

3º Trabalho para casa da unidade curricular de Introdução à Programação, 2010/2011

(actualizado a 2010/12/10, às 15h00)

Regras: Este trabalho é **individual** e realizado ao abrigo do código de ética dos estudantes do DI, disponível na página de IP. O trabalho é entregue em papel.

Data de entrega: 14 de Dezembro de 2010, até às 11h30, na secretaria do DI.

Enunciado

Nas aulas de IP programou vários métodos da classe `ContactBook`. Uma das utilizações frequentes dos livros de contacto, em formato electrónico, é na lista de contactos de um telemóvel. Frequentemente, podemos pesquisar todos os nomes que comecem por um determinado prefixo. Por exemplo, se o prefixo for "A", podemos encontrar nomes como "Adelaide", "Adriano", "Afonso", "Alexandre", "Ana", "Anabela", "André", "Antero", e "Avelino". Se o nosso prefixo for um bocadinho maior, a filtragem de nomes será mais precisa. Por exemplo, se em vez de "A" usarmos "An", para a mesma lista, teremos apenas "Ana", "Anabela", "André" e "Antero". Se o prefixo fosse "Ana", restariam apenas "Ana" e "Anabela". Na classe `ContactBook` viu 3 métodos de percurso, também conhecidos como iteradores, que permitiam visitar, um por um, todos os elementos do livro de contactos. Neste TPC tem que adicionar à classe `ContactBook` os seguintes métodos de percurso (iteradores), que permitem visitar, um por um, todos os contactos cujo nome começa com um determinado prefixo:

```
/*
 * Permite iniciar um percurso no livro de contactos, considerando apenas os contactos
 * que começam pelo prefixo prefix.
 */
public void initIterationWithPrefix(String prefix);

/*
 * Devolve o contacto "corrente" com um determinado prefixo (o prefixo a usar é o que
 * foi estabelecido quando a operação initIterationWithPrefix() foi invocada) no
 * percurso, e passa para o contacto seguinte que também tenha o mesmo prefixo.
 */
public Contact nextWithPrefix();

/*
 * Indica se (ainda) existe um contacto "corrente".
 */
public boolean hasNextWithPrefix();
```

Para ilustrar a utilização de iteradores adicione o método de teste

```
listAllContactsWithPrefix à classe Main:
/*
 * Método que escreve a listagem de todos os contactos contidos em book,
 * e começados pela String prefix na consola.
 */
public static void listAllContactsWithPrefix(String prefix, ContactBook book);
```

Implemente o método `listAllContactsWithPrefix` usando os 3 métodos de percurso referidos acima. Para testar, crie uma classe `Main`, contendo o programa principal (na função `main`), que deve invocar o método `listAllContactsWithPrefix`. O seu programa de teste deve criar um livro de contactos com dados lidos a partir de um ficheiro, bem como pedir ao utilizador um prefixo. De seguida o seu programa deve listar todos os contactos do livro que comecem pelo prefixo indicado pelo utilizador, usando o método `listAllContactsWithPrefix`.

Sugestão: para verificar se um nome de um contacto começa por um determinado prefixo, use o método `startsWith`

```
public boolean startsWith(String prefix)
```

da classe `String`. Veja a documentação do método em

[http://download.oracle.com/javase/6/docs/api/java/lang/String.html#startsWith\(java.lang.String, int\)](http://download.oracle.com/javase/6/docs/api/java/lang/String.html#startsWith(java.lang.String, int))

Material a entregar

Documento, em papel, com as seguintes secções:

1. Identificação do aluno, com o nome completo, número de aluno, turno prático em que está inscrito (P1, P2, ..., P8) e o nome do docente do turno prático.
2. Listagem, manuscrita ou impressa, de parte da classe `ContactBook`, incluindo as três operações de iteração e as variáveis e constantes (se for caso disso) que estas operações possam manipular.
3. Listagem, manuscrita ou impressa, da função estática `listAllContactsWithPrefix`, bem como de uma função `main` que a exercite, dentro da classe `Main`.

Erros frequentes a evitar:

- Variáveis desnecessárias
- Declaração de variáveis de instância públicas, ou de métodos da interface do objecto privados
- Código muito confuso
- Identificadores (de classes, constantes, variáveis, métodos, parâmetros) incompreensíveis, ou pouco expressivos
- Inexistência de constantes, ou não utilização das existentes
- Ausência de comentários (que fazem mesmo falta)

Aviso MUITO IMPORTANTE:

Este trabalho de casa é um elemento de avaliação individual, elaborado ao abrigo do código de ética dos estudantes da LEI. Como trabalho individual que é, **não pode ser realizado em grupo**. É expressamente **PROIBIDO** receber ajuda de colegas, quer estejam ou não a realizar a cadeira este ano, de explicadores, ou de outras pessoas que não os docentes da cadeira, na realização de trabalhos de avaliação. A violação destas regras é considerada uma tentativa de fraude, a tratar de acordo com o acima citado código de ética dos estudantes da LEI. Para seu bem, respeite escrupulosamente esta regra.