

Aula Prática 10

Resumo:

- Listas ligadas.
- Listas bi-ligadas.

Exercício 10.1

Implemente uma pilha e uma fila com uma lista ligada (simples). Infira a interface destes módulos através do programa `p101.java` fornecido.

Exercício 10.2

Implemente um novo módulo com as funcionalidades de um *array* mas cuja representação interna é uma lista ligada simples. O módulo deve ser genérico (instanciável para qualquer tipo de elemento).

A interface do módulo deve ser a seguinte:

<code>Array(int length)</code>	cria um <i>array</i> com índices <code>[0; length[</code>
<code>void set(int idx, T e)</code>	insere <i>e</i> no índice <i>idx</i> do <i>array</i>
<code>boolean defined(int idx)</code>	devolve verdadeiro se alguma vez existiu um <i>set</i> para esse índice
<code>T get(int idx)</code>	devolve valor existente no índice <i>idx</i> do <i>array</i> , se é aplicável se esse elemento estiver definido (função anterior verdadeira).
<code>int length()</code>	número de elementos do <i>array</i>

Crie este módulo no pacote `linkedlist` (onde foram criados os módulos do exercício anterior), e teste-o com o programa `p102.java`.

Exercício 10.3

Implemente uma fila com uma lista bi-ligada circular (ou seja, com os extremos ligados entre si). Infira a interface do módulo através do programa `p103.java` fornecido.

Exercício 10.4

Um dos problemas que os *arrays* em Java podem colocar reside no facto de o seu número de elementos ser imutável (uma vez o *array* criado). Essa limitação faz com que na presença

de problemas onde a dimensão máxima do *array* possa ter de variar em tempo de execução, sejamos obrigados ou a tirar cópias do *array* (para um novo *array* de dimensão adequada), ou a utilizar outros módulos que permitam esse comportamento dinâmico de *arrays*.

Neste exercício pretende-se desenvolver um módulo **BlockArrayInt** com uma interface tipo *array* (acesso aleatório aos elementos por intermédio de um índice inteiro), mas em que a representação interna desse *array* assenta numa lista ligada de blocos, sendo cada bloco um *array* (primitivo) de dimensão fixa (definida no construtor do módulo). Se o número de elementos por bloco for **blockSize** então os índices de **[0,blockSize-1]** apontarão para dentro do primeiro bloco (primeiro elemento da lista ligada), os índices **[blockSize,2*blockSize-1]** apontarão para o segundo, etc..

Para servir de teste ao módulo, o programa **p104.java** armazena (num objecto do tipo **BlockArrayInt**) e escreve todos os números primos encontrados até um valor colocado pelo utilizador como argumento do programa. Nesse ficheiro pode também encontrar a interface desejada para o módulo, mas onde a implementação está incompleta (apenas define um bloco).

Comece por experimentar o programa com valores menores do que 233 para perceber o funcionamento do programa. Caso utilize um valor acima de 232 verificará que ocorre um erro em tempo de execução. Implemente o módulo usando a representação interna acima descrita, por forma a que o programa funcione para qualquer valor numérico (quando o módulo estiver a funcionar experimente, por exemplo, utilizar o valor 1000000).