Esercitazioni di Laboratorio

3. Laboratorio 3

Esercizio 1

Obiettivo: scrivere classi di collaudo di una classe data
Scrivere un programma di collaudo Bonifico.java per la classe BankAccount (scaricabile a questo link: BankAccount.class) in cui:

- Creare due conti bancari (account1 e account2).
- Depositare in account1 1000 euro.
- Depositare in account2 100 euro.
- Stampare a video il valore del saldo di ciascun conto.
- Prelevare da account1 400 euro e depositare la stessa quantita' in account2.
- Stampare a video il valore del saldo di ciascun conto.

Abbiamo simulato un bonifico dal primo conto al secondo!

NB: per poter eserguire il tester il file BankAccount.class deve trovarsi nella cartella/directory dove c'e' Bonifico.java

La classe BankAccount e' una classe gia' compilata in bytecode (cioe' voi non potete vedere il sorgente, pensate che sia una scatola nera), con i seguenti metodi:

Costruttore:

- `BankAccount()`

Metodi pubblici *non statici*:

- `void deposit(double)`
- `double getBalance()`
- `void withdraw(double)`

La classe e' stata illustrata a lezione. **Lo scopo di questo esercizio e' utilizzare una classe creata da altre persone senza il sorgente e ricompilare il codice**, quindi non dovete creare un file BankAccount.java con il sorgente e poi compilarlo (se pero' avete problemi con il bytecode da me fornito, potete copiare la classe BankAccount.java dalle slide e compilarla per ottenere il .class).

Esercizio 2

Argomento: uso della documentazione java; conversioni tra stringhe e tipi numerici
Esplorare la Documentazione API di Java. Cercare la documentazione delle classi **String**, **Integer**, **Double**, studiarne la descrizione, in particolare studiare la descrizione dettagliata dei costruttori e metodi visti a lezione:

- Metodi della classe String: length, substring, toUpperCase, toLowerCase, ...
- Metodi delle classi Integer e Double: parseInt, parseDouble, toString, ...

Scaricare, compilare ed eseguire la classe **NumTypeTester.java**. Osservare i risultati, manipolare i dati secondo quanto suggerito nei commenti interni alla classe.

Scaricare, compilare ed eseguire la classe **StringNumTester.java**. Osservare i risultati, manipolare i dati secondo quanto suggerito nei commenti interni alla classe.

Esercizio 3

Argomento: decisioni, confronto tra numeri reali

Osservate con attenzione il codice sorgente e i commenti del programma eseguibile NumericTest.java. Compilate il programma ed eseguitelo osservando i messaggi a standard output.

Esercizio 4

Argomento: uso dell'input standard

Scrivere un programma che

- acquisisca da standard input una riga contenente tre parole o *token* (ovvero tre stringhe separate da un carattere di spaziatura), non serve controllare che nella riga ci siano piu' di tre parole, cioe' dopo aver letto i 3 token essi vengono stampati ignorando eventuali caratteri extra.
- stampi le tre parole a standard output, una per riga nell'ordine in cui sono state acquisite

Esempio: se si introduce la stringa ***uno due tre***, l'esecuzione della classe produce le seguenti stampe:

```
uno
due
tre
```

Esercizio 5

Argomento: uso dell'input standard; uso di caratteri di escape

Modificare il programma dell'esercizio precedente in maniera tale che

- le tre stringhe vengano stampate a standard output nell'ordine inverso rispetto a quello in cui sono state inserite
- la stampa delle tre stringhe venga effettuata utilizzando una singola invocazione del metodo **print** o del metodo **println**

Esercizio 6

Argomento: uso dell'input standard, costruzione di stringhe con la concatenazione

Scrivere un programma che

- acquisisca tre numeri in virgola mobile da standard input
- stampi a standard output i tre numeri e la loro somma.

