Unità A2 – Utilizzo di classi	**MODULO L1**

(A) CONOSCENZA TERMINOLOGICA

Dare una breve descrizione dei termini introdotti:

- Array di oggetti
- Oggetti come parametri
- Oggetti come valori di ritorno
- Riferimento oggetto corrente **this**
- Attributi **static**
- Metodi **static**
- Polimorfismo
- Variabili di classe
- Metodi di classe

(B) CONOSCENZA E COMPETENZA

Rispondere alle seguenti domande producendo anche qualche esempio

B1)Conoscenza

1. Cosa sono gli attributi **static**?
2. Cosa sono i metodi **static**?
3. Come si può rappresentare graficamente un *array di oggetti*?
4. Cosa significa passare un oggetto come *parametro di un metodo*?
5. Cosa significa *oggetti come valori di ritorno di un metodo*?

B2) Competenza

1. A cosa serve il riferimento **this**?
2. Come si dichiara e come si usa un *array di oggetti*?
3. Qual è la sintassi per dichiarare un *metodo con oggetti come parametri*?
4. Qual è la sintassi per dichiarare un *metodo con un oggetto come valore di ritorno*?
5. Come si usano gli *attributi* **static**?
6. Come si dichiarano e come si usano i *metodi* **static**?

(C) ESERCIZI DI COMPRENSIONE

1. Gli attributi statici hanno la caratteristica di avere lo stesso valore in tutte le …….. di una classe. Sono utili quando si desidera rendere comune il suo valore tra tutti gli oggetti della classe. I metodi statici, invece, sono metodi che sono legati alla ……. e non agli oggetti che nascono da questa. Un classico esempio di metodi statici sono quelli della classe **Math** e della classe **String**.
2. I metodi degli oggetti che nascono dalla stessa ….. sono presenti in una sola copia in memoria, ossia sono ……., per cui necessita, quando si crea un nuovo …… della classe, uno strumento che consenta di riconoscere qual è l'oggetto che ha lanciato un dato metodo. Quest'oggetto è il …….. oggetto corrente, ed è caratterizzato dalla parola ……
3. Gli oggetti possono comparire nella ……. di un metodo: in particolare possiamo avere oggetti passati come …….. e oggetti come valori di …… . I primi seguono la sintassi dei parametri ……. nelle funzioni, i secondi si usano quando il metodo deve produrre un ……. come valore di ritorno.
4. Quando si devono trattare sequenze di oggetti dello …….. tipo è utile dichiararli in un …… di oggetti. In questo modo, ciascun oggetto viene individuato mediante un …….. che ne specifica la …… .
5. In una stessa ……. possono coesistere due firme dello stesso …….., purchè abbiano ……. diversa. Questa proprietà, tipica della OOP, si dice ……..
6. Per ciascuna delle frasi sotto riportate, indicare se si riferisca agli attributi statici o ai metodi statici.

	Metodi statici	**Attributi statici**
Hanno un valore visibile da tutti gli oggetti della classe		
Si possono inizializzare nel costruttore		
Si dicono anche variabili di classe		
Non richiedono un oggetto per essere istanziati		
Si dichiarano con **static** *tipo ident;*		
Si dicono anche metodi di classe		
Si accedono con il nome della classe		
Si istanziano con il nome della classe		

7. Si consideri il seguente codice:

```
import java.io.*;
class Rubrica
{       public static void main (String args[])
        {       final …….. MAX = 3;
```

```
                    int cod;
                    String cognome, nome;
                    String telefono;
                    Persona rubrica[] = new Persona[…….];
                    for (int i=....; i<..... i++)
                    {           Persona p = new Persona();
                                lettura cognome, nome e telefono di una persona;
                                assegnazione attributi tramite i metodi set();
                                rubrica[….] = ….;
                    }
                    stampa elementi della rubrica;
          } // end main
}
```

a. individuare le parti da scrivere al posto dei puntini;
b. scrivere il codice relativo alla funzione di lettura di cognome, nome e telefono di una persona;
c. scrivere il codice relativo all'assegnazione delle variabili lette ai corrispondenti attributi;
d. scrivere la routine di stampa dell'intera rubrica.

8. Si consideri il seguente codice:

```
public class Numero
{  private int n;
   public Numero()      {   n = 0;    }
   void setNumero (int num)      {  n = num;   }
   static void getNumero (……————)
            {  System.out.println("Dato: " + ……————); }
   void getNumero (————……..)
            {  System.out.println("Dato: " + ……————); }
}
```

a. individuare le parti da scrivere al posto dei puntini;
b. scrivere il codice per istanziare nel programma principale, il metodo *getNumero*(), sia nella forma statica che in quella non statica

9. Si consideri la seguente classe:

```
class Bicycle
{  int cadence = 0;
   int speed = 0;
   int gear = 1;
   void changeCadence (int newValue)      {  cadence = newValue; }
   void changeGear (int newValue)         {  gear = newValue;  }
   void speedUp (int increment)           { speed = speed + increment; }
   void applyDown (int decrement)                {  speed = speed - decrement;}
   void printStates()                            {  System.out.println ("cadence:"+cadence+
                                                     " speed:"+speed+" gear:"+gear);  }
}
```

