

# MOOC Init Prog Java

## Exercices facultatifs semaine 3

### Une histoire de prêt (niveau 2)

L'objectif de cet exercice est de résoudre le problème suivant :

Une banque fait un prêt à une personne X pour un montant total de **S0** euros. Cette personne rembourse chaque mois un montant fixe **r** et paye (en plus) un intérêt variable  $i = ir * S$ , où **ir** est le taux d'intérêt mensuel (fixe) et **S** la somme restant à rembourser (avant déduction du remboursement mensuel).

Le but est de déterminer la somme des intérêts encaissés par la banque une fois le prêt remboursé.

Écrivez pour cela le programme `Pret.java` qui calcule la somme des intérêts encaissés et la durée en mois du remboursement, puis qui affiche ces informations à l'écran.

Le programme doit en outre demander à l'utilisateur les valeurs **S0** (strictement positif), **r** (strictement positif) et **ir** (compris strictement entre 0 et 1) et s'assurer de leur validité.

Testez votre programme avec les valeurs suivantes :  $S_0=30000$ ,  $r=1200$ ,  $ir=0.01$  (i.e. 1%). La somme des intérêts encaissés (sur 25 mois) est alors de 3900 euros.

---

## Suite et série (niveau 2)

a) Écrivez un programme qui calcule les 10 premiers termes de la suite  $U_n$  telle que:

$$U_0 = 1, \quad U_{n+1} = \frac{U_n}{n+1}$$

Vous devez trouver:

$$U_0 = 1$$

$$U_1 = 1$$

$$U_2 = 0.5$$

$$U_3 = 0.16666666666666666$$

$$U_4 = 0.041666666666666664$$

$$U_5 = 0.008333333333333333$$

$$U_6 = 0.0013888888888888889$$

$$U_7 = 1.984126984126984E-4$$

$$U_8 = 2.48015873015873E-5$$

$$U_9 = 2.7557319223985893E-6$$

$$U_{10} = 2.7557319223985894E-7$$

b) Modifiez votre programme pour qu'il calcule simultanément la suite  $U_n$  et la série  $V_n$ , où

$$V_n = \sum_{i=0}^n U_i.$$

Vérifiez que  $V_n$  converge vers  $e = \exp(1) = 2.71828...$

---

## Figures en mode texte (niveau 1)

a) Ecrivez un programme qui affiche les valeurs 1 à 9 en ligne, à l'aide d'une boucle `for`:

```
123456789
```

b) Modifiez le programme pour qu'il affiche 9 lignes similaires, à l'aide de 2 boucles `for`:

```
123456789
123456789
.
.
.
123456789
```

c) Comment modifier le programme pour qu'il affiche un triangle ?

```
1
12
123
1234
12345
123456
1234567
12345678
123456789
```

d) Modifiez une dernière fois votre programme, pour qu'il affiche une pyramide inversée:

```
          1
         12
        123
       1234
      12345
     123456
    1234567
   12345678
  123456789
```

---

## Triangle (niveau 2)

Utilisez des boucles afin de construire un triangle isocèle formé par le caractère étoile (\*).  
Affichez-en  $n$  lignes, où  $n$  sera entré au clavier par l'utilisateur.

Exemple: pour  $n = 5$  :

```
  *
 * * *
* * * * *
* * * * * *
* * * * * * *
```

---