

LABORATORIO DI PROGRAMMAZIONE 2008–2009
TURNO 3 (SERALE)
10.X.2008

VINCENZO MARRA

INDICE

Esercizio 1	2
Il primo programma Java: CiaoMondo	2
Tempo: 5 min.	2
Esercizio 2	2
Il secondo programma Java: CiaoMondo con il package prog.	2
Tempo: 10 min.	2
Esercizio 3	2
Sequenza di CiaoMondo.	2
Tempo: 15 min.	2
Commento 1	2
Esercizio 4	3
Eco di interi.	3
Tempo: 10 min.	3
Esercizio 5	3
Eco di stringhe.	3
Tempo: 10 min.	3
Esercizio 6	3
Eco di stringhe concatenate.	3
Tempo: 10 min.	3
Esercizio 7	3
Decorazione di stringhe.	3
Tempo: 15 min.	4
Esercizio 8	4
Le quattro operazioni (sugli interi).	4
Tempo: 15 min.	4
Commento 2	4
Esercizio 9	4
Le quattro operazioni (sui razionali).	4
Tempo: 20 min.	4
Commento 3	4

ESERCIZIO 1

Il primo programma Java: CiaoMondo.

Tempo: 5 min.

Usando la classe di sistema `System`, scrivere un programma che produca in uscita la scritta `CiaoMondo`.

ESERCIZIO 2

Il secondo programma Java: CiaoMondo con il package prog.

Tempo: 10 min.

Ripetere l'Esercizio 1 usando la classe `ConsoleOutputManager` del package `prog.io` in luogo della classe di sistema `System`.

ESERCIZIO 3

Sequenza di CiaoMondo.

Tempo: 15 min.

Modificare l'Esercizio 2 in modo da produrre il risultato seguente:

```
Ciao Mondo (1)
Ciao Mondo (2)
Ciao Mondo (3)
```

Si scriva inizialmente una versione che invochi tre volte il metodo `println` di `ConsoleOutputManager`, usando il costrutto sequenza della programmazione strutturata. Si scriva poi un'altra versione dello stesso programma che invochi una sola volta il metodo `println`. *Suggerimento.* L'invocazione `println("x\ny")` (eseguita su un oggetto di classe `ConsoleOutputManager`) produce in uscita

```
x
y
```

Commento 1. Nella chiamata `println("x\ny")`, la sottostringa `\n` è detta *escape sequence*, e rappresenta *un singolo* carattere. L'effetto della sequenza `\n` è di andare a capo: infatti, `\n` sta per *new line*. Vi sono altre utili *escape sequence*. Ne riportiamo qualcuna qui sotto.

Escape	Nome	Effetto
<code>\n</code>	newline	ritorno a capo
<code>\r</code>	carriage return	rientro del carrello
<code>\t</code>	horizontal tab	tabulazione orizzontale
<code>\b</code>	backspace	cancella il carattere precedente
<code>\"</code>	double quotes	virgolette
<code>\uHHHH</code>	unicode char	carattere in codifica Unicode

Fate qualche esperimento fino a che l'uso e il significato delle *escape sequence* riportate sopra non vi sia chiaro. Ad esempio, l'invocazione

```
println(" Pollo\t\tP\bCollo\t\tPollo}\r\u007bB")
```

produce in uscita

```
{Bollo      Collo      Pollo}
```

ESERCIZIO 4

Eco di interi.

Tempo: 10 min.

Scrivete un programma che legga in ingresso un numero intero, e lo riscriva in uscita. Usate le classi `ConsoleInputManager` e `ConsoleOutputManager` del package `prog.io`, e i metodi appropriati di tali classi. *Nota.* Cosa succede se l'utente inserisce una stringa e non un intero?

ESERCIZIO 5

Eco di stringhe.

Tempo: 10 min.

Scrivete un programma che legga in ingresso una stringa, e la riscriva in uscita. Usate le classi `ConsoleInputManager` e `ConsoleOutputManager` del package `prog.io`, e i metodi appropriati di tali classi. *Nota.* Cosa succede se l'utente inserisce un intero e non una stringa? Si confronti con l'Esercizio 4.

ESERCIZIO 6

Eco di stringhe concatenate.

Tempo: 10 min.

Scrivete un programma che legga in ingresso tre stringhe, e scriva in uscita una sola riga contenente le tre stringhe, visualizzate l'una di seguito all'altra nell'ordine in cui sono state inserite. Usate le classi `ConsoleInputManager` e `ConsoleOutputManager` del package `prog.io`, e i metodi appropriati di tali classi.

ESERCIZIO 7

Decorazione di stringhe.

Tempo: 15 min.

Scrivete un programma che legga in ingresso una stringa, e riproduca in uscita la stringa incorniciata da asterischi. Per esempio, se la stringa in ingresso è `Pinco Pallo`, il programma fornisce in uscita

```
*****  
* Pinco Pallo *  
*****
```

Usate le classi `ConsoleInputManager` e `ConsoleOutputManager` del package `prog.io`, e i metodi appropriati di tali classi. Usate inoltre il metodo `int length()` della classe `String`.

ESERCIZIO 8

Le quattro operazioni (sugli interi).

Tempo: 15 min.

Scrivete un programma che legga in ingresso due numeri interi, diciamo `x` e `y`, e produca in uscita la loro somma, la loro differenza, il loro prodotto, il loro quoziente (intero). Informate l'utente tramite opportuni messaggi del significato dei valori prodotti in uscita.

Commento 2. In questo esercizio, cosa succede se l'utente fornisce come valori in ingresso `x=1` e `y=2`? Perché? Inoltre, cosa succede succede se i valori in ingresso sono `x=1` e `y=0`? Perché?

ESERCIZIO 9

Le quattro operazioni (sui razionali).

Tempo: 20 min.

Scrivete un programma che legga in ingresso due numeri razionali, espressi come rapporto fra due interi, e produca in uscita la loro somma, la loro differenza, il loro prodotto, il loro quoziente. Informate l'utente tramite opportuni messaggi del significato dei valori prodotti in uscita. Usate per l'implementazione la classe `Frazione` del package `prog.utili`, e i metodi da essa messi a disposizione del programmatore.

Commento 3. Com'è trattato dall'implementazione della classe `Frazione` il problema della divisione per zero? Si confronti con il Commento 2.

(V. Marra) DIPARTIMENTO DI INFORMATICA E COMUNICAZIONE, UNIVERSITÀ DEGLI STUDI DI MILANO, VIA COMELICO, 39-41, I-20135 MILAN, ITALY
E-mail address: `marra@dico.unimi.it`