

Linguaggi

*Corso di Laurea Ingegneria Informatica (M-Z)
A.A. 2006-2007*

Alessandro Longheu
<http://www.diit.unict.it/users/alongheu>
alessandro.longheu@diit.unict.it

Esercitazione Collezioni in Java

1

A. Longheu – Linguaggi M-Z – Ing. Inf. 2006-2007

Esempio Scuola

- Si vuole informatizzare la gestione degli studenti di una scuola superiore. Attualmente in un file di testo sono memorizzati i dati relativi a tutti gli studenti della scuola. Ogni registrazione presente nel file contiene i dati dello studente, ed in particolare:
 - nome e cognome dello studente (stringa di caratteri)
 - anno (intero)
 - sezione (char)
 - codice identificativo studente (chiave) (stringa)
- In un secondo file sono memorizzati i giorni nei quali ciascun studente si è assentato. Tale file contiene, in ogni registrazione, le seguenti informazioni:
 - codice identificativo studente (stringa)
 - data dell'assenza (aaaammgg) (stringa)
 - giustificata (si/no) (boolean)

2



Esempio Scuola

- Tutti questi dati devono essere rappresentati in memoria centrale utilizzando una lista collegata per memorizzare gli studenti. Per ciascun studente deve essere, inoltre, presente la lista dei giorni di assenza.
- Implementare un programma Java che preveda le seguenti funzionalità:
 - Caricamento dei dati dai file nella struttura dati precedentemente definita.
 - Creazione di un file di testo che contiene tutti gli studenti (nome e cognome, anno, sezione e codice identificativo) che hanno almeno un'assenza non ancora giustificata.
 - Stampa dello studente più assenteista.

3



File: Assenza.java

```

public class Assenza{
    private String data;
    private int giustific;

    public Assenza (String d, int g){
        data = d;
        giustific = g;
    }

    public String getData() { return data;}

    public int getGiustific(){return giustific;}
}

```

4



File: Studente.java

```
import java.util.*;
public class Studente{
    private String nome;
    private int anno;
    private char sezione;
    private String codice;
    private LinkedList listaAssenze;
    public Studente (String n, int a, char s, String c){
        nome = n;
        anno = a;
        sezione = s;
        codice = c;
        listaAssenze = new LinkedList();
    }
}
```

5



File: Studente.java

```
public String getCodice(){return codice; }

public void addAssenza (Assenza a){listaAssenze.addFirst (a);}

public boolean getNonGiustifico(){
    Assenza a = null;
    for (Iterator it = listaAssenze.iterator());
        it.hasNext();){
        a = (Assenza)it.next ();
        if (a.getGiustifico() == 0) return true;
    }
    return false;
}
}
```

6



File: **Studiante.java**

```
public int getNumAssenze(){return listaAssenze.size();}  
  
public String toString(){  
    return nome + "\n" + anno + "\n" + sezione +  
        "\n\n";  
}
```

7



File: **ArchivioScuola.java**

```
import java.util.*;  
import java.io.*;  
  
public class ArchivioScuola extends LinkedList{  
    public ArchivioScuola(){} }  
    public void loadFromFile() {  
        // Caricamento del primo file  
        try {  
            String nome;  
            int anno;  
            char sezione;  
            String codice;  
            Studente s;
```

8



File: ArchivioScuola.java

```

BufferedReader fs = new BufferedReader
    (new FileReader
        ("STUDENTI.TXT"));
    nome = fs.readLine ();
    while (nome != null){
        anno = Integer.parseInt (fs.readLine());
        sezione = fs.readLine().charAt(0);
        codice = fs.readLine();
        s = new Studente (nome, anno, sezione,
            addFirst (s));
        nome = fs.readLine();
    }
    codice);
    }
    fs.close();
}

```

9



File: ArchivioScuola.java

```

catch (IOException e)
{System.out.println ("Errore lettura file
    STUDENTI.TXT");
    return;
}
// Caricamento del secondo file
try{
    String codice;
    String data;
    int giustific;
    Assenza a;
    Studente s;
    BufferedReader fa = new BufferedReader
        (new FileReader ("ASSENZE.TXT"));

```

10



File: ArchivioScuola.java

```

codice = fa.readLine();
while (codice != null) {
    data = fa.readLine ();
    giustific = Integer.parseInt

a = new Assenza (data, giustific);
s = searchStudenteByCodice (codice);
if (s != null) s.addAssenza (a);
codice = fa.readLine ();
}
fa.close ();
}
catch ( IOException e ) {
    System.out.println ("Errore lettura ASSENZE.TXT");
    return;  } }
}
}
(fa.readLine());

```

11



File: ArchivioScuola.java

```

public Studente searchStudenteByCodice (String codice)  {
    Studente s = null;
    for (Iterator it = iterator(); it.hasNext(); )  {
        s = (Studente)it.next();
        if (codice.equals (s.getCodice())) return s;
    }
    return null;
}

public String toString() {
    String res = new String ("");
    for (Iterator it = iterator(); it.hasNext(); res +=
        it.next().toString());
    return res;
}

```

12



File: ArchivioScuola.java

```
public void saveNGFile()
try {
    Studente s = null;
    BufferedWriter fa = new BufferedWriter
        (new FileWriter ("NONGF.TXT"));
    for (Iterator it = iterator();it.hasNext();) {
        s = (Studente)it.next();
        if (s.getNonGiustif()==true){fa.write (s + "");
            fa.newLine ();}
        }
        fa.close ();
    }
    catch (IOException e) {
        System.out.println ("Errore scrittura NONGF.TXT");
        return;
    }
}
```

13



File: ArchivioScuola.java

```
public void printSA() {
    Studente s = null;
    int idx = -1;
    int num = 0;
    for (Iterator it = iterator(); it.hasNext(); ) {
        s = (Studente)it.next();
        if (s.getNumAssenze() > num){
            num =s.getNumAssenze();
            idx = indexOf(s);}
        }
        if (idx == -1) System.out.println ("l'Archivio non
            caricato correttamente\n");
        else {s = (Studente)get(idx); System.out.println (s);
        }
    }
}
```

14



File: Scuola.java

```
import java.io.*;

public class Scuola{
    static ArchivioScuola archivio;
    static char menu() {
        System.out.println ("Gestione Scuola\n");
        System.out.println ("Menu principale:\n\n1. Carica
        archivio da file\n2. Crea file degli studenti con almeno un'assenza
        non giustificata\n3. Stampa studente piu' assenteista\n4. Stampa
        tutto\n5. Esci\n\n");
        System.out.print ("Inserire la scelta: ");
        char ch = '0';
        BufferedReader kb = new BufferedReader
        (new InputStreamReader (System.in));
```

15



File: Scuola.java

```
try { ch = kb.readLine().charAt(0);}
catch (IOException e) {
    System.out.println ("Errore lettura da tastiera");
    System.exit(-1);
}
return ch;
}
```

16



File: Scuola.java

```
public static void main (String args[]) {  
    archivio = new ArchivioScuola();  
    while (true) {  
        switch (menu()) {  
            case '1':  
                archivio.loadFromFile();  
                break;  
            case '2':  
                archivio.saveNGFile();  
                break;  
            case '3':  
                archivio.printSA();  
                break;  
            case '4':  
                System.out.println  
                    (archivio);  
                break;  
            case '5':  
                return;  
            }  
        }  
    }  
}
```