Scrivere un metodo per ciascuna delle seguenti richieste:
a. creazione di due oggetti *bike1* e *bike2* di classe Bicycle;
b. aumento di 10 del valore della velocità di *bike1*;
c. impostare la prima marcia su *bike1*;
d. visualizzare lo stato di *bike1*;
e. impostare a 50 la cadenza di *bike2*;
f. aumentare di 10 la velocità di *bike2*;
g. impostare la seconda marcia su *bike2*;
h. impostare a 40 la cadenza di *bike2*;
i. aumentare di 10 la velocità di *bike2*;
j. impostare la terza marcia su *bike2*;
k. visualizzare lo stato di *bike1*;

(D) ESERCIZI DI APPLICAZIONE

Realizzare una applicazione che risolva ciascuno dei seguenti problemi

1. Progettare una classe *Numeri* che consenta di memorizzare in un vettore di oggetti, una serie di numeri interi casuali. Dopo aver fornito i metodi di *default*, prevedere metodi per il calcolo della moda, della mediana e di altri parametri statistici rilevanti.
2. Scrivere un'applicazione che, data una serie di 10 oggetti di classe *Punto*, conti quanti di essi hanno ascissa uguale a 3 e quanti ordinata pari a 2.
3. Scrivere un'applicazione che facendo uso della classe *Punto*, sia in grado di mantenere un contatore degli oggetti creati.
4. Progettare e realizzare un programma che consenta di effettuare le quattro operazioni aritmetiche su numeri razionali, espressi come frazioni, utilizzando una classe *Frazione* opportunamente costruita.

(E) ESERCITAZIONI PRATICHE

Esercitazione n. 1

Obiettivi: dichiarazione ed utilizzo di array di oggetti.
Problema: creare un'applicazione per la gestione di un elenco di libri.

1) Attivare l'ambiente dsi sviluppo (TextPad, Eclipse, ecc)
2) Creare un nuovo file e salvarlo come Libro.java. Scrivere il codice per implementare la classe Libro avente come attributi:
   a. codLibro
   b. nCopie
   c. autore
   d. editore
   e. titolo
   f. prezzo

   e come metodi
   a. due costruttori, uno con parametri ed uno senza;
   b. getCopie(), getAutore(), getEditore(), getTitolo(), getPrezzo()
   c. set()Copie(), setAutore(), setEditore(), setTitolo(), setPrezzo()
3) Compilare file *Libro.java* fino ad ottenere un codice corretto
4) Creare un nuovo file e salvarlo con il nome *TestLibro.java* e implementarlo in modo che:
   a. crei un array *elenco[ ]* di 5 oggetti di classe *Libro*;
   b. carichi i 5 oggetti con valori degli attributi riportati nella seguente tabella

Oggetto	codLibro	nCopie	autore	editore	titolo	prezzo
Elenco[0]	1210	2	A. Manzoni	Bompiani	I promessi sposi	25.50
Elenco[1]	1250	3	D. Alighieri	Einaudi	La divina commedia	22.30
Elenco[2]	1349	1	A. Lorenzi	ATLAS	Corso di Java	18.30
Elenco[3]	825	2	M. Pazzaglia	Cedam	Letteratura italiana	24.70
Elenco[4]	944	4	G. Callegarin	Cedam	Informatica 1	20.00

   c. legga da input il codice di un libro e, se presente, aggiorni l'attributo *prezzo* con un valore *nuovoPrezzo* letto da input;
   d. stampi l'elenco completo dei libri presenti;
   e. dato il codice di un libro, se presente, ne stampi il titolo, il prezzo e il numero di copie.

(E) ESERCITAZIONI PRATICHE

Esercitazione n. 2

Obiettivi: dichiarazione ed utilizzo di array di oggetti. Utilizzo di membri statici
Problema: creare un'applicazione di geometria analitica elementare.

1) Attivare l'ambiente dsi sviluppo (TextPad, Eclipse, ecc).
2) Creare un nuovo file e salvarlo come *Punto.java*. Scrivere il codice per implementare la classe *Punto* avente come attributi:
   a. x (ascissa)
   b. y (ordinata)

   e come metodi
   a. due costruttori, uno con parametri ed uno senza;
   b. getx(), gety();
   c. setx(), sety(), stampaPunto(), distanza(), puntoMedio(), proiezionex(), proiezioney().
3) Implementare i metodi con opportune scelte degli eventuali parametri.
4) Compilare file *Punto.java* fino ad ottenere un codice corretto.
5) Creare un nuovo file e salvarlo con il nome *testPunto.java* e implementarlo in modo che:
   a. crei due oggetti di classe *Punto*, P1(3, 2) e P2 (0, 0) e ne stampi le coordinate nel formato consueto "P (ascissa, ordinata)";
   b. stampi la distanza P1P2 (verificare il risultato con il calcolo manuale);
   c. calcoli e stampi l'oggetto P3, punto medio tra P1 e P2 (verificare il risultato con il calcolo manuale).
6) Modificare la classe *Punto* aggiungendo due metodi statici:
   a. distanza (Punto P1, Punto P2)
   b. puntoMedio (Punto P1, Punto P2)

   e verificare che l'applicazione *testPunto* dia gli stessi risultati di quanto svolto al punto 5), sempre con i due oggetti P1(3, 2) e P2 (0, 0).