Esercitazione 12 - Simulazione d'esame - JAVA 6

Azzolini Damiano - damiano.azzolini@unife.it
Fraccaroli Michele - michele.fraccaroli@unife.it

Tutorato Fondamenti di Informatica - Modulo B

Simulazione d'esame 1 - C1 I

Esercizio da realizzare in un UNICO FILE.java Nella soluzione, *prediligere il maggior riutilizzo di codice e la maggiore protezione possibile*.

- Si realizzi una classe **Persona** che ha due attributi: **nome** e **cognome** di tipo **String**. La classe codifica un metodo costruttore a due argomenti e definisce un metodo **toString()** che restituisce la stringa concatenata di **nome** e **cognome**.
- Si realizzi un metodo **main** in una classe **ProvaC1** che simuli l'estrazione di tre persone in modo casuale da un insieme di persone. Implementare i seguenti passi:
  - dichiarare e istanziare due vettori di stringhe come specificato:
    `String[] nomi = new String[]"aldo", "luca", "maria"; String[] cognomi = new String[]"rossi", "bianchi";`

Simulazione d'esame 1 - C1 II

- dichiarare un array **Persone** di oggetti della classe **Persona**.
- Gli oggetti di tale array sono creati, ad uno ad uno (6=3*2, in totale), scorrendo i due vettori (usare due cicli for annidati, uno per nomi, uno per cognomi);
- Si considerino le prime tre persone dell'array **Persone** e, usando il metodo **toString()**, se ne stampi il valore su un file di testo chiamato **persone.txt**.

Esempio di output su **persone.txt**:

```
nome:  aldo cognome:  rossi
nome:  aldo cognome:  bianchi
nome:  luca cognome:  rossi
```

Simulazione d'esame 1 - C2 I

Esercizio da realizzare in un UNICO FILE.java Nella soluzione, *prediligere il maggior riutilizzo di codice e la maggiore protezione possibile*.

Si realizzi una classe **Carta** che rappresenta una carta da gioco. Tale classe ha due attributi: **valore** e **seme** di tipo **String**. La classe codifica un metodo costruttore a due argomenti e definisce un metodo **toString()** che deve restituire la stringa concatenata di valore e seme.

Si realizzi un metodo main in una classe **ProvaC2** che dichiari un array **mazzo** di 4 oggetti istanze della classe **Carta**:

- Gli oggetti di tale array devono rappresentare le seguenti carte: asso di cuori (valore: asso e seme: cuori)
  due di quadri
  tre di picche
  cinque di quadri

Simulazione d'esame 1 - C2 II

- Si considerino le prime due carte del mazzo e, usando il metodo **toString()**, se ne stampi il valore su un file di testo chiamato **output.txt**.

Esempio di output su **output.txt**:

```
asso cuori
due quadri
```

Simulazione d'esame 1 - C3 I

Esercizio da realizzare in un UNICO FILE.java Nella soluzione, *prediligere il maggior riutilizzo di codice e la maggiore protezione possibile*.

Si realizzi una classe **Carta** che rappresenta una carta da gioco. Tale classe ha due attributi: **valore** e **seme** di tipo **String**. La classe codifica un metodo costruttore a due argomenti e definisce un metodo **toString()** che deve restituire la stringa concatenata di valore e seme.

Si realizzi un metodo main in una classe **ProvaC3** che dichiari un array **mazzo** di 3 oggetti istanze della classe **Carta**:

- Gli oggetti di tale array devono rappresentare le seguenti carte: asso di cuori
  due di quadri
  cinque di quadri

Simulazione d'esame 1 - C3 II

- Si considerino le prime due carte del mazzo e, usando il metodo **equals()**, si stampi un messaggio di tipo String (diverse o uguali) su un file di testo **output.txt**.

Esempio di output su **output.txt**:
`diverse`

Simulazione d'esame 2 I

Esercizio da realizzare in un UNICO FILE.java Nella soluzione, *prediligere il maggior riutilizzo di codice e la maggiore protezione possibile*.

- Si realizzi un componente interfaccia **Comparabile** in cui è definito il metodo: `public boolean maggioreDi(Object x)`
  che restituisce un booleano
- Si realizzi poi una classe **Bambino** che implementa **Comparabile**, con attributo privato di tipo stringa **nome** (il nome del bambino) e un attributo privato di tipo intero **eta** (l'età del bambino). Tale classe, codifica un metodo costruttore a 2 argomenti. La classe ridefinisce inoltre il metodo **toString()** restituendo la stringa concatenazione dei due attributi. Il metodo **maggioreDi(Object x)** restituisce **true** se l'età dell'oggetto su cui si è invocato è maggiore

Simulazione d'esame 2 II

dell'età dell'oggetto passato per riferimento (dopo l'operazione di casting). Altrimenti restituisce **false**.

- Si realizzi poi un metodo **main** in una classe **Prova** che:
  - Crei 2 oggetti **b1** e **b2** istanze della classe **Bambino.b1** deve avere **nome** uguale "elena" e **eta** 13, e b2 **nome** uguale "luca" e **eta** 14;
  - Gli attributi "elena", 13, "luca" e 14 dovranno essere letti da un file di test chiamato **dati.txt**.
  - Utilizzando il metodo **maggioreDi(Object x)** verifichi quale bambino è maggiore e lo stampi a video (tramite il metodo **toString()**).

Esempio di output su **output.txt**:

```
Il bambino maggiore:
luca 14
```

Simulazione d'esame 3 I

Esercizio da realizzare in un UNICO FILE.java Nella soluzione, *prediligere il maggior riutilizzo di codice e la maggiore protezione possibile*.

- Una classe astratta **Persona** definisce una generica persona (con attributo **nome** per il nome, ed **anno** per l'anno di nascita), il metodo costruttore per la classe, e il metodo astratto: `public abstract String whoAmI();`
- Si definisca la sottoclasse **Studente** che deriva da **Persona** e che aggiunge un attributo intero matricola per indicare la matricola dello studente. Nella sottoclasse **Studente**, si definisca un metodo costruttore a tre argomenti (nome, anno nascita e matricola) e si definisca il metodo **whoAmI()** in modo da restituire la concatenazione del nome, dell'anno di nascita e della matricola come stringa.

Simulazione d'esame 3 II

- Si realizzi un metodo **main** in una classe **Prova** che crei due oggetti **anna** e **mario** istanza di **Studente**, il primo oggetto con attributi "Anna", 1997, 12345 e il secondo oggetto con attributi "Mario", 1997, 13344.
- Chiamando il metodo **whoAmI()** sui due oggetti **anna** e **mario** stampi sul file di uscita **output.txt** le stringhe risultanti.

Simulazione d'esame 4 I

Esercizio da realizzare in un UNICO FILE.java Nella soluzione, *prediligere il maggior riutilizzo di codice e la maggiore protezione possibile*.

Si vuole leggere da tastiera una sequenza di interi positivi, memorizzarli in una lista, ordinare la lista, stampare successivamente.

- Si realizzi un metodo **main** in una classe **Prova** che:
  - Dichiari un oggetto **lista** di tipo **List** e lo allochi (**new**) come istanza della classe che si ritiene più opportuna
  - Legga da input la sequenza di valori interi positivi, terminata da 0, e inserisca ciascun numero in **lista** chiamando il metodo opportuno
  - Ordini la lista (a tale fine si ricorda che la classe **Collections** fornisce vari metodi statici)
  - Stampi a video la **lista** così ordinata.