

Fondamenti di informatica T-1 (A – K)

Esercitazione 8: classi e oggetti

AA 2018/2019

Tutor
Lorenzo Rosa
lorenzo.rosa@unibo.it

Esercitazione 8

Introduzione al calcolatore e Java

Linguaggio Java, basi e controllo del flusso

Eclipse ed esercizi di consolidamento

Stringhe ed array

Metodi, classi, oggetti

Ereditarietà e polimorfismo

Collezioni Java e interfacce

Esercizi d'esame

Esercizio 6/1 (tipo esame)

- L'agenzia di viaggi "Sedano" ha deciso di informatizzare la gestione degli itinerari proposti ai turisti che si recano a visitare lo stato caraibico di St. Marquez.
- Innanzitutto, vanno memorizzati i dati relativi ad ogni località turistica, ovvero il nome della località e la durata (in giorni) ed il costo (in euro) di una visita.
- Si scriva una classe Localita per l'agenzia "Sedano" che:
 1. Possieda un opportuno costruttore con parametri.
 2. Presenti opportuni metodi che permettano di accedere alle variabili di istanza dell'oggetto.
 3. Presenti il metodo `toString` che fornisca la descrizione della località.

Esercizio 6/2 (tipo esame)

Si scriva un'applicazione (classe Applicazione) per l'agenzia "Sedano" che:

1. crei un oggetto Localita, chiedendo all'utente da tastiera le informazioni necessarie.
2. Stampi a video la descrizione della località appena creata.

Soluzione 1/2

```
public class Localita {  
  
    private String nome;  
    private int durata, costo; } campi  
  
    public Localita(String nome, int durata, int costo) {  
        this.nome = nome;  
        this.durata = durata;  
        this.costo = costo;  
    } } costruttore  
  
    public String getNome() {  
        return nome;  
    } } metodi  
  
    public int getDurata() {  
        return durata;  
    } }  
  
    public int getCosto() {  
        return costo;  
    } }  
  
    public String toString() {  
        return "costo per " + nome + " per " + durata + " giorni: " + costo;  
    } }
```

Soluzione 2/2

```
import java.util.Scanner;

public class Applicazione {

    public static void main(String[] args) {

        Scanner tastiera = new Scanner(System.in);

        System.out.print("Inserisci nome localita: ");
        String nome = tastiera.next();

        System.out.print("Inserisci durata viaggio: ");
        int durata = tastiera.nextInt();

        System.out.print("Inserisci il costo: ");
        int costo = tastiera.nextInt();

        Localita l = new Localita(nome, durata, costo);

        System.out.println(l.toString());
    }
}
```

Esercizio 2/1 - Libro

Il rettore ha deciso di informatizzare la gestione dei libri dell'ateneo. Per ogni libro occorre memorizzare:

1. l'autore
2. il titolo
3. la casa editrice
4. l'anno di pubblicazione.

Si scriva una classe Libro che:

1. Possieda un opportuno **costruttore** con parametri.
2. Presenti opportuni **metodi** che permettano di **accedere** alle variabili di istanza dell'oggetto.
3. Presenti il metodo `toString` che fornisca la descrizione del libro (`String`).
4. Presenti il metodo `uguale` che, dato un `Libro` in ingresso, restituisca un booleano: `true` se tutti i campi del `Libro` in ingresso sono uguali ai propri, `false` altrimenti.

Esercizio 2/1 - Test

- E' sempre fondamentale, dopo aver scritto una classe, verificare che funzioni.
- Si scriva una classe `TestLibro`, contenente il `main`, in cui:
 1. si crei un oggetto di tipo `Libro` con dati a piacere.
 2. si chieda all'utente di inserire 'y' per continuare, oppure qualunque altro tasto per uscire.
 3. in caso di 'y', si chiedano all'utente tutte le informazioni necessarie per la creazione di un libro e si crei con esse un secondo oggetto `Libro`.
 4. se il primo libro risulta *uguale* al secondo, si stampi a video la stringa "*libro trovato*". Altrimenti, si stampi "*i libri non corrispondono*".
 5. si torni al punto 2.

Esercizio 2/2 - Biblioteca

Si scriva una classe `Biblioteca` che memorizzi le informazioni relative ai libri contenuti all'interno della biblioteca. Occorre memorizzare il **nome** della biblioteca, l'**indirizzo**, un **codice** identificativo univoco (intero) e prevedere **tre campi** di tipo `Libro`.

La classe `Biblioteca` deve inoltre:

1. presentare un opportuno costruttore (inizialmente la biblioteca non contiene alcun libro, quindi tutti i campi devono valere *null*).
2. presentare opportuni metodi che permettano di accedere alle variabili di istanza dell'oggetto (non si considerino i libri)
3. possedere un metodo `aggiungi` che, dato un oggetto `Libro`, lo inserisca in uno dei campi, purché vi sia posto, e restituisca *true*; altrimenti, restituisca *false*.
4. presentare un metodo `cerca` che, dato il nome di un autore, restituisca il primo libro di tale autore all'interno della biblioteca (*null* se non esiste).
5. possedere un metodo `cancella` che, dato il nome di un autore, **rimuova** dalla biblioteca tutti i libri di tale autore (se presenti).
6. Possedere il metodo `toString` che restituisca una stringa che fornisca una descrizione della biblioteca, compreso il numero di libri effettivamente presenti.

Esercizio 2/3 - Applicazione

Si scriva un'applicazione per l'ateneo, che:

1. Crei due oggetti `Biblioteca` (nome e altri dati a piacere);
2. Crei un oggetto `Libro`, lette da tastiera le informazioni necessarie.
3. Trovi la prima biblioteca che non contiene alcun libro dell'autore del libro di cui al punto 2.
4. Inserisca il libro di cui al punto 2 nella biblioteca di cui al punto 3 (se tale biblioteca esiste).
5. Letto da tastiera il nome di un autore, provveda a rimuovere i libri di tale autore da tutte le biblioteche.