



Exercici 14: Diagrama de seqüència (repàs)

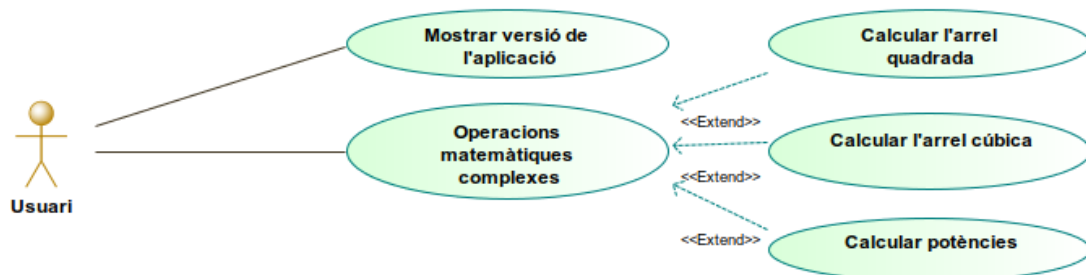
Descripció:

- **Objectiu:** Aprofundir i consolidar els coneixements adquirits sobre UML, en concret sobre el diagrama de seqüència.
- **Desenvolupament:** Individual.
- **Lliurament:** No cal lliurar cap document.

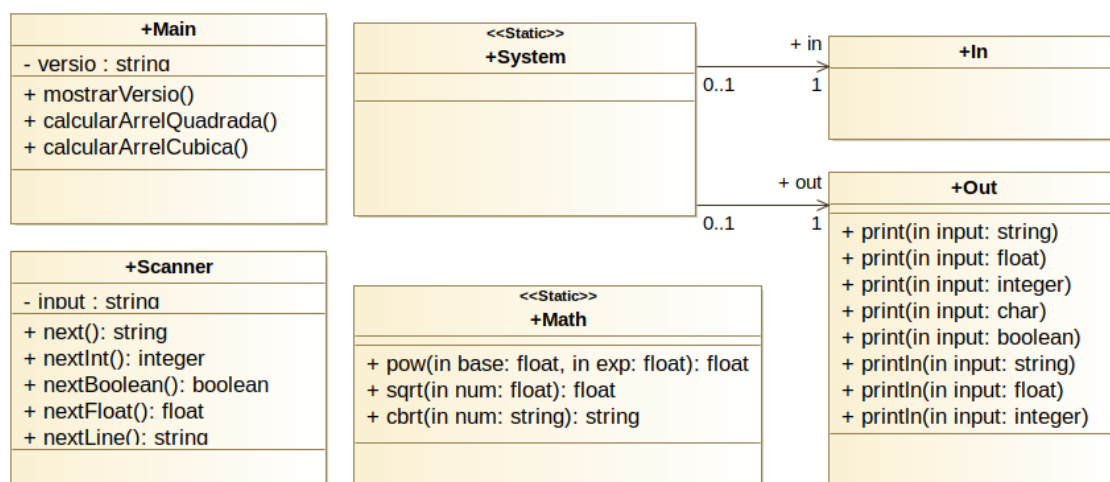
Enunciat

Estem desenvolupant un programa informàtic que permet fer una sèrie d'operacions, com les que es mostren al següent diagrama:

El següent diagrama de casos d'ús és un exemple de les eines que poden fer servir els usuaris:



El llenguatge de programació Java proporciona tota una sèrie de llibreries (framework) que permeten, entre d'altres, facilitar les tasques de lectura i escriptura de dades i també certes operacions com les potències i les arrels. El següent diagrama de classes mostra un exemple:





