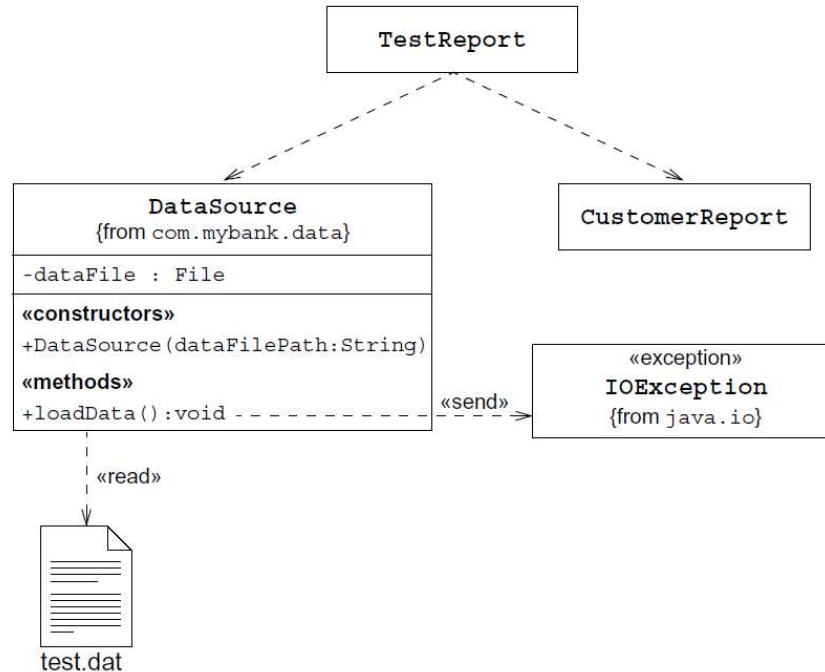


Console I/O and File I/O

Lab 14: Reading a Data File

In this exercise, you create a class that reads customer and account data from a flat file.

Figure below shows the UML diagram for the DataSource class that you create for the TestReport program.



Code below shows an example of the format of the customer data file. The first line contains an integer, which determines the number of customers in the data file. A customer record contains the first name, last name, and the number of accounts, separated by tab characters. Each account record contains a single-character code that determines the type of account and also the data in that record.

```
<number-of-customers>

<first-name> <last-name> <number-of-accounts>
<account-type-code> <datum1> <datum2>
```

Code below shows an example of the format of the customer data file. This data file contains four customer records. The first is for Jane Simms; Jane has two bank accounts. The first account is a savings account, with an initial balance of 500.00 and an interest rate of 5 percent (0.05). The second account is a checking account with an initial balance of 200.00 and overdraft protection of 400.00.

4

Jane	Simms	2
S	500.00	0.05
C	200.00	400.00

Owen	Bryant	1
C	200.00	0.00

Tim	Soley	2
S	1500.00	0.05
C	200.00	0.00

Maria	Soley	1
S	150.00	0.05

1. Create a data directory in the BankProject.
2. Download and copy the test.dat data file into the data directory.
3. Create the DataSource class in the com.mybank.data source package of the BankProject. The class must satisfy the UML diagram in the diagram shown above. The loadData method must use Bank utility methods (addCustomer and getCustomer) to populate the customers recorded in the data file. Furthermore, the Customer class has the addAccount method to add the accounts from the data file.
4. Delete the following Java classes that are no longer used.
 - TestReport
 - TestAccount
 - TestAccount2
 - TestBatch
 - TestBanking
5. Download and copy the new TestReport class into com.mybank.test package.
6. Set the TestReport class as the main class of the BankProject and run the BankProject with an argument data/test.dat.
7. The output should be similar to the following.

```
Reading data file: ../data/test.dat
CUSTOMERS REPORT
=====

Customer: Simms, Jane
    Savings Account: current balance is 500.0
    Checking Account: current balance is 200.0

Customer: Bryant, Owen
    Checking Account: current balance is 200.0

Customer: Soley, Tim
    Savings Account: current balance is 1500.0
    Checking Account: current balance is 200.0

Customer: Soley, Maria
    Savings Account: current balance is 150.0
```