CE HeapSort CODE

MISSING 10 Possible Points

| 10/24/2022

Attempt 1

IN PROGRESS
Next Up: Submit Assignment

Add Comment

Unlimited Attempts Allowed

Details

Sorting

CE: HeapSort CODE

Learning Objectives

- Use a heap to access elements in sorted order.
- Develop algorithms to generate a random enum and a random mail code.
- Review implementing the interface Comparable<T>

Overview

In this CE, you will create a class Mail that allows us to create mail objects that are defined by a mail code and a type that indicates the urgency of the delivery. Mail also implements Comparable<Mail>, which enables us to sort mail objects by their natural order.

You will also write a program that creates a heap of random mail objects and then simulates the delivery by taking out the most urgent mail items first.

Previous (https://slcc.instructure.com/courses/817632/modules/items/18753017)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18753020)

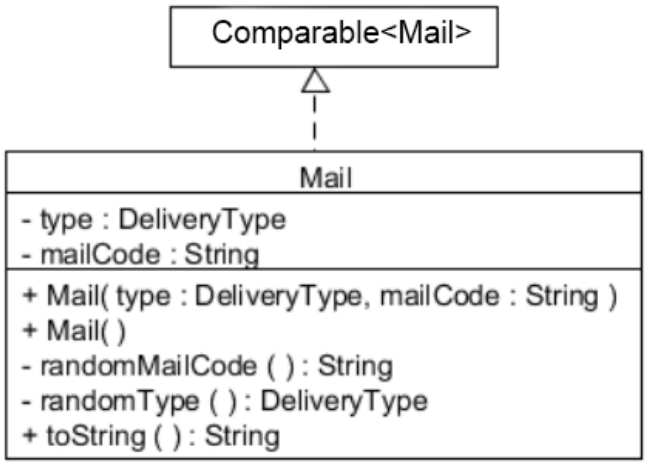
Instruction

Implement the program described below using only classes from algs4, Comparable<E>, and class Arrays. Classes from java.lang (classes that don't require an import statement) are always allowed.

- Create a package called **ceMail**.
  It should include the following four classes: **DeliveryType,** which is an enum, **Mail**, **MailTestClient,** which is provided, and **DemoHeap,** which includes the main method.
- **Enum DeliveryType:**
  This enum class includes no fields nor methods, only the following five enum constants: GROUND, AIR, PRIORITY, TWO_DAY, and ONE_DAY.

  Note that all enums are Comparable. That means they already have a compareTo method. The order depends on the order in which the enum constants are declared.
  http://www.java2s.com/Tutorials/Java/Enum/Java_Enum_compareTo.htm (http://www.java2s.com/Tutorials/Java/Enum/Java_Enum_compareTo.htm)
- **Class Mail:**
  Implement class Mail based on the UML class diagram below.

  In order to generate a random number, use the method uniform from class StdRandom.



The field mailCode should store a five-letter code that consists of all upper-case letters. E.g., YBGLC (This is not quite how the US Postal Service works, but it is close enough for

Previous (https://slcc.instructure.com/courses/817632/modules/items/18753017)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18753020)

The **parameterless constructor** initializes the fields with a random type and a random mail code (a string consisting of 5 random upper-case letters)

The method **toString** should return a string of the following format:
`{mailcode}({type})`
Where {mailcode} and {type} are substituted by the corresponding field values.
E.g., `YRGLC(GROUND)`

The method **compareTo** compares two Mail objects based on the type.
If the types are the same, the Mail objects are compared by mailCode.

Once you are done implementing the enum DeliveryType and the class Mail you can test your code using **MailTestClient.java (https://slcc.instructure.com/courses/817632/files/135713704/download?wrap=1) (https://slcc.instructure.com/courses/817632/files/135713704/download?download_frd=1)** . Verify that the elements are sorted by type first and within the same type by the mail code. Here is the ***expected output*** to compare with.

- **Class DemoHeap:**
  In this class, you will sort Mail objects. However, rather than using Sedgewick's class Heap, which includes a sort method, we'll use a heap and remove the elements one by one.

  Choose either a **MinPQ** or a **MaxPQ** to create the heap so that the most urgent mail items get removed first. Fill the heap with 25 randomly generated Mail objects. Print the random mail objects as you add them to the heap.

  Simulate the delivery of the mail objects by removing them one by one from the priority queue. The most urgent mail items need to be delivered first.

  Match the format of the sample output below, including the titles, dashes, linebreaks, etc.

Sample Output

Previous (https://slcc.instructure.com/courses/817632/modules/items/18753017)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18753020)

```
25 random mail objects:
-----------------------
LHBRQ(AIR)
JLHYU(ONE_DAY)
PZYNW(AIR)
RVCMY(ONE_DAY)
OLXVM(PRIORITY)
KUQWU(GROUND)
RRPYH(GROUND)
ZNARX(AIR)
AQTBI(ONE_DAY)
TNISD(ONE_DAY)
XTOSS(GROUND)
VLBUV(AIR)
GFUZF(ONE_DAY)
WWNJZ(AIR)
MHHSM(PRIORITY)
OHCJK(PRIORITY)
JIRID(GROUND)
UCHYZ(PRIORITY)
OHSRG(AIR)
TKJFR(PRIORITY)
RMJJD(ONE_DAY)
HYCHX(PRIORITY)
VTXNE(AIR)
FRJHL(TWO_DAY)
VTMLU(PRIORITY)

Mail Delivery:
--------------
TNISD(ONE_DAY)
RVCMY(ONE_DAY)
RMJJD(ONE_DAY)
JLHYU(ONE_DAY)
```

Previous (https://slcc.instructure.com/courses/817632/modules/items/18753017)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18753020)

```
TKJFR(PRIORITY)
OLXVM(PRIORITY)
OHCJK(PRIORITY)
MHHSM(PRIORITY)
HYCHX(PRIORITY)
ZNARX(AIR)
WWNJZ(AIR)
VTXNE(AIR)
VLBUV(AIR)
PZYNW(AIR)
OHSRG(AIR)
LHBRQ(AIR)
XTOSS(GROUND)
RRPYH(GROUND)
KUQWU(GROUND)
JIRID(GROUND)
```

Submission

Create a screen recording following the **guidelines for lab recordings (https://slcc.instructure.com/courses/817632/pages/guidelines-for-ce-recordings)** .
The video should be **25-50 seconds** long.
Post the video.

(https://slcc.instructure.com/courses/817632/modules/items/18753011)

(https://slcc.instructure.com/courses/817632/modules/items/18753012)

(https://slcc.instructure.com/courses/817632/modules/items/18753013)

(https://slcc.instructure.com/courses/817632/modules/items/18753014)

(https://slcc.instructure.com/courses/817632/modules/items/18753016)

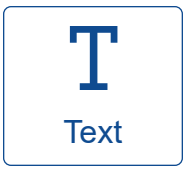
(https://slcc.instructure.com/api/v1/courses/817632/module_item_redirect/18753017)

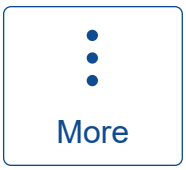
Previous (https://slcc.instructure.com/courses/817632/modules/items/18753017)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18753020)

Choose a submission type





Previous (https://slcc.instructure.com/courses/817632/modules/items/18753017)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18753020)