**Scenario:**

“Tarangini Digital Services” is a DTH operator. They are looking at developing an application to let their subscribers manage their subscriptions.

**Task**

You have been assigned with a high priority task of implementing **THREE** functions to the existing **Bill Management System.** The below attached Java file (BudgetManager.java) has following classes definitions. Use this Java file to implement the functions.

1. BillManagementService - that generates and processes tracks the bills on each subscriber.
2. BillVO - Object representation of a bill record
3. SubscriberVO - Object representation of a subscriber record
4. PackageVO - Object representation of a package record
5. DTHDataRepo - Is a repository that supplies a list of Subscribers and Packages hardcoded.



***Points to remember***

1. *You can use STS/Eclipse IDE with JDK1.8 to develop the code for the implementation. Ensure that you follow best Java coding practice while coding.*
2. *Do not modify any attribute names and method signatures provided as part of the given Java file.*
3. *Code the implementation in the respective methods of the given class ONLY.*
4. *Do not hard code the return values of methods, if found your code will not be considered for processing.*
5. *Use Java SE 8.0 API only for the function implementation.*
6. *Upload/Submit the updated* BillManagementService*.java file for evaluation.*

Following are the THREE functions to be implemented.

1. **Function 1 – Computes the bills of each subscriber**

***Description***

**Retrieve the subscribers and package related data from the DTHDataRepo.getSubscribers() and DTHDataRepo.getPackages() function**. Compute the bill details of each subscriber as directed below and return the list of bills as BillVO objects.

1. BillVO::mobileNumber = SubscriberVO::getMobileNumber()
2. BillVO::registrationDate= SubscriberVO::getRegistrationDate()
3. BillVO:: billingTerm = SubscriberVO::getBillingTerm()
4. BillVO:: validTillDate = if billingTerm is “Monthly” then (SubscriberVO::getRegistrationDate() + 1 month) else (SubscriberVO::getRegistrationDate() + 1 year)
5. BillVO::amountPayable=if billingTerm is “Monthly” then (sum of PackageVO::monthlyFee of all packages subscribed by the subscriber) else (sum of PackageVO::annualFee of all packages subscribed by the subscriber)

***Implementation***

Class: **BillManagementService**

Method: **public List<BillVO> getBills()**

You can write additional utility methods wherever needed.

***Input***

***No params***

***Output:***

List<BillVO> where each BillVO represents the bill details of each subscriber

***Constraints:***

Use only streams api.

**Use DTHDataRepo.getSubscribers() to retrieve subscriber details.**

**Use DTHDataRepo.getPackages() to retrieve packages details.**

*Sample Input:*

No params.

*Sample Output:*

[

BillVO [mobileNumber=9247175830, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=614.0, billingTerm=Monthly],

BillVO [mobileNumber=9247212345, registrationDate=2021-01-10, validTillDate=2022-01-10, amountPayable=6776.0, billingTerm=Annually],

BillVO [mobileNumber=9848012345, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=9009.0, billingTerm=Annually],

BillVO [mobileNumber=9848023456, registrationDate=2021-01-11, validTillDate=2021-02-11, amountPayable=819.0, billingTerm=Monthly],

BillVO [mobileNumber=8093466666, registrationDate=2021-01-12, validTillDate=2021-02-12, amountPayable=919.0, billingTerm=Monthly],

BillVO [mobileNumber=8093477777, registrationDate=2021-01-12, validTillDate=2021-02-12, amountPayable=614.0, billingTerm=Monthly],

BillVO [mobileNumber=8093456565, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=616.0, billingTerm=Monthly],

BillVO [mobileNumber=8093478789, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=9009.0, billingTerm=Annually],

BillVO [mobileNumber=9247212541, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=717.0, billingTerm=Monthly],

BillVO [mobileNumber=9247112350, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=11231.0, billingTerm=Annually]

]

1. **Function 2 – Sort the List of bills on BillingTerm and ValidTillDate in Ascending order.**

***Description***

**Given the list of BillVO objects sort and return the list on billingTerm and validTillDate in ascending order,**

***Implementation***

Class: **BillManagementService**

Method: **public List<BillVO> sortBills(List<BillVO> unorderedBillsList)**

You can write additional utility methods wherever needed.

***Input***

unordered List<BillVO>

***Output***:

Ordered List<BillVO> on BillTerm and ValidTillDate in Ascending order

*Constraints*:

Use ONLY java.util.stream package and lambda expression for sorting.

Use Comparator for comparison mechanism.

