

## Тема 10. Колекции, алгоритми

1. Да се състави интерфейс IEmail с методи:  
за сравняване на текстови съобщения.

```
public int compareToId(String id);  
public int compareToMessage(String mess);
```

2. Да се състави абстрактен клас EMailMessage  
с член променлива String strEMailMessage;

Конструктори:

по подразб.

експлицитен

Абстрактни методи:

```
setEMailMessage(String mess); // изпращане на съобщение
```

```
getEMailMessage(); // получаване на съобщение
```

3. Да се състави публичен клас EMailImpl

EMailImpl наследник на EMailMessage имплементира IEmail, Comparable

Частни:

номер число

идентиф. низ

пощ. Провайдер низ

съобщение низ

Пример:

```
1-name1@yahoo.com:=abcd trs bak
```

Конструктори

a) num, id, provider, mess

б) по общ стринг, напр:

```
1-name1@yahoo.com:=abcd trs bak
```

-, @, := са разделители на елементите

Методи:

set/get

// създава стринг по параметрите: num, id, provider, mess

// сравнител по идентиф.

// сравнител по съобщ.

// стрингова интерпретация

// за равенство

4. Да се добави интерфейс IPost с методи:

· за образуване на e-mail съобщения към файл-параметър (compose);

· за получаване на e-mail съобщения от файл-параметър (getMessage).

5. Да се добави клас PostBox, имплементиращ интерфейса IPost за съхраняване  
на

колекции от писма (EMailImpl) - inColl входни, outColl изходни.

Частни членове от тип Collection: inColl - за входните писма; outColl - за  
изходните.

Конструктор-създава ArrayList имплементации на колекциите.

Методи:

-за получаване на членовете - getIn, getOut;

-за сортиране на входната колекция по идентификатор - providerIdSort() -  
връща

сортиран списък List<EMailImpl>

-за сортиране на входната колекция по текста на съобщението - messageSort() -  
връща

сортиран списък List<EMailImpl>

-за запълване на изходната колекция от подадена като параметър колекция

```
fillOut(Collection<EMailImpl> fromColl);
```

-интерфейсните методи getMessage, compose, ..

6. Да се добави интерфейс IOInterface с методи:

за четене на e-mail съобщения: public void read();

за запис на e-mail съобщения: public void write();

7. Да се създаде клас PostBoxTree, имплементиращ интерфейса IOInterface за

съхраняване на колекции от писма (EMailImpl) - inColl входни, outColl изходни.  
Частни членове от тип Collection: inColl - за входните писма; outColl - за изходните.  
Конструктор-създава TreeSet имплементации на колекциите.  
Методи:  
-за получаване на членовете - getIn, getOut;  
-за сортиране на входната колекция по идентификатор - providerIdSort() - връща сортиран списък List<EMailImpl>  
-за сортиране на входната колекция по текста на съобщението - messageSort() - връща сортиран списък List<EMailImpl>  
-за запълване на изходната колекция от подадена като параметър колекция fillOut(Collection<EMailImpl> fromColl);  
-интерфейсните методи read, write...  
8. Да се имплементира допълнително интерфейса IPost  
Главна функция  
-създава обект от PostBox;  
-зарежда от файл "dataIn.txt" посредством интерфейса с данни. Данните във файла са по формат, в който ":" е разделител между Id и Text, например:  
7-AliBaba@yahoo.bg:=aTest Message 1  
2-AliBaba1@abv.bg:=cvTest Message 2  
3-AliBaba2@gmail.bg:=tyTest Message 3  
4-asdbaasdtkir3@abv.bg:=aaTest Message 4  
5-baasdtkir4@post.bg:=baTest Message 5  
6-asdAliBaba5@abv.bg:=aiTest Message 6  
-Сортира по Id, извежда;  
-Сортира по Text, извежда;  
-Зарежда изходната колекция от входната;  
-Изпраща изходната във файл "dataOut.txt" посредством интерфейса.