode='Kotak21', customerName='Kotak pvt Ltd', customerAddress1='Kanpur', customerAddress2='U.P.', customerPinCode=102002, email=
Customer{customerId=0, customerCode='Sind23', customerName='PunjabSind pvt Ltd', customerAddress1='Faridkot', customerAddress2='Punjab', customerPinCode=201
Customer{customerId=0, customerCode='Canara29', customerName='Canara pvt Ltd', customerAddress1='C-52,Mathura', customerAddress2='U.P.', customerPinCode=2013
Customer{customerId=0, customerCode='SBI27', customerName='SBI pvt Ltd', customerAddress1='22,Ghaziabad', customerAddress2='U.P.', customerPinCode=201309, ema
Customer{customerId=0, customerCode='Mahi45', customerName='Mahindra pvt Ltd', customerAddress1='34,Noida', customerAddress2='U.P.', customerPinCode=201310, e
Customer{customerId=0, customerCode='Micro25', customerName='Microsoft pvt Ltd', customerAddress1='D-56,Noida', customerAddress2='U.P.', customerPinCode=20131
Customer{customerId=0, customerCode='Rel24', customerName='Reliance pvt Ltd', customerAddress1='200,Meerut', customerAddress2='U.P.', customerPinCode=108935,
Customer{customerId=0, customerCode='Fisher21', customerName='Kingfisher pvt Ltd', customerAddress1='456,Mandi Dabwali Dist. Sirsa', customerAddress2='Haryan
Customer{customerId=0, customerCode='Cogni23', customerName='Cognizant pvt Ltd', customerAddress1='08,Ghaziabad', customerAddress2='U.P.', customerPinCode=105
Process finished with exit code 0
```

```
Run: UserApplication x
', customerPinCode=102002, emailAddress='Kotak@yahoo.com', contactNumber=8802641195, primaryContactPerson='Aditi', activeInactiveFlag=A, createDate=2010-08-2
s2='Punjab', customerPinCode=201307, emailAddress='PunjabSind@yahoo.com', contactNumber=9463800843, primaryContactPerson='Rahul', activeInactiveFlag=A, creat
ss2='U.P.', customerPinCode=201308, emailAddress='Canara@yahoo.com', contactNumber=9820063303, primaryContactPerson='Nippun', activeInactiveFlag=A, createDat
.P.', customerPinCode=201309, emailAddress='SBI@yahoo.com', contactNumber=9999903632, primaryContactPerson='Supreet', activeInactiveFlag=A, createDate=2010-08
'U.P.', customerPinCode=201310, emailAddress='Mahindra@yahoo.com', contactNumber=9811351558, primaryContactPerson='Sahil', activeInactiveFlag=A, createDate=20
ss2='U.P.', customerPinCode=201311, emailAddress='Microsoft@yahoo.com', contactNumber=9252645006, primaryContactPerson='Mahinga', activeInactiveFlag=A, create
='U.P.', customerPinCode=108935, emailAddress='Reliance@yahoo.com', contactNumber=9582403099, primaryContactPerson='Saurabh', activeInactiveFlag=A, createDate
Sirsa', customerAddress2='Haryana', customerPinCode=102567, emailAddress='Kingfisher@yahoo.com', contactNumber=9410993125, primaryContactPerson='Ashish', act
ress2='U.P.', customerPinCode=105786, emailAddress='Cognizant@yahoo.com', contactNumber=9988952821, primaryContactPerson='Abhishek', activeInactiveFlag=A, cre
```

```
Run: UserApplication x
createDate=2010-08-21, createdBy='Priyanka', modifiedDate=2010-08-25, modifiedBy='Anand', authorizedDate=2010-08-26, authorizedBy='Shashank', recordStatusEr
InactiveFlag=A, createDate=2010-08-21, createdBy='Shelly', modifiedDate=2010-08-25, modifiedBy='Anand', authorizedDate=2010-08-26, authorizedBy='Shashank', r
tiveFlag=A, createDate=2010-08-21, createdBy='Priyanka', modifiedDate=null, modifiedBy='null', authorizedDate=2010-08-25, authorizedBy='Shashank', recordStat
A, createDate=2010-08-21, createdBy='Himanshu', modifiedDate=null, modifiedBy='null', authorizedDate=2010-08-25, authorizedBy='Shashank', recordStatusEnums=N
Flag=A, createDate=2010-08-21, createdBy='Vikas', modifiedDate=null, modifiedBy='null', authorizedDate=2010-08-25, authorizedBy='Shashank', recordStatusEnums
nactiveFlag=A, createDate=2010-08-21, createdBy='Mukesh', modifiedDate=null, modifiedBy='null', authorizedDate=2010-08-25, authorizedBy='Shashank', recordSta
iveFlag=A, createDate=2010-08-21, createdBy='Anant', modifiedDate=null, modifiedBy='null', authorizedDate=2010-08-25, authorizedBy='Shashank', recordStatusEr
tPerson='Ashish', activeInactiveFlag=A, createDate=2010-08-21, createdBy='Asmita', modifiedDate=null, modifiedBy='null', authorizedDate=2010-08-26, authorize
veInactiveFlag=A, createDate=2010-08-21, createdBy='Saurabh', modifiedDate=null, modifiedBy='null', authorizedDate=2010-08-26, authorizedBy='Shashank', recor
```

```
Run: UserApplication x
8-26, authorizedBy='Shashank', recordStatusEnums=N, activeInactiveFl
edDate=2010-08-26, authorizedBy='Shashank', recordStatusEnums=N, act
10-08-25, authorizedBy='Shashank', recordStatusEnums=N, activeInacti
authorizedBy='Shashank', recordStatusEnums=N, activeInactiveFlagEnu
5, authorizedBy='Shashank', recordStatusEnums=N, activeInactiveFlagE
010-08-25, authorizedBy='Shashank', recordStatusEnums=N, activeInact
8-25, authorizedBy='Shashank', recordStatusEnums=N, activeInactiveFl
```