

Introdução à Programação

Exame Escrito

Paulo Borba
Centro de Informática
Universidade Federal de Pernambuco

29 de abril de 2002

Questão 1 Muitos sistemas de leitura de *email* permitem filtrar as mensagens recebidas, de forma que mensagens indesejadas são automaticamente apagadas. Para saber se uma mensagem deve ser apagada, estes sistemas armazenam uma lista de filtros. Existem filtros básicos, como o **Contém**, que indica que uma mensagem deve ser apagada se uma parte dela contiver uma dada *string*. Existem também filtros compostos, como o **E**, que contém dois outros filtros (básicos ou compostos) e indica que uma mensagem deve ser apagada caso seus dois filtros indiquem o mesmo. Um outro tipo de filtro composto, o **Não**, indica que uma mensagem deve ser apagada caso o filtro que ele contém indique o contrário.

Seguindo os princípios de qualidade de software, defina em Java os tipos **Filtro**, **Contém**, **E** e **Não** para representar estes conceitos. Todo filtro deve ter o método **filtra**, que retorna um booleano indicando se a mensagem passada como parâmetro deve ser apagada. O construtor de **Contém** recebe como parâmetro a *string* a ser procurada e um código inteiro indicando em que parte (**To**, **From** ou conteúdo) da mensagem procurá-la. Assuma que o tipo **Mensagem** já foi definido e contém métodos **get** para cada um de seus atributos do tipo *string*: **to**, **from**, e **conteúdo**. O método **indexOf** da classe **String** retorna **-1** caso a *string* recebida como parâmetro não faça parte da *string* na qual o método foi executado. □

Questão 2 Assumindo que a classe **SistemaDeEmail** tem os atributos **listaFiltros** e **emailsRecebidos**, ambos do tipo `java.util.List`, defina nesta classe o método **filtrar**, que remove de **emailsRecebidos** todas as mensagens que devem ser filtradas por algum filtro de **listaFiltros**. O método “**Iterator iterator()**” de **List** pode ser usado para “navegar” pela lista, elemento a elemento, já que o tipo **Iterator** contém os seguintes métodos:

- `boolean hasNext()`, “returns true if the iteration has more elements”;
- `Object next()`, “returns the next element in the iteration”;
- `void remove()`, “removes from the underlying collection the last element returned by the iterator”.

Siga os princípios de qualidade de software. □

Questão 3 Considere as seguintes funções em C:

```
void f(int *x, int *y, int *z) {
    int tmp = *x; *x = *y; *y = *z; *z = tmp;
}
main() {
    int x = 6; int y = 1; int z = 3; f(&x,&y,&z);
}
```

Indique quais seriam os valores de **x**, **y** e **z** após

- a execução da chamada da função **f** em **main**;
- a execução da chamada **f(&y,&y,&y)**, ao invés de **f(&x,&y,&z)**, em **main**.

Altere a função **f** para que o resultado da chamada **f(&x,&y,&z)** seja atribuir o valor de **x** tanto a **y** quanto a **z**. □