



## Programação 2

2º Semestre

### Folha de Exercícios 6

Listas (implementação baseada em vectores)

Efectue as tarefas de programação descritas abaixo, usando a linguagem C++ em ambiente Linux.

1. Implemente e teste os seguintes métodos adicionais para a classe **VList**:

a) Determinar o comprimento de uma lista.

```
int VList<Object>::length()
```

b) Determinar o número de elementos de uma lista iguais a um objecto `obj` especificado.

```
int VList<Object>::countEqualsTo(Object obj)
```

c) Eliminar os elementos de uma lista que existem em outra lista (`lst`) especificada.

```
void VList<Object>::remove(const VList<Object> &lst)
```

d) Juntar uma lista ao fim de outra, sem modificar a lista copiada.

```
void VList<Object>::append(const VList<Object> &lst)
```

e) Intercalar numa lista os elementos de outra. A segunda lista não é modificada.

```
void VList<Object>::zip(const VList<Object> &lst)
```

Exemplo: [ a b c d ] intercalada com [ v w x y z ] passaria a conter [ a v b w c x d y z ].

2. Implemente e teste os seguintes métodos adicionais para a classe **VListItr**:

a) Avança para o próximo elemento da lista de índice par.

```
void VListItr<Object>::nextEvenIndex()
```

b) Avança para o próximo elemento da lista de índice ímpar.

```
void VListItr<Object>::nextOddIndex()
```

c) Avança para o próximo elemento da lista igual a um objecto `obj` especificado.

```
void VListItr<Object>::nextEqualTo(Object obj)
```

3. Uma estratégia diferente da apresentada para remover elementos de uma lista usando a classe **VList** é a denominada “remoção preguiçosa”. Na remoção de um elemento, apenas marcamos este como apagado (usando um atributo adicional). O número de elementos apagados e não apagados da lista é guardado como informação (membros) da classe. Quando o número de elementos apagados e não apagados da lista são iguais, é realizada a operação de remoção efectiva dos elementos marcados como apagados.

a) Quais as vantagens e desvantagens da estratégia de “remoção preguiçosa”.

b) Implemente a estratégia de remoção preguiçosa na classe **VList**.

4. A classe **VList** permite inserir um novo elemento na lista a seguir a um elemento indicado por um interador. Define um novo método

```
void VList<Object>::insertBefore(const Object &x, VListItr<Object> pos)
```

que insere o novo elemento antes do elemento referenciado pelo iterador.