

- 1.- Realitza una classe amb un mètode **factorialR** que, utilitzant la recursivitat, generi el factorial d'un número donat.
- 2.- Afegeix a la classe anterior un altre mètode **factorialI** que faci el mateix però de manera iterativa. Quin avantatge té el mètode iteratiu davant del recursiu?
- 3.- Afegeix a la classe anterior dos mètodes **fibonacciR** i **fibonacciI** que calculi la sèrie de Fibonacci de manera recursiva i iterativa, respectivament.
- 4.- Realitza un programa que, utilitzant recursivitat mostri per pantalla la següent piràmide:

```

      1
    1  1
  1  2  1
1  3  3  1
1  4  6  4  1

```

El programa podrà generar una piràmide de qualsevol número de files (inicialment prova a fer-ho sense donar la forma de piràmide, després ja mirareu de fer que tingui la forma que correspon)...

- 5.- Realitza un programa que mostri per pantalla el següent quadrat:

```

1    1    1    1    1
1    2    3    4    5
1    3    6   10   15
1    4   10   20   35
1    5   15   35   70

```

El programa podrà generar un quadrat de qualsevol dimensió. Utilitza la recursivitat per resoldre el problema

- 6.- (Dificultat alta) Resol els exercicis 4 i 5 de manera iterativa

7.- Crea un paquet Utilitats.mates amb dues classes Sumar i Potenciar. La classe Sumar tindrà un mètode `int suma(int,int)` el qual retornarà la suma dels dos paràmetres introduïts i la classe Potenciar tindrà un mètode `int potencia(int,int)` que retornarà el resultat d'elevat el primer paràmetre al segon paràmetre. Realitza un programa que faci ús d'aquest paquet.

- 8.- Es vol crear una classe que representi un **DNI espanyol** i que tingui les següents característiques:

- La classe emmagatzemarà el número de DNI en un **int**, sense guardar la lletra, ja que aquesta es pot calcular a partir del número. Aquest atribut serà privat a la classe.

- Per accedir al DNI es disposarà de dos mètodes **obtenir (get)**, un que proporcionarà el número de DNI (només les xifres numèriques) i un altre que tornarà el NIF complet (inclosa la lletra). Ambdós mètodes hauran de ser públics.
- Per modificar el DNI es disposarà de dos mètodes sobrecarregats **establir (set)** que permetran modificar el DNI. Un en què haurà de proporcionar el NIF complet (número i lletra) i un altre en què únicament serà necessari proporcionar el DNI (les set o vuit xifres). Si el DNI/NIF és incorrecte hauria de llançar una **excepció**. *Les excepcions les veurem la propera unitat formativa, però us mostraré com caldria fer-ho. Suposem que volem crear un mètode que calculi la divisió de dos nombres, aquest mètode hauria de llançar una excepció quan el denominador sigui zero (no es pot dividir per zero) així podríem crear el mètode de la següent manera:*

```
public double dividir(double numerador, double denominador) throws ArithmeticException{
    if(denominador==0 )
        throw new ArithmeticException("No es pot dividir per zero");
    return numerador/denominador;
}
```

- La classe disposarà d'alguns mètodes interns privats per calcular la lletra d'un número de DNI qualsevol, per comprovar si un DNI amb la seva lletra és vàlid, per extreure la lletra d'un NIF, etc. Aquests mètodes que no fan servir cap variable d'objecte es podrien declarar com a estàtics
 - private static char calcularLletraNIF (int dni)
 - private boolean validarNIF (String nif)
 - private static char extreureLletraNIF (String nif)
 - private static int extreureNumeroNIF (String nif)

Per tal de poder calcular correctament la lletra del DNI podeu mirar en aquesta pàgina com es forma [Càlcul lletra NIF](#)