

Esercitazioni di Laboratorio

2. Laboratorio 2

Esercizio 1

Se non l'avete ancora fatto, provate a rispondere alle domande del [questionario di conoscenze pregresse](#). Vi ricordo che il questionario non ha valenza per il superamento dell'esame. Non c'è valutazione da parte della docente. I dati verranno analizzati solo in modo aggregato. Vi ringrazio per la collaborazione che per noi è molto preziosa per calibrare al meglio il corso.

Esercizio 2

Obiettivo: esercitarsi nella consultazione della documentazione Java Platform API.

Consultare con un web browser la documentazione di Java. Potete partire dalla pagina web dell'aula didattica Taliercio <https://www.adt.unipd.it/>, selezionare sulla sinistra "guide on-line", cliccare su: Java openjdk 11.0.12 documentazione delle API

Nella finestra SEARCH in alto a destra, scrivete **java.io** e cliccate sul nome del pacchetto.

Dopo aver cliccato su java.io, nella pagina appariranno tutte le classi che si trovano nel pacchetto. Consultare la documentazione per le seguenti classi (cliccando sul nome corrispondente) e rispondete alle seguenti domande. Per rispondere potete collegarvi al solito WOOLAP cliccando qui sotto:

[Questionario Lab 2](#)

e rispondere al questionario che dovrebbe apparire (non ci sono limiti di tempo e non conta per la classifica!)

- classe `PrintStream`
 - esiste un solo metodo `void println()` o più metodi con lo stesso nome?
 - in che cosa differiscono i metodi `void println()`?
 - che cosa fa il metodo `void println(String s)`?
 - e il metodo `void print(String s)`?

Nella finestra SEARCH in alto a destra, scrivete **java.lang** e cliccate sul nome del pacchetto. Cliccate poi sui nomi delle seguenti classi e osservatene la documentazione.

- classe `System`
 - Notate che tra i "fields" (=campi della classe) c'è una variabile che si chiama **out** di tipo `PrintStream`! Ecco perché la chiamiamo `System.out`
 - che cosa fa il metodo: `static void exit(int status)`?
- classe `Double`
 - Cosa fa il metodo: `static double ParseDouble(String)`?

Esercizio 3

Obiettivo: esercitarsi nella consultazione della documentazione Java Platform API e accesso a costanti definite nelle classi.

Scrivere una classe di collaudo (cioè una classe eseguibile, quindi con il metodo `main`) che invii a standard output, per ciascun tipo fondamentale di dati numerici del linguaggio Java, il massimo e il minimo rappresentabile. Ricordo che nel pacchetto `java.lang` ci sono le classi "wrapper" (=involucro) che hanno lo stesso nome (ma con la maiuscola) dei tipi fondamentali (eccetto `Integer` e `Character`) e al loro

interno le costanti MAX_VALUE e MIN_VALUE . Ricordo che a queste costanti si accede con la notazione: **NomeClasse.nomeCostante**

Esercizio 4

Obiettivo: esercitarsi nella definizione e utilizzo di variabili numeriche e stampa di risultati.

Scrivere un programma che:

- definisca una variabile per memorizzare il valore del raggio di un cerchio e assegni alla variabile il valore 10.12;
- definisca una costante per memorizzare il valore di π (pi greco) espresso con 15 cifre decimali (3.14159265358979)
- memorizzi il valore della circonferenza in un'apposita variabile
- invii a standard output il seguente messaggio:
"***La circonferenza di un cerchio di raggio e' pari a ...***"
dove al posto dei caratteri ... ci siano i valori del raggio e della circonferenza, rispettivamente
- memorizzi il valore dell'area in un'altra variabile
- invii a standard output il seguente messaggio:
"***L'area di un cerchio di raggio ... e' pari a ...***"
dove al posto dei caratteri ... ci siano i valori del raggio e della circonferenza, rispettivamente

NB: E' possibile combinare piu' stringhe e piu' variabili in un unico enunciato con l'operatore di concatenazione tra stringhe "+" (lo vedremo poi a lezione, ma per "snellire" la scrittura del codice potete usarlo gia' da adesso). Ad esempio:

```
System.out.println("Il risultato dell'operazione 1 e': "+result1+" e il risultato dell'operazione 2 e': "+result2);
```

Esercizio 5

Obiettivo: esercitarsi nella definizione e utilizzo di variabili numeriche e stampa di risultati.

Scrivere un programma che:

- definisca una variabile per memorizzare il valore del raggio di un cilindro pari a 10.12;
- definisca una variabile per memorizzare il valore dell'altezza di un cilindro pari a 10.87
- definisca una costante per rappresentare pi-greco come nell'esercizio precedente
- calcoli il valore dell'area della base del cilindro e lo memorizzi in un'apposita variabile
- memorizzi il valore del volume del cilindro in una nuova variabile
- invii a standard output il seguente messaggio: "***Il volume del cilindro di raggio ... e altezza ... e' pari a ...***" dove al posto dei caratteri ... ci siano i valori del raggio, dell'altezza e del volume del cilindro.

NB: Quando sia necessario usare la costante π , non la si definisca come nell'esercizio precedente, ma si usi la costante Math.PI definita nella classe java.lang.Math

Esercizio 6

Obiettivo: esercitarsi con operazioni divisione e modulo fra interi

In Java l'operatore / calcola il risultato della divisione, mentre l'operatore % calcola il resto di una divisione fra interi.

