2. Laboratorio 2

Esercizio 2.1

Se non l'avete ancora fatto, provate a rispondere alle domande del questionario di autovalutazione. Vi ricordo che il questionario non ha valenza per il superamento dell'esame. Non c'e' valutazione da parte della docente.

Esercizio 2.2

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 finestrella 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 WOOCLAP

https://app.wooclap.com/UJBOCD?from=event-page ;

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 piu' 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 finestrella 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'e' una variabile che si chiama **out** di tipo PrintStream! Ecco perche' 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 2.3

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

Scrivere una classe di collaudo (cioe' 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 . A queste costanti si accede con la notazione: NomeClasse.nomeCostante

Esercizio 2.4

Obiettivo: esercitarsi nella definizione e utilizzo di variabili e semplici operatori numerici. Stampa di risultati.

Scrivere un programma TestDivision che definisca due numeri interi m e n e invii a standard output il risultato e il resto della divisione intera m/n. Compilare e eseguire. Che cosa succede se il divisore vale zero? Scrivere poi un programma che definisca due numeri reali x e y e invii a standard

output il risultato della divisione x/y. Compilare ed eseguire. Esegue correttamente? Provare con le coppie di numeri:

x = 7.0 y = 0.0
x = -7.0 y = 0.0
x = 0.0 y = 0.0

Esercizio 2.5

Obiettivo: esercitarsi nella definizione e utilizzo di variabili e semplici operatori numerici. Stampa di risultati.

Scrivere un programma ProvaVariabili che:
- Definisca una variabile intera larghezza e la inizializzi a 20;
- Definisca una variabile intera lunghezza e la inizializzi a 30;
- Modifichi il valore di largezza attribuendogli il valore 40;
- Modifichi il valore di lunghezza aggiungendo 15 al suo attuale valore
- Stampi in output il messaggio "L'area del tavolo e' "
- Stampi in output il prodotto dei valori delle variabili larghezza e lunghezza

Esercizio 2.6

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 2.7

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 2.8

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' significativa: 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 proprieta' 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 piu' significativa) e 123 %100 = 23...

oppure

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

Esercizio 2.9

Obiettivo: per coloro a cui "Hello world!" esce dagli occhi, o per chi comunque ama le sfide

Scrivere un programma per scambiare il contenuto di due variabili, ma...
scambiare i valori di a e b **senza usare una variabile temporanea** per memorizzare uno dei due valori (NB: ci sono almeno due soluzioni; una di queste funziona perche' in Java le espressioni a destra del simbolo di assegnazione vengono valutate da sinistra verso destra. In c++ tale soluzione non e' detto che funzioni perché il c++ analizza le espressioni in un ordine tale da ottimizzare le prestazioni, quindi non necessariamente da sinistra verso destra).

NB: potete usare **solo** assegnazioni e le operazioni che abbiamo visto finora (**somma/sottrazione/moltiplicazione/divisione/modulo**, non potete utilizzare operazioni bit-a-bit, operatori booleani, o altro che non sia stato spiegato a lezione), quindi **tutti possono risolvere il quesito, ma non e' banale**.

Esercizio 2.10 - BombLab2

Questo e' un esercizio di programmazione passiva, ovvero non dovete scrivere codice ma capirlo. Scaricare il file BombLab2.java che trovate nella cartella "File utili e soluzione degli esercizi - Lab2").
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.
All'interno della classe sono definiti tre metodi:

```
public static int first(int n);
public static int secondi(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 chiedera' di introdurre il codice per disattivare il primo stadio della bomba, che e' il risultato che dara' 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 proseguira' chiedendo il codice per il secondo (metodo second) e poi per il terzo (metodo third) stadio.

Riuscirete a disattivare la bomba?