**Dipartimento di Informatica e Sistemistica**
**Antonio Ruberti**

**“Sapienza” Università di Roma**

# Esercitazione 5

## *Corso di Tecniche di programmazione*

### *Laurea in Ingegneria Informatica*

***(Canale di Ingegneria delle Reti e dei Sistemi Informatici - Polo di Rieti)***

Anno Accademico 2007/2008

**Tutor: Ing.  Diego Rughetti**

# Esercizio 1

Definiamo la classe SequenzaStringhe per memorizzare una sequenza di stringhe mediante una struttura collegata lineare. Lo scheletro della classe che implementeremo è riportato di seguito.

La classe SequenzaStringhe usa una variabile di istanza nodoinit di tipo NodoLista che rappresenta il riferimento al primo nodo della lista. La variabile nodoinit sarà  posta a null per rappresentare la lista vuota.

```
class NodoLista {
  public String info;
  public NodoLista next;

  public NodoLista(String i, NodoLista n) {
     info = i; next = n;
  }
}
```

# Esercizio 1 (2)

```
public class SequenzaStringhe {
   private NodoLista nodoinit;

   // costruttore che crea una lista vuota
   public SequenzaStringhe() { ... }

   // aggiunge la stringa x in posizione k
   public void add (int k, String x) { ... }

   // concatena alla lista this una lista passata come argomento
   public void append (SequenzaStringhe l) { ... }

   // modifica il valore del k-esimo elemento con la stringa x
   public void set (int k, String x) { ... }

   // copia il contenuto della lista e restituisce una nuova lista
   public SequenzaStringhe copy () { ... }
}
```

# Soluzione (1)

```
// costruttore che crea una lista vuota
public SequenzaStringhe() {
   nodoinit = null;
}

// aggiunge la stringa x in posizione k
public void add (int k, String x)  {

    // creo il nodo generatore
    NodoLista q = new NodoLista(null,nodoinit);
    nodoinit = q;
    for (int i=0; i<k && q!=null; i++)
         q = q.next;

    if (q!=null)
         q.next = new NodoLista(x,q.next);

    nodoinit = nodoinit.next;
   }
```

# Soluzione (2)

```
// concatena alla lista this la lista passata come argomento
   public void append (SequenzaStringhe l) {

      //creo nodo generatore
      NodoLista q = new NodoLista(null,nodoinit);
      nodoinit = q;
      while (q.next!=null)
          q = q.next;
      NodoLista p = l.nodoinit;
      while (p!=null) {
         q.next = new NodoLista(p.info,null);
         q = q.next;
         p = p.next;
       }

      //elimina il nodo ausiliario
      nodoinit = nodoinit.next;
   }
```

# Soluzione (3)

```
// modifica il valore del k-esimo elemento con la stringa x
public void set (int k, String x) {
      NodoLista p = nodoinit;
      for (int i=0; i<k && p!=null; i++)
            p = p.next;
       if (p!=null)
            p.info = x;
}
```

# Soluzione (4)

```
// copia il contenuto della lista e restituisce una nuova lista
public SequenzaStringhe copy () {
    // Crea il primo nodo della lista risultato
    // (nodo generatore)
    NodoLista a = new NodoLista();
    NodoLista q = a;
    NodoLista p = nodoinit;
    while (p!=null) {
        q.next = new NodoLista(p.info,null);
        q = q.next; p = p.next;
    }
    SequenzaStringhe r = new SequenzaStringhe();
    r.nodoinit = a.next; return r;
}
```

# Esercizio 2

Completare la classe SequenzaStringhe con i seguenti metodi.

```
// elimina l'elemento in posizione k
public void remove (int k) { ... }

// elimina la prima occorrenza della stringa x
public void remove (String x) { ... }

// restituisce la stringa contenuta nel k-esimo elemento
public String get (int k) { ... }

// verifica se una stringa x e' presente nella lista
public boolean contains (String x) { ... }

// restituisce l'indice della prima occorrenza di x nella lista
// o -1 se questa non e' presente
public int indexOf (String x) { ... }

// restituisce l'indice dell'ultima occorrenza di x nella lista
// o -1 se questa non e' presente
public int lastIndexOf (String x) { ... }
```

# Esercizio 2 (2)

```
// verifica se la lista e' vuota
public boolean isEmpty () { ... }

// restituisce la lunghezza della lista
public int size () { ... }

// restituisce la lista composta dagli elementi j,j+1,...,k-1
public SequenzaStringhe subList (int j, int k) { ... }

// restituisce una stringa con il contenuto della lista
public String toString () { ... }

// restituisce un array di stringhe contenente tutti
// gli elementi della lista
public String[] toArray () { ... }
```

# Soluzione (1)

```
// elimina l'elemento in posizione k

public void remove (int k)  {
      if (k<0)
            throw new EccezioneLista("Indice fuori dei limiti");
      else if (nodoinit==null)
            throw new EccezioneLista("Lista vuota");
      else if (k==0)
            // Eliminazione in testa (vedi sezione 11.12)
            nodoinit = nodoinit.next;
      else {
            // Eliminazione in posizione qualsiasi (v. sezione 11.13)
            NodoLista p = nodoinit;
            for (int i=0; i<k-1 && p!=null; i++)
                  p = p.next;
            if ((p==null)||(p.next==null))
                  // k > dimensione della struttura
                  throw new EccezioneLista("Indice fuori dei limiti");
            else
                  // p e' il riferimento al nodo in posizione k-1
                  p.next = p.next.next;
      }
   }
```