Esempio:

```
int m = 10;
int n = 3;
int q = m / n; //q vale 3 perche' il quoziente della divisione fra interi 10/3 e' 3
int r = m % n; //r vale 1 perche' il resto della divisione fra interi 10/3 e' 1
```

Scrivere un programma che:

- definisca una variabile costante inizializzandola a un valore compreso fra 0 e 999 (con massimo tre cifre decimali).
- invii a standard output le singole cifre del numero separandole con uno spazio: ad esempio il numero 123 verra' visualizzato come 1 2 3. Se il numero e' inferiore a 100, stampi sempre tre cifre, introducendo zeri nelle cifre piu' significative: ad esempio il numero 23 verra' visualizzato come 0 2 3.

Suggerimento per l'algoritmo da usare:

Applicare la definizione di notazione posizionale in base 10 e le proprietà delle operazioni divisione e modulo dell'aritmetica intera.

Ad esempio: $123 = 1 * 10^2 + 2 * 10^1 + 3 * 10^0$

quindi: $123/100 = 1$ (cifra più significativa) e $123 \% 100 = 23...$

oppure

$123 \% 10 = 3$ (cifra meno significativa) e $123 / 10 = 12 ...$

Esercizio 7

Obiettivo: esercitarsi con le operazioni aritmetiche

Scrivere il programma `PrintEasterDayForYear` che calcoli la data della domenica di Pasqua di un anno specifico (dichiarare una variabile di tipo intero e assegnarle il valore dell'anno che volete utilizzare). La domenica di Pasqua è la prima domenica dopo la prima luna piena di primavera e la sua data può essere calcolata con questo algoritmo, individuato da Carl Friedrich Gauss nel 1800.

- Parti dal valore assegnato all'anno y , un numero intero non negativo
- Dividi y per 19, ottenendo il resto a . Ignora il quoziente.
- Dividi y per 100, ottenendo quoziente b e resto c .
- Dividi b per 4, ottenendo quoziente d e resto e .
- Dividi $(8b+13)$ per 25, ottenendo il quoziente g . Ignora il resto.
- Dividi $(19a+b-d-g+15)$ per 30, ottenendo il resto h . Ignora il quoziente.
- Dividi c per 4, ottenendo quoziente j e resto k .
- Dividi $(a+11h)$ per 319, ottenendo il quoziente m . Ignora il resto.
- Dividi $(2e+2j-k-h+m+32)$ per 7, ottenendo il resto r . Ignora il quoziente.
- Dividi $(h-m+r+90)$ per 25, ottenendo il quoziente n . Ignora il resto.
- Dividi $(h-m+r+n+19)$ per 32, ottenendo il resto p . Ignora il quoziente.
- Pasqua è il giorno p del mese n dell'anno y .

Dopo aver provato con valori di variabili assegnati direttamente nel codice sorgente, modificare il programma in modo che l'anno y venga specificato dall'utente leggendolo da standard input e non fissato nel codice.

Esercizio 8

Obiettivo: esercitarsi con operazioni divisione e modulo fra interi

Scrivere un programma che converta un numero **intero in base 10** in una **base qualsiasi** $b > 1$. Si scelga il numero da convertire nell'intervallo chiuso $[0, b^3 - 1]$ in modo che sia esprimibile con tre cifre nella nuova base. La base b e il numero da convertire vanno dichiarati come variabili intere nel codice, dove potete assegnare loro un valore. Esempio se la base scelta è 7, si scelga il numero fra 0 e $342(7^3 = 343)$.

Suggerimento per l'algoritmo da usare: algoritmo di conversione da base decimale a base generica b (invece che dividere per 2, come visto in classe per il caso binario, si divide per b). Verificare il risultato: ad esempio se il numero decimale è 256 e la base $b=7$: $256_{10} = 514_7 = 5*7^2 + 1*7^1 + 4*7^0$

Dopo aver provato con valori di variabili assegnati direttamente nel codice sorgente, modificare il programma in modo che la base e il valore decimale da convertire vengano specificati dall'utente da standard input e non fissati nel codice.

Esercizio 9

Realizzare una classe eseguibile `ProvaStringa.java` in cui creerete un oggetto s di tipo `String` e gli assegnerete una frase a piacere come valore iniziale. Con l'aiuto della documentazione Java della classe `String` provate a testare il risultato di alcuni metodi, ad esempio `length`, `substring`, `charAt`, `toUpperCase`, `toLowerCase`, `repeat`, etc.

Visualizzare a standard output i risultati. In particolare fate attenzione agli indici da utilizzare per metodi come `charAt` o `substring` (il primo carattere della stringa ha indice 0; i parametri di `substring` includono il primo indice ed escludono l'ultimo!).

Esercizio 10 - BombLab2

Questo è un esercizio di programmazione passiva, ovvero non dovete scrivere codice ma capirlo. Scaricare il file [BombLab2.java](#).

Salvatelo in una cartella e compilatelo con il comando `javac BombLab2.java`

Ora aprite il file sorgente `BombLab2.java` con il vostro editor. **Ignorate il main! E' troppo presto per capirlo se non avete mai programmato prima.**

All'interno della classe sono definiti tre metodi:

```
public static int first(int n);
```

```
public static int second(int n);  
public static int third(int n);
```

Il vostro compito e' capire quale risultato viene restituito da ciascun metodo quando come parametro esplicito n viene passato il vostro numero di matricola.

Assumendo che il vostro numero di matricola sia 123456 (**voi mettete il vostro**), mandate in esecuzione il programma scrivendo:
java BombLab 123456

A questo punto il programma vi chiederà di introdurre il codice per disattivare il primo stadio della bomba, che è il risultato che darà il **metodo first** se riceve come parametro n la vostra matricola. Inserite il risultato che pensate sia corretto, premete INVIO (o ENTER o RETURN a seconda della vostra tastiera) e incrociate le dita!

Se avete inserito il risultato corretto, il programma proseguirà chiedendo il codice per il secondo (metodo second) e poi per il terzo (metodo third) stadio.

Riuscirete a disattivare la bomba?

