\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**package** estruturalistaligadas;

**public** **class** Listaligadaduplamente {

**private** Listaencadeada primeira;

**private** **int** totalDeElementos;

**private** Listaencadeada ultima;

**public** **void** adicionaNoComeco(Object elemento) {

**if**(**this**.totalDeElementos == 0) {

Listaencadeada nova = **new** Listaencadeada(elemento);

**this**.primeira = nova;

**this**.ultima = nova;}

**else** {

Listaencadeada nova = **new** Listaencadeada(**this**.primeira, elemento);

**this**.primeira.setAnterior(nova);

**this**.primeira = nova; }

**this**.totalDeElementos++;}

**public** **void** adiciona(Object elemento) {

**if**(**this**.totalDeElementos == 0) {

adicionaNoComeco(elemento);

} **else** {

Listaencadeada nova = **new** Listaencadeada(elemento);

**this**.ultima.setProximo(nova);

nova.setAnterior(**this**.ultima);

**this**.ultima = nova;

**this**.totalDeElementos++;} }

**public** **void** adiciona(**int** posicao, Object elemento) {

**if**(posicao == 0) {

adicionaNoComeco(elemento);

} **else** **if** (posicao == **this**.totalDeElementos) {

**this**.adiciona(elemento);

} **else** {

Listaencadeada anterior = busca(posicao - 1);

Listaencadeada proxima = anterior.getProximo();

Listaencadeada nova = **new** Listaencadeada(anterior.getProximo(), elemento);

nova.setAnterior(anterior);

anterior.setProximo(nova);

proxima.setAnterior(nova);

**this**.totalDeElementos++;} }

**private** **boolean** posicaoOcupada(**int** posicao) {

**return** posicao >= 0 && posicao < **this**.totalDeElementos;

}

**private** Listaencadeada busca(**int** posicao) {

**if**(!posicaoOcupada(posicao)) {

**throw** **new** IllegalArgumentException("posicao inexistente");

}

Listaencadeada atual = primeira;

**for**(**int** i = 0; i < posicao; i++) {

atual = atual.getProximo();

}

**return** atual;}

**public** **void** removeDoComeco() {

**if**(**this**.totalDeElementos == 0) {

**throw** **new** IllegalArgumentException("lista vazia"); }

**this**.primeira = **this**.primeira.getProximo();

**this**.totalDeElementos--;

**if**(**this**.totalDeElementos == 0) {

**this**.ultima = **null**; } }

**public** **void** removeDoFim() {

**if**(**this**.totalDeElementos == 1) {

**this**.removeDoComeco();

} **else** {

Listaencadeada penultima = **this**.ultima.getAnterior();

penultima.setProximo(**null**);

**this**.ultima = penultima;

**this**.totalDeElementos--; } }

**public** **void** remove(**int** posicao) {

**if**(posicao == 0) {

**this**.removeDoComeco();

} **else** **if** (posicao == **this**.totalDeElementos - 1) {

**this**.removeDoFim();

} **else** {

Listaencadeada anterior = **this**.busca(posicao - 1);

Listaencadeada atual = anterior.getProximo();

Listaencadeada proxima = atual.getProximo();

anterior.setProximo(proxima);

proxima.setAnterior(anterior);

**this**.totalDeElementos--; } }

**public** **int** tamanho() {

**return** **this**.totalDeElementos;

}

**public** **boolean** Localiza(Object elemento) {

Listaencadeada atual = **this**.primeira;

**while**(atual != **null**) {

**if**(atual.getElemento().equals(elemento)) {

**return** **true**;

}

atual = atual.getProximo();

}

**return** **false**;

}

@Override

**public** String toString () {

**if**(**this**.totalDeElementos == 0) {

**return** "()"; }

Listaencadeada atual = primeira;

StringBuilder builder = **new** StringBuilder("(");

**for**(**int** i = 0; i < totalDeElementos; i++) {

builder.append(atual.getElemento());

builder.append(",");

atual = atual.getProximo();

}

builder.append(")");

**return** builder.toString(); }}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_