*Sample Input:*

[

BillVO [mobileNumber=9247175830, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=614.0, billingTerm=Monthly],

BillVO [mobileNumber=9247212345, registrationDate=2021-01-10, validTillDate=2022-01-10, amountPayable=6776.0, billingTerm=Annually],

BillVO [mobileNumber=9848012345, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=9009.0, billingTerm=Annually],

BillVO [mobileNumber=9848023456, registrationDate=2021-01-11, validTillDate=2021-02-11, amountPayable=819.0, billingTerm=Monthly],

BillVO [mobileNumber=8093466666, registrationDate=2021-01-12, validTillDate=2021-02-12, amountPayable=919.0, billingTerm=Monthly],

BillVO [mobileNumber=8093477777, registrationDate=2021-01-12, validTillDate=2021-02-12, amountPayable=614.0, billingTerm=Monthly],

BillVO [mobileNumber=8093456565, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=616.0, billingTerm=Monthly],

BillVO [mobileNumber=8093478789, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=9009.0, billingTerm=Annually],

BillVO [mobileNumber=9247212541, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=717.0, billingTerm=Monthly],

BillVO [mobileNumber=9247112350, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=11231.0, billingTerm=Annually]

]

*Sample Output:*

[

BillVO [mobileNumber=9247212345, registrationDate=2021-01-10, validTillDate=2022-01-10, amountPayable=6776.0, billingTerm=Annually], BillVO [mobileNumber=9848012345, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=9009.0, billingTerm=Annually], BillVO [mobileNumber=8093478789, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=9009.0, billingTerm=Annually], BillVO [mobileNumber=9247112350, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=11231.0, billingTerm=Annually], BillVO [mobileNumber=9247175830, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=614.0, billingTerm=Monthly], BillVO [mobileNumber=8093456565, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=616.0, billingTerm=Monthly], BillVO [mobileNumber=9247212541, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=717.0, billingTerm=Monthly], BillVO [mobileNumber=9848023456, registrationDate=2021-01-11, validTillDate=2021-02-11, amountPayable=819.0, billingTerm=Monthly], BillVO [mobileNumber=8093466666, registrationDate=2021-01-12, validTillDate=2021-02-12, amountPayable=919.0, billingTerm=Monthly], BillVO [mobileNumber=8093477777, registrationDate=2021-01-12, validTillDate=2021-02-12, amountPayable=614.0, billingTerm=Monthly]

]

1. **Function 3 – retrieve the List of bills whose validTillDate is on or before given date**

***Description***

**Given a List of BillVO objects and a date, retrieve the List of bills whose validTillDate is on or before given date, WITHOUT USING filter() OPERATION ON STREAMS.**

***Implementation***

Class: **BillManagementService**

Method: **public List<BillVO> getBillsOnOrBefore(List<BillVO> billsList,LocalDate date)**

You can write additional utility methods wherever needed.

***Input***

List<BillVO> having all bill records,a date as LocalDate object

***Output***:

List<BillVO> having BillVO objects whose validTillDate is on or before given date

*Constraints*:

Use ONLY java.util.stream package and lambda expression but DO NOT USE filter() operation.

*HINT*

Use takeWhile() or dropWhile() operation as appropriate .

*Sample Input:*

List<BillVO>

[

BillVO [mobileNumber=9247175830, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=614.0, billingTerm=Monthly],

BillVO [mobileNumber=9247212345, registrationDate=2021-01-10, validTillDate=2022-01-10, amountPayable=6776.0, billingTerm=Annually],

BillVO [mobileNumber=9848012345, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=9009.0, billingTerm=Annually],

BillVO [mobileNumber=9848023456, registrationDate=2021-01-11, validTillDate=2021-02-11, amountPayable=819.0, billingTerm=Monthly],

BillVO [mobileNumber=8093466666, registrationDate=2021-01-12, validTillDate=2021-02-12, amountPayable=919.0, billingTerm=Monthly],

BillVO [mobileNumber=8093477777, registrationDate=2021-01-12, validTillDate=2021-02-12, amountPayable=614.0, billingTerm=Monthly],

BillVO [mobileNumber=8093456565, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=616.0, billingTerm=Monthly],

BillVO [mobileNumber=8093478789, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=9009.0, billingTerm=Annually],

BillVO [mobileNumber=9247212541, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=717.0, billingTerm=Monthly],

BillVO [mobileNumber=9247112350, registrationDate=2021-01-11, validTillDate=2022-01-11, amountPayable=11231.0, billingTerm=Annually]

],

Date: "2021-02-10"

*Sample Output:*

**List as below**

[

BillVO [mobileNumber=9247175830, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=614.0, billingTerm=Monthly], BillVO [mobileNumber=8093456565, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=616.0, billingTerm=Monthly], BillVO [mobileNumber=9247212541, registrationDate=2021-01-10, validTillDate=2021-02-10, amountPayable=717.0, billingTerm=Monthly]

]