# Soluzione (2)

```
// elimina la prima occorrenza della stringa x

public void remove (String x) {
      boolean b = false;
      if (nodoinit.info.equals(x)) {
            // eliminazione del primo nodo della lista
            nodoinit = nodoinit.next;
      }
      else {
            NodoLista p = nodoinit;
            while (p.next!=null && !p.next.info.equals(x))
                  p = p.next;
            if (p.next!=null)
                  // p e' il riferimento al nodo che precede
                  // il nodo contenente x
                  p.next = p.next.next;
      }
   }
```

# Soluzione (3)

```
// restituisce la stringa contenuta nel k-esimo elemento
public String get (int k) {
	if (k<0)
		throw new EccezioneLista("Indice fuori dei limiti");
	NodoLista p = nodoinit;
	for (int i=0; i<k && p!=null; i++)
		p = p.next;
	if (p==null)
		// k >= dimensione della struttura o lista vuota
		throw new EccezioneLista("Indice fuori dei limiti");
	else
		// p e' il riferimento al nodo in posizione k
		return p.info;
}

// verifica se una stringa x e' presente nella lista
public boolean contains (String x) {
	NodoLista p = nodoinit;
	while (p!=null && !p.info.equals(x))
		p = p.next;
	return p!=null;
}
```

# Soluzione (4)

```
// restituisce l'indice della prima occorrenza di x o -1 se questa non e' presente
public int indexOf (String x) {
	int r = -1, c = 0;
	NodoLista p = nodoinit;
	while (p!=null && r<0) {
		if (p.info.equals(x))
			r = c;
		c++; p = p.next;
	}
	return r;
}

// restituisce l'indice dell'ultima occorrenza di x o -1 se questa non e' presente
public int lastIndexOf (String x) {
	int r = -1, c = 0;
	NodoLista p = nodoinit;
	while (p!=null) {
		if (p.info.equals(x))
			r = c;
		c++; p = p.next;
	}
	return r;
}
```

# Soluzione (5)

```
// verifica se la lista e' vuota

public boolean isEmpty () {
      return nodoinit == null;
}

// restituisce la lunghezza della lista

public int size () {
      int c = 0;
      NodoLista p = nodoinit;
      while (p!=null) {
            c++;
            p = p.next;
      }
      return c;
}
```

# Soluzione (6)

```
// restituisce la lista composta dagli elementi j,j+1,...,k-1

public ListaStringhe subList (int j, int k) {
      // 1. crea il primo nodo della lista risultato
      // (nodo generatore)
      NodoLista a = new NodoLista(); a.next = null;
      NodoLista q = a;
      // 2. vai alla posizione j
      NodoLista p = nodoinit; int i=0;
      while (i<j && p!=null) {
            p = p.next; i++;
      }
      // 3. copia gli elementi fino a k
      while (i<k && p!=null) {
            q.next = new NodoLista(); q = q.next;
            q.info = p.info; q.next = null;
            p = p.next; i++;
      }
      // 4. restituisci la lista calcolata
      ListaStringhe r = new ListaStringhe();
      r.nodoinit = a.next;
      return r;
}
```

# Soluzione (7)

```
// restituisce una stringa con il contenuto della lista

public String toString() {
	NodoLista p = nodoinit;
	String ris = "( ";
	while (p!=null) {
		ris += p.info + " ";
		p = p.next;
	}
	ris += ")";
	return ris;
}
```

# Soluzione (8)

```
// restituisce un array di stringhe contenente tutti gli elementi della lista

public String[] toArray() {
	// 1. Calcola la dimensione della struttura
	int n = this.size();
	// 2. Crea l'array
	String[] a = new String[n];
	// 3. Copia le informazioni nell'array
	NodoLista p = nodoinit;
	int i = 0;
	while (p!=null) {
		a[i] = p.info;
		i++; p = p.next;
	}
	return a;
}
```

# Esercizio per casa (1)

Si vuole realizzare una classe Java PhotoGallery che rappresenta una galleria di fotografie online. Ogni PhotoGallery ha una url (rappresentata da una stringa) dove essa è situata e contiene foto, ciascuna rappresentata semplicemente come una stringa che denota il link alla foto vera e propria. Le funzionalità degli oggetti della classe sono:

- crea: che preso come parametro una stringa url, crea un oggetto PhotoGallery avente come indirizzo url e con zero foto memorizzate;
- numFoto: che restituisce il numero di foto presenti nella PhotoGallery;
- aggiungi: che presa una foto (cioè una stringa) come parametro, l'aggiunge alla PhotoGallery se non è già presente; altrimenti non fa nulla;
- elimina: che presa una foto (cioè una stringa) come parametro, la elimina dalla PhotoGallery, se presente; altrimenti non fa nulla;
- presente: che presa una foto (cioè una stringa) come parametro, restituisce true se la foto è presente nella PhotoGallery; false altrimenti;
- retituisciTutteLeFoto: che restituisce un array contenente tutte le foto (rappresentate come stringhe) presenti nella PhotoGallery;
- svuota: che elimina tutte le foto dalla PhotoGallery.

# Esercizio per casa (2)

Domanda 1.

Si realizzi la classe PhotoGallery.

Domanda 2.

Si realizzi una classe cliente della classe PhotoGallery contenente un metodo statico leggiDaFile che, dati una stringa f che rappresenta un nome di un file contenente foto (cioè stringhe) ed una PhotoGallery g, aggiunga tutte le foto presenti nel file f in g.