Unità 2 – I file binari	**MODULO L1**

(A) CONOSCENZA TERMINOLOGICA

Dare una breve descrizione dei termini introdotti:

- file binari
- file di tipi primitivi
- file di oggetti
- serializzazione

(B) CONOSCENZA E COMPETENZA

Rispondere alle seguenti domande producendo anche qualche esempio

B1) Conoscenza

1. Quali sono le classi degli *stream binari*?
2. Quali sono le classi degli *stream binari bufferizzati*?
3. Quali sono le classi degli *stream binari di tipi primitivi*?
4. Quali sono le classi degli *stream binari di oggetti*?
5. In quali casi è preferibile utilizzare *stream binari* anziché di testo?

B2) Competenza

1. Quali sono i metodi principali degli *stream binari*?
2. Come si bufferizzano gli *stream binari*?
3. Quali sono i metodi principali degli *stream binari di tipi primitivi*?
4. Quali sono i metodi principali degli *stream binari di oggetti*?

(C) ESERCIZI DI COMPRENSIONE

1. Gli stream ……. offrono la possibilità di memorizzare ……..quantità di dati, di consentire una elaborazione ……., poiché i dati sono rappresentati da sequenze di …….. e offrono anche un certo livello di protezione delle informazioni poiché non sono …….. con un comune editor.
2. Le classi principali per trattare gli stream binari sono …… per la lettura e …… per la scrittura. Da queste, derivano tutte le altre classi per gli stream bufferizzati, per gli stream di tipi …….. e per gli stream di ……
3. Gli stream binari usano le classi ……. e …….. rispettivamente per la lettura e per la scrittura. Se si vuole usare la bufferizzazione, si usano rispettivamente le classi …… e ……
4. Gli stream di dati ……. usano le classi …… e …… e possiedono metodi specifici per …….. e …….. dati di diverso tipo.
5. Gli stream binari di oggetti usano le classi …… e …… per la lettura e per la scrittura. I metodi di queste classi sono molto potenti in quanto riescono a trattare oggetti come singoli record logici. Occorre ricordare che uno stream di oggetti richiede l'interfaccia ……, che associa ad ogni oggetto un numero di …….. univoco, e che non ammette l'accesso diretto.
6. Associare le proposizioni di sinistra con le classi corrispondenti sulla destra:

1	Classe per stream binari di input	**A**	**BufferedOutputStream**
2	Classe per stream binari di tipi primitivi in input	**B**	**BufferedInputStream**
3	Classe per stream binari bufferizzati in input	**C**	**FileOutputStream**
4	Classe per stream binari di output	**D**	**DataInputStream**
5	Classe per stream binari di tipi primitivi in output	**E**	**FileInputStream**
6	Classe per stream binari bufferizzati in output	**F**	**DataOutputStream**

7. Per ciascuna delle proposizioni riportate, indicare se vera o falsa.

	Vero	**Falso**
La bufferizzazione si applica esclusivamente agli *stream* di testo		
FileInputStream è la classe per gli stream binari di input		
DataInputStream è la classe usata per gli stream di dati primitivi in input		
FileOutputStream è una classe bufferizzata per gli stream binari in output		
DataOutputStream è la classe per leggere dati primitivi da stream binario		
BufferedOutputStream è una classe per la bufferizzazione in output		
ObjectInputStream è la classe di input per gli stream binari di oggetti		

8. Completare la tabella con le classi opportune.

	Input	**Output**
Stream binari		
Bufferizzazione *stream* binari		
Stream dati primitivi		
Stream di ogggetti		

9. Completare la tabella con i prototipi dei metodi relativi alle classi degli *stream* binari.

	Sintassi	Descrizione
Lettura		
Scrittura		
Svuota buffer		
Chiusura		

10. Completare la tabella con i prototipi dei metodi relativi alle classi di bufferizzazione.

	Sintassi	Descrizione
Lettura		
Scrittura		
Svuota buffer		
Chiusura		

11. Per ciascuno dei seguenti esercizi, analizzare il codice, rilevare eventuali errori, completare parti mancanti e descriverne il funzionamento:
12. **import java.io.* ;**

a.
```
public class copiaFile
{       public static void main(String[ ] args)  throws IOException
        {   FileInputStream f = new FileInputStream("p.exe");
            int byteLetto; int nb=0;
            while ((byteLetto=f.read())!=…..)    nb++;
             f.close( );
            System.out.println (nb);
        }
}   // fine classe
```

b.
```
import java.io.* ;
public class numeri
{       public static void main(String[ ] args) throws IOException
        {           String …… = "numeri.dat";
                    int i;
                    final int N=10;
                    FileOutputStream fout= new FileOutputStream(filename);
                    DataOutputStream dataout = new DataOutputStream (…..);
                    for (i=1; i <= N; i++)
                    {     dataout.writeInt (i);   }
                    dataout……( );
        }           // fine main
}       // fine classe
```

c.
```
import java.io.* ;
public class numeri
{       public static void main(String[ ] args)
        {           String filename = "numeri.dat";
                    int i;
                    FileInputStream fin = new FileInputStream (filename);
                    DataInputStream datain = new DataInputStream(fin);
                    try
                    {     while ((i=datain.readInt())!=…….)
                                System.out.println (i);
                    }
                    catch (EOFException E)
                    {     datain.close( );      }
        }           // fine main
}       // fine classe
```

13. Completare la tabella con i prototipi dei metodi relativi alle classi degli *stream* di tipi di dato primitivo.

Lettura	**Sintassi**	**Descrizione**
Interi		
Reali		
Reali doppi		
Caratteri		
Booleani		
UTF		
Scrittura	**Sintassi**	**Descrizione**
Interi		
Reali		
Reali doppi		
Caratteri		
Booleani		
UTF		
Vari	**Sintassi**	**Descrizione**
Svuota buffer		
Chiusura		

14. Disegnare la gerarchia di classi che eredita da **Object** le sottoclassi **InputStream** e **OutputStream** con le relative sottoclassi

(D) ESERCIZI DI APPLICAZIONE

1) File binari

1. Da una rilevazione meteorologica giungono dati interi che rappresentano i millimetri di pioggia giornalieri caduti in una certa zona. Si devono accumulare in un file, volta per volta, i dati ricevuti. Scrivere un'applicazione che consenta il caricamento dei dati e la stampa della media aritmetica dei dati stessi, che indica la piovosità media della zona.
2. Scrivere un'applicazione che, byte per byte, effettui la copia di un file in un altro.