Esempio: se si introduce: 1.0 2.0 3.0

il programma produce la seguente stampa:

```
1.0 + 2.0 + 3.0 = 6.0
```

Attenzione : se state lavorando su un sistema operativo che usa la lingua italiana, e usate il metodo **nextDouble** di **Scanner** per leggere i numeri in virgola mobile da standard input, i numeri vanno inseriti usando la virgola come carattere di separazione fra parte intera e parte frazionaria. Il programma invia a standard output numeri in virgola mobile in cui il carattere di separazione fra parte intera e frazionaria e` il punto. Usate Locale come visto a lezione se volete inserire i numeri in virgola mobile usando il punto come separatore.

Esercizio 7

Argomento: uso dell'input standard; manipolazione di stringhe (uso della classe String)
Scrivere un programma che

- riceva in ingresso una stringa rappresentante un aggettivo qualificativo maschile o femminile (terminante con il carattere 'o' o con il carattere 'a')
- stampi a standard output l'aggettivo inserito, con solo il primo carattere maiuscolo,
- stampi a standard output la forma diminutiva (-ino / -ina) dell'aggettivo inserito
- stampi a standard output il grado superlativo assoluto (-issimo / -issima) dell'aggettivo inserito

Ad esempio, se viene inserita la stringa "bello", il programma visualizzera` l'output
Aggettivo inserito: Bello
Forma diminutiva: Bellino
Superlativo assoluto: Bellissimo
Se invece viene inserita la stringa "cara", il programma visualizzera` l'output
Aggettivo inserito: Cara

Forma diminutiva: Carina
Superlativo assoluto: Carissima
Suggerimento: studiare la documentazione della classe String e trovare i metodi utili a risolvere il problema, **in particolare vi suggeriamo di usare i metodi substring e l'operatore + per la concatenazione**

Esercizio 8

Argomento: uso dell'input standard; operazioni tra numeri; decisioni di confronto

Progettare il programma TwoNumbers che chieda all'utente di inserire due numeri reali e ne visualizzi (senza usare la classe Math):

- la somma
- il prodotto
- il valore medio
- il valore massimo
- il valore minimo
- il valore assoluto della differenza

Esercizio 9

Argomento: realizzazioni di classi

Provate a costruire una classe MyRectangle che simuli il comportamente della classe Rectangle vista a lezione (usiamo solo interi per semplicita'). Prevedete che siano presenti 4 variabili d'istanza: due per le coordinate x,y e due per larghezza e altezza.

Implementate un costruttore:

public **MyRectangle**(int posX, int posY, int w, int h);

Implementate i metodi di accesso:

public int **getX**();

public int **getY**();

public int **getWidth**();

public int **getHeight**();

Implementate i metodi modificatori:

public void **resize**(double mult); // ridimensiona il rettangolo della quantita' mult (es. se mult=2 le dimensioni raddoppiano, se multi=0.5 si dimezzano);

public void **translate**(int dx, int dy); //"sposta" il rettangolo in posizione x+dx e y+dy

Implementate il metodo per stampare l'oggetto MyRectangle descrivendone lo stato

public String **toString**(); // restituisce una stringa che descrive il rettangolo attraverso i valori di x, y, larghezza e altezza (es: "Rettangolo: x=10, y=30, w=50, h=20")

Esercizio 10

Argomento: realizzazione di una classe di collaudo

Implementate un programma MyRectangleTester.java per testare le funzionalita' della vostra classe. Create uno o piu' oggetti MyRectangle, invocate i metodi e ispezionate i risultati stampando in output i valori delle variabili d'istanza (attraverso i metodi di accesso o attraverso il metodo toString()).

Esercizio 11

Argomento: un esercizio per imparare a pensare "out-of-the-box"
Progettare il programma PrintSelectedMonth che chieda all'utente un numero intero compreso tra 1 e 12 e visualizzi il nome del mese corrispondente, come in questo esempio di funzionamento:

Inserisci il numero del mese (1-12): 5
Maggio

NB: non potete utilizzare ne' l'enunciato IF, ne' gli array (che non abbiamo ancora visto). Il problema cosi' è abbastanza difficile, ma provateci!