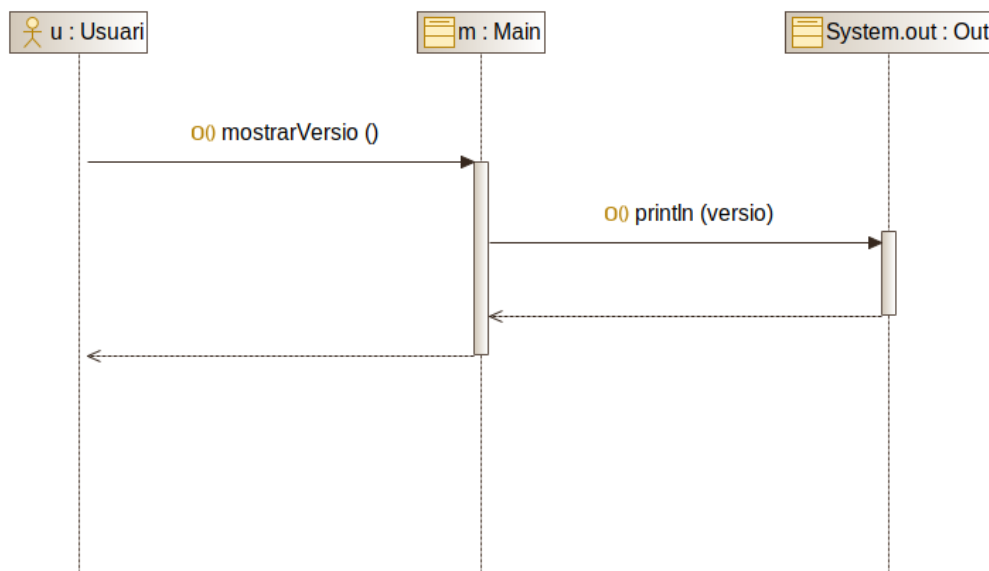
Exercici 1

Dibuixa el diagrama de seqüència del cas d'ús que permet mostrar per pantalla la versió de l'aplicació:

1. Pots assumir que ja disposes d'una instància de la classe "Main" a memòria.
2. La classe "System" és estàtica, i aquesta conté una classe "Out". Per a simplificar, pots assumir que disposes d'una instància de la classe "Out" a memòria anomenada "System.out".
3. L'actor és el que inicia el cas d'ús, per tant, serà l'actor el que invocarà el mètode "mostrarVersio" de la classe "Main".
4. El mètode "mostrarVersio" ha de fer el següent:
 1. Invocar el mètode "println" de la classe Out.
 2. Passar-li el valor "versió" de la classe "Main" com a paràmetre.

A continuació apareix una mostra del codi Java equivalent:

```
public class Main {  
    public void mostrarVersio() {  
        String versio = "1.0.9";  
        System.out.println(versio);  
    }  
}
```





Exercici 2

Dibuixa el diagrama de seqüència del cas d'ús que permet mostrar per pantalla el resultat de calcular una potència:

1. Pots assumir que ja disposes d'una instància de la classe "Main" a memòria.
2. La classe "System" és estàtica, i aquesta conté una classe "Out". Per a simplificar, pots assumir que disposes d'una instància de la classe "Out" a memòria anomenada "System.out".
3. La classe "System" és estàtica, i aquesta conté una classe "In". Per a simplificar, pots assumir que disposes d'una instància de la classe "In" a memòria anomenada "System.in".
4. La classe "Math" és estàtica i, per simplificar, pots assumir que disposes d'una instància a memòria.
5. L'actor és el que inicia el cas d'ús, per tant, serà l'actor el que invocarà el mètode "calcularPotencia" de la classe "Main".
6. El mètode "calcularPotencia" ha de fer el següent:
 1. Crea una nova instància de la classe "Scanner" fent servir el constructor. Atenció, el constructor demana un objecte de tipus "In", pots fer servir "System.in"
 2. Invocar el mètode "nextFloat" de la classe "Scanner" per a llegir quina serà la base de la potència.
 3. Invocar el mètode "nextFloat" de la classe "Scanner" per a llegir quin serà l'exponent de la potència.
 4. Invocar el mètode "pow" de la classe "Math".
 5. Cal passar-li els dos paràmetres d'entrada (base i exponent).
 6. Invocar el mètode "println" de la classe Out.
 7. Passar-li el resultat de l'operació com a paràmetre.

```
public void calcularPotencia() {  
    Scanner scanner = new Scanner(System.in);  
  
    float base = scanner.nextFloat();  
    float exp = scanner.nextFloat();  
  
    double result = Math.pow(base, exp);  
    System.out.println(result); //double o float, és indiferent  
}
```



Institut Puig Castellar
CFGs: DAM
M05: Entorns de desenvolupament
UF3: Introducció al disseny orientat a objectes
Exercici 14: Diagrama de seqüència (repàs)