2) File di oggetti

1. Si devono registrare in un file *clienti.dat* i dati dei clienti di una società. Per ogni cliente vanno registrati il codice, nominativo, la ragione sociale, l'indirizzo, la partita IVA e le coordinate bancarie. L'applicazione deve fornire anche metodi per stampare i dati di un cliente, dato il codice.
2. Dato un file *indirizzi.dat*, contenente i dati di una serie di persone (cognome, nome, indirizzo e telefono) scrivere un'applicazione che consenta di stampare i dati di tutti i nominativi di una data città, immessa da input.
3. I dati relativi ad articoli di magazzino sono memorizzati in un file *archivio.dat*. Per ogni articoli sono indicati il codice, la descrizione, il prezzo e lo sconto, che può essere del 10%, del 20% o del 30%. Scrivere un'applicazione che stampi l'elenco di tutti gli articoli aventi come sconto un valore immesso da input.

(E) ESERCITAZIONI PRATICHE

Esercitazione n. 1

Titolo: gestione di un file di libri mediante la tecnica ad oggetti
Obiettivi: utilizzo di file di oggetti mediante serializzazione

Si deve automatizzare la gestione di libri, i cui dati sono memorizzati in un file di oggetti organizzato sequenzialmente e con accesso sequenziale. Creare una classe *Libro* in modo che per ogni libro vi siano i seguenti campi:

- *Codice* (intero);
- *Autore* (stringa);
- *Titolo* (stringa);
- *Editore* (stringa);
- *Prezzo* (double).

I metodi da prevedere siano almeno::

a. costruttori e metodi di default;
b. metodo di stampa con il formato indicato nella figura a fianco.

Effettuare il testing della classe *Libro*.

```
Codice...: 1
Autore...: Manzoni
Editore..: Russo
Titolo...: I promessi sposi
Prezzo...: 12.5
--------------------------
```

Successivamente, creare la classe *Libreria* in modo che preveda:

a. un attributo contenente il nome del file di oggetti da creare su disco;
b. un costruttore che crei il file di oggetti;
c. la creazione e la scrittura su file dei seguenti oggetti:
   Libro l1 = new Libro(1, “Manzoni”, “Russo”, “I promessi sposi”, 12.50);
   Libro l2 = new Libro(2, “Alighieri”,”Russo”, “La divina Commedia”, 15.5);
   Libro l2 = new Libro(3, “Boccaccio”,”De Agostini”, “Il Decamerone”, 18.7);
d. un metodo statico *Stampa()*che mostri a video le schede di tutti gli oggetti *l1*, *l2* ed *l3*, secondo il formato riportato nella figura a fianco.
e. un metodo *VisualizzaLibro()* che

```
C:\WINDOWS\system32\cmd.exe
Codice...: 1
Autore...: Manzoni
Editore..: Russo
Titolo...: I promessi sposi
Prezzo...: 12.5
--------------------------
Codice...: 2
Autore...: Alighieri
Editore..: Russo
Titolo...: La divina Commedia
Prezzo...: 15.5
--------------------------
Codice...: 3
Autore...: Boccaccio
Editore..: De Agostini
Titolo...: Il Decamerone
Prezzo...: 18.7
```

(E) ESERCITAZIONI PRATICHE

Esercitazione n. 2

Titolo: gestione di una semplice centralina telefonica mediante la tecnica ad oggetti
Obiettivi: utilizzo di file di oggetti mediante serializzazione

Si deve automatizzare la gestione di una semplice centralina telefonica, associando ad ogni "interno", formato da un codice di 4 cifre, il corrispondente dipendente. Ogni dipendente ha un "interno" univoco e lavora in una stanza in cui possono essere presenti altri dipendenti. Creare una classe *Numero* in modo che per ogni numero "interno" vi siano almeno i seguenti campi:

- *Numero*;
- *Stanza*;
- *Cognome*;
- *Nome*;

I metodi da prevedere siano almeno::

a. costruttori e metodi di default;
b. metodo di stampa, per la stampa completa dei dati di un singolo utente, con il formato seguente:

Interno……..: *numero interno*
Dipendente…: *Cognome Nome*
Stanza……: *numero stanza*

Effettuare il testing della classe *Numero*.

Successivamente, creare la classe *Centralina*, in modo che preveda:

a. un attributo contenente il nome del file di oggetti da creare su disco (es. "Interni.dat")
b. un costruttore che crei il file di oggetti;
c. la creazione e la scrittura su file dei seguenti oggetti:
   Numero n1 = new Numero ("0001", 1, "Martellini", "Giovanni");
   Numero n2 = new Numero ("0002", 2, "Bartali", "Simone");
   Numero n3 = new Numero ("0003", 3, "Frasca", "Lucia");
d. un metodo statico *Stampa()*che mostri a video le schede di tutti gli oggetti *n1*, *n2* ed *n3*, secondo il formato indicato sopra.
e. un metodo *VisualizzaInterno()* che, dato un numero di interno, visualizzi, se esiste, l'oggetto relativo, altrimenti produca un messaggio di errore.