

Laboratorio di Programmazione 1

Simone Zaccaria
Marko Bojovic
Mauro Santoro

Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano - Bicocca

Laboratorio 4: 24-27 novembre 2014

1. Scrivere un metodo `inverti` che prende come parametro una stringa e ritorna al chiamante una stringa che ha i caratteri invertiti rispetto alla stringa originale. Ad es. `alex -> xela`

Suggerimenti:

- Si ricorda che per ora definiamo solo metodi `'static'` che possono essere chiamati molto semplicemente dal (metodo) `main` di una classe.
 - Per provare il metodo si definisca appunto una classe e un metodo `main` appropriato che:
 - (a) chieda all'utente una stringa
 - (b) chiami il metodo `inverti`
 - (c) stampi a video la stringa invertita
2. Scrivere un programma in Java che permetta all'utente di convertire temperature da Fahrenheit a Celsius e da C a F. Il programma deve permettere all'utente di continuare a convertire temperature fino a quando decide di uscire (mostrare e gestire un menu apposito a tre scelte: `F-> C`, `C-> F`, `Uscita`).

Nell'implementazione del programma si dovranno definire, almeno, i seguenti metodi:

- `convFtoC`: prende come parametro una temperatura in F e ritorna la temperatura in C
- `convCtoF`: prende come parametro una temperatura in C e ritorna la temperatura in F
- `menu`: stampa a video il menu per l'utente

Suggerimenti:

- Si ricorda che per ora definiamo solo metodi `'static'` che possono essere chiamati molto semplicemente dal (metodo) `main` di una classe; i metodi descritti sopra saranno chiamati appunto dal `main`.
 - Le formule di conversione sono le seguenti:
$$C = (F - 32) / 1,8$$
$$F = C * 1,8 + 32$$
3. Si scriva un programma in Java che permette all'utente di convertire numeri interi positivi da binari a decimale. Il programma deve permettere all'utente di continuare a convertire numeri fino a quando decide di uscire (mostrare e gestire un menu apposito a due scelte: `C-> Converti`, `U-> Uscita`).

Nell'implementazione del programma si dovranno definire, almeno, i seguenti metodi:

- `convertiBinario`: prende come parametro un numero intero positivo in notazione binaria e ritorna il numero intero in notazione decimale
- `numeroOk`: prende come parametro un numero intero e verifica che sia positivo e sia effettivamente espresso in notazione binaria (solo 0 e 1); ritorna `true` se il numero è ok, `false` altrimenti.

Suggerimenti: Si ricorda che per ora definiamo solo metodi 'static' che possono essere chiamati molto semplicemente dal (metodo) main di una classe; i metodi descritti sopra saranno chiamati appunto dal main.

4. Scrivere un programma che chiede all'utente un intero positivo e che invoca un metodo che traccia su schermo una serie di righe di asterischi in modo che compongano un mezzo quadrato, come:

```
*  
**  
***
```

con dimensione del lato (nell'esempio sopra pari a 3) pari all'intero introdotto sopra dall'utente.

5. Scrivere un programma che chiede per 3 volte (con ciclo while) una coppia di numeri float: capienza_vasca e capienza_secchio.

Ogni volta il programma dovrà invocare con tali due valori il metodo `quanti_secchi()` descritto sotto e visualizzarne il valore restituito.

Metodo `quanti_secchi()`:

- riceve due argomenti di tipo float, chiamiamoli 'a' e 'b'
- simula il ripetuto versare 'a' litri in una vasca di capacità 'b' litri
- N.B. la vasca perde ad ogni versamento 1.1 litri
- la simulazione può essere con un ciclo o un'espressione matematica
- il metodo restituisce l'intero corrispondente a quanti versamenti devo fare per riempire la vasca, ad esempio: a==2.4 b==12, restituisco 10

6. Scrivere un programma che chiede all'utente tre interi positivi continuando a chiederne altri tre fintanto che NON sono terna pitagorica (tre numeri a b c sono terna pitagorica se $a^2 + b^2 = c^2$)

Incapsulare e poi usare il test tramite un metodo che verifica tale proprietà ricevendo come argomenti i tre interi e restituendo un valore booleano: 'true' quando i tre argomenti costituiscono una terna pitagorica, altrimenti 'false'

7. Scrivere un programma Java che chiede all'utente una stringa e poi la visualizza all'interno di una cornice di asterischi:

```
es. pippo ->  
*****  
  
*pippo*  
*****
```

Si produca la visualizzazione in un metodo (invocato dal main) 'void' che riceve come argomento la stringa e che stampa la stringa circondata dalla giusta quantità di asterischi (in base alla lunghezza della stringa).