
Examen NFP135 : 2 heures

Les documents imprimés sur papier sont autorisés, smart-phone/tablettes/ordinateurs interdits.

Chaque exercice vaut 3.5 points pour un total de $6 \times 3.5 = 21$ points y compris un point bonus.

Exercice 1 Écrire un programme `processing` pour tracer :

- une ligne verte qui relie les points (20, 50) et (70, 70) ;
- un rectangle rempli de couleur bleu ;
- un gros cercle non-rempli qui couvre tous les dessins précédents.

Exercice 2 Écrire une fonction `vitesseMoyenne(...)` sous `Java` or `Processing` (votre choix, car c'est presque le même code). Cette fonction prend deux arguments de type `double` : une distance (en km) et une durée (en heures). Cette fonction doit renvoyer un `double` qui représente la vitesse moyenne, c.à.d., la distance divisée par la durée. On dit que l'en-tête (ou la signature) de la fonction est :

```
double vitesseMoyenne(double distance, double durée)
```

Le corps de cette fonction (votre réponse) devrait avoir une seule ligne pour faire le calcul $\frac{\text{distance}}{\text{durée}}$ et renvoyer le résultat avec `return`.

Exercice 3 Écrire une fonction `boolean noteValide(double x)` sous `java` or `processing` (votre choix, car c'est presque le même code). Observez que cette fonction prend comme entrée un seul argument de type `double` noté `x`. Elle doit renvoyer un `boolean` qui indique si l'argument représente une note valide ou pas. On dit que l'en-tête (ou la signature) de la fonction est :

```
boolean noteValide(double x)
```

Note : Une note est valide si elle est supérieure ou égale à zéro et inférieure ou égale à 20.

Exercice 4 Écrire une fonction Java ou Processing `int puissance(...)` qui prend deux arguments `x` et `n` de type `int` et qui renvoie la valeur x^n . Le résultat sera un `int`. L'en-tête (ou la signature) complet est :

```
int puissance(int x, int n)
```

Attention : si `n` est inférieur à zéro, il faut renvoyer -1 et afficher « impossible de calculer cette puissance avec un exposant négatif ».

Exercice 5 Écrire une fonction d'en-tête

```
int nbNotesValides(int[] tab)
```

qui renvoie le nombre de notes valides dans un tableau `tab`. Vous pouvez appeler la fonction `noteValide(...)` écrite à l'exercice 3.

Exercice 6 Écrire un programme `java` ou `processing` qui calcule le nombre de valeurs positives (ou nulles) et le nombre de valeurs négatives dans un tableau d'entiers `t` déclaré et initialisé au début du programme principal. Si vous utilisez `Java`, le programme principal c'est la fonction `main`. Si vous utilisez `Processing`, le programme principal c'est la fonction `setup`. Le résultat doit être stocké dans deux variables globales (donc statiques si vous utilisez Java) `nbPosit` et `nbNegat`.