# 3. Laboratorio 3

## Esercizio 1

**Obiettivo: esercitarsi con operazioni**
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 e non fissato nel codice.

## Esercizio 2

**Obiettivo: esercitarsi con operazioni divisione e modulo fra interi**
Scrivere un programma che converta un numero decimale 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 e' 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 e' 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 e non fissati nel codice.

## Esercizio 3

Scrivere un programma di collaudo Bonifico.java per la classe BankAccount (di cui dovreste gia' disporre del bytecode, altrimenti lo trovate nella cartella "File utili e soluzioni degli esercizi" oppure potete produrlo compilando una classe BankAccount che potete realizzare voi stessi, come visto a lezione) 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

## Esercizio 4

**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 5

**Argomento: uso dell'input standard**
Scrivere un programma che saluti voi stessi. Chiedere all'utente di inserire il proprio nome. Leggere da standard input il nome inserito e stampare una stringa di saluto in output. Ad esempio, nel mio caso devo stampare "Hello, Cinzia".

## Esercizio 6

*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)
- 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 7

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

Modificare il programma dell'**Esercizio 6** 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 8

***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 9

**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.

## Esercizio 10

**Argomento: uso dell'input standard; manipolazione di stringhe e numeri**

Scrivere il programma PrintTimeInterval.java che legga dall'input standard due orari nel formato "24 ore", ciascuno di quattro cifre (ad esempio, 0900 oppure 1730), con il secondo orario successivo al primo visualizzi sull'output standard il numero di ore e di minuti (separatamente) che intercorrono fra i due orari secondo il seguente esempio

Inserire il primo orario: 0900
Inserire il secondo orario: 1730
Tempo trascorso: 8 ore e 30 minuti

## Esercizio 11

**Argomento: uso dell'input standard; manipolazione di stringhe e numeri**

Scrivere il programma PrintTimeInterval2.java, modificando il programma PrintTimeInterval visto in precedenza in modo che funzioni correttamente anche se il secondo orario è inferiore al primo (cioè per un intervallo di tempo che comprenda la mezzanotte), come in questo esempio di funzionamento:

Inserire il primo orario: 1730
Inserire il secondo orario: 0900
Tempo trascorso: 15 ore e 30 minuti

## Esercizio 12

**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 13

**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 14

**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

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

**SOLO DOPO AVER FATTO LE LEZIONI SUGLI IF**

## Esercizio 15

***Argomento: uso dell'input standard, confronto tra numeri***

Progettare il programma TwoNumbers che chieda all'utente di inserire due numeri 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 16

*Argomento: decisioni; confronto tra numeri reali*

Dopo averne osservato il codice, compilare la classe Numeric.java e Apprx.java. Eseguire quest'ultima e riflettere sui risultati.

Scrivere un programma che
- legga due numeri in virgola mobile
- visualizzi un messaggio che dice se i due numeri sono o meno approssimativamente uguali

## Esercizio 17

**Argomento: lettura da input, decisioni, confronto tra numeri**

Progettare il programma EvenNumber che:

• chieda all'utente di fornire un numero pari (ovviamente intero...)

• se il numero fornito è dispari, chieda di nuovo all'utente di fornire un numero pari, dopo aver visualizzato una segnalazione d'errore

• se il numero fornito (al primo o secondo tentativo) è pari, scriva un messaggio di congratulazioni all'utente, riportando anche il numero fornito (es. "Bravo! Il numero 26 è pari")

• se, invece, neanche il secondo numero fornito è pari, scriva un messaggio di disappunto Attenzione: se il primo numero fornito è pari, il programma NON deve chiedere un secondo numero.