

Sam, your unscrupulous supervisor, is developing a VB.net-based solution to automatically sort incoming letters (as in snail-mail, not e-mail) based on complex artificial intelligence, neural net processors, and other buzzwords from *Terminator*. Sam has been leeching the time of interns in the MIS Department to further this project, so Sam can take it public and make a mint. You're tasked with adding a sorting component to this incoming personal mailbox. Your basis is Sam's current method for receiving letters at home, which the sorting product will be based off of. Sam states:

I receive JunkMail, PersonalMail, Bills, and Magazines. Each of these are of type MailItem. Each day, I collect the set of these MailItems and make them into a MailSet. This MailSet has an iterator that will give your program one MailItem at a time until it runs out of MailItems.

You don't have to worry about how this iterator works, but you do have to know that there is a getNextMailItem() function to it that will retrieve the next MailItem from the MailSet. There's also a hasNext() function that returns either true or false, with true indicating that there is at least one MailItem remaining in the iterator that can be retrieved with getNextMailItem(). Finally, I also put a reset() function in the iterator that will take all of the day's MailItems and put them back into the iterator. It doesn't undo the sorting that's already occurred, but the already sorted MailItems may appear in the iterator again. Please avoid duplicates.

For every MailItem, there exist the functions isAJunkMail(), isAPersonalMail(), isABill(), and isAMagazine(). For convenience' sake, these return either true or false, and for each MailItem one and only one of these "accessor" functions will return true, indicating what type of mail it is.

Each type of mail has its respective MailContainer. Bills and PersonalMail belong in the ImportantContainer. Magazines belong in the BathroomReadingContainer. JunkMail belongs in the TrashContainer. Each MailContainer has a acceptMail(MailItem) function that adds a MailItem to the MailItems contained in that MailContainer. Since this is pseudocode, you need not worry about pointers: as soon as you have given a MailItem to the MailContainer using acceptMail(), you may begin the loop again and reassign your myMailItem to a new MailItem from the iterator.

Bills are important to me, and they must be sorted out first! It is imperative that they are all sorted out before any other sorting occurs. Don't tell me about efficiency—I want them ASAP.

\* \* \*

From your excellent UB education, you know that your pseudocode solution will need to include the following logical framework:

```
myMailSet = new MailSet()           ' You will hold my mail
myMailSet.collectTodaysMail()       ' Checks the mail

iterator = myMailSet.getIterator()  ' Gives us an iterator over that mail. If you've got some time and
    ' want to acceptMail() your BathroomReadingContainer some, an explanation
    ' of iterators is at http://en.wikipedia.org/wiki/Iterator . It's outside
    ' the scope of this course, however.

myImportantContainer = new ImportantContainer()
myBathroomReadingContainer = new BathroomReadingContainer()
myTrashContainer = new TrashContainer() ' Sets up containers for my mail to go to.

do while iterator.hasNext() ' Glossary, p. 802; an example of this is on p. 223. The concept is that
```

```

        ' code within the while loop will execute while the condition given is
        ' true: here, while new MailItems exist.

myMailItem = iterator.getNextMailItem() ' assignment of our mail item to be sorted

if myMailItem.isABill() ' Glossary, p. 805; an example of selection is on p. 135, and a syntax
    ' flowchart is on p. 134. Chapter 7, in particular pp. 131-36, cover
    ' selection in-depth. It's also a topic for Exam 1.

        ' code to handle a Bill. Remember, sort out Bills first! Save the rest.

Loop ' this indicates the end of the "do while" loop.

' At this point, Bills should be sorted out. Now for the rest...
iterator.reset()

do while iterator.hasNext()
    myMailItem = iterator.getNextMailItem()

    if myMailItem.isAJunkMail()

        ' code to handle JunkMail.

    else if myMailItem.isAMagazine()

        ' etc.

    ' . . .
Loop
' After this, all MailItems should be successfully sorted. So, we're done.

```

Using the above framework, functions from your supervisor, and their statement of the described functionality, fill in the remainder of this sorting program and (as an exception to general practice) remove all comments from the code for the version you submit. If something with regard to the sorting algorithm is unclear, or your supervisor's statement of desired functionality lacks a detail that you require, make note of it in a comment (*i.e.* prefaced with a ' ) and indicate the reasonable assumption you made to resolve it.

Because this may be easier for you to edit in your preferred IDE of Visual Studio 2008 Express, I've provided `Logic.vb` , which contains the pseudocode framework given above without comments. You may use this as a starting point for your pseudocode solution.

Note that this is a pseudocode assignment and not a Visual Basic assignment. You do not need to adhere to the syntactical peculiarities of VB for this—you'll note that mine lacks any declarations because I have been using Python too long. You do have to provide an internally consistent, logical ordering system for your pseudocode. It should be apparent immediately how to read its logical flow.

Please bring **TWO** printed (*i.e.*, paper) copies of your solution to recitation in the week that it is due.