

TP 3 : Types primitifs, transtypage et API Java

I. Les types primitifs en Java et le transtypage

Considérez le code suivant :

```
char a = 'A';
char b = 'a' + 2;
int c = a;

System.out.println(a + " " + b + " " + c);
```

1. Sans tester le code, à votre avis, est-il correct ?
2. Faites un copier/coller de ce code dans un programme *Java*. Compilez-le et exécutez-le.
3. Comment expliquez-vous le résultat obtenu ?

Considérez maintenant le code suivant :

```
byte entier1 = 1;
int entier2 = 255;
long entier3 = 2;

entier1 = entier2 + entier1;
entier3 = entier2 + entier1;
System.out.println(entier1 + " " + entier2 + " " + entier3);
```

4. Toujours sans tester le code, à votre avis, est-il correct ?
5. Ce code ne compile pas. Modifiez-le pour le compiler sans modifier le type des variables.
6. Cette instruction est-elle correcte : `entier1 = (byte)entier2 + entier1` ? Testez-la et observez le résultat.

Considérez maintenant le code suivant :

```
float reel1 = 1.5;
double reel2 = 2.8;

System.out.println(reel1 + " " + reel2);
```

7. Est-ce que ce code est correct ?
8. Corrigez ce code sans modifier le type des variables et sans utiliser le transtypage.

Considérez maintenant le code suivant :

```
int entier4 = 5;
double reel3 = entier4;
int entier5 = reel3;

System.out.println(entier4 + " " + reel3 + " " + entier5);
```

9. Ce code est-il correct ?
10. Corrigez ce code afin qu'il donne un résultat correct.

Considérez le code suivant :

```
boolean toto = false;
int entier6 = toto;

System.out.println(toto + " " + entier6);
```

11. Ce code est-il correct ?
12. Le transtypage peut-il corriger le problème ?

Le but de cet exercice était de vous sensibiliser aux différents types de variable rencontrés en *Java*, au problème des valeurs constantes (notamment réelles), ainsi qu'au transtypage.

II. La classe Math

Il est possible d'accéder à la documentation *Java* en ligne à l'adresse suivante :

<https://docs.oracle.com/en/java/javase/13/docs/api/index.html>

ou en cherchant **API Java 13** (ou **API Java 12**) sur votre moteur de recherche préféré.

1. Sélectionnez le module **java.base** situé dans la partie gauche de la page, puis le package **java.lang**, puis la classe **Math** dans la section **Class Summary**. Regardez les différentes méthodes et constantes proposées.
2. Quelle méthode permet de calculer la puissance (x^y) ?
3. Quelle méthode permet de calculer le cosinus ?
4. Quelle méthode permet de générer un nombre aléatoire ?
5. Quelles constantes sont proposées dans la classe **Math** ?

III. Génération de nombres aléatoires

1. Écrivez un programme qui demande un nombre n à l'utilisateur et qui affiche n réels aléatoires dans l'intervalle $[0; 1[$ (en utilisant une méthode de la classe **Math**) à l'écran.
2. Modifiez votre programme pour qu'il affiche les valeurs minimale et maximale obtenues.
3. Modifiez votre programme de sorte à ce que les nombres aléatoires soient plutôt dans l'intervalle $[0; 10[$.
4. Modifiez votre programme de sorte à ce que les nombres aléatoires soient des entiers dans l'intervalle $[-5; 5]$.

IV. Le cercle

Écrivez un programme qui demande un rayon r à l'utilisateur et qui calcule le périmètre du cercle correspondant à ce rayon. Si le rayon saisi est négatif, un message d'erreur doit être affiché et une nouvelle valeur doit être demandée.

V. Le nombre à deviner

Le jeu du nombre à deviner consiste à générer un nombre aléatoire à le faire deviner à l'utilisateur. Pour cela, à chaque proposition de l'utilisateur, le programme indique si le nombre proposé est supérieur ou inférieur à la solution. Le jeu se termine lorsque l'utilisateur trouve la solution.

1. Écrivez le programme de ce jeu en supposant que l'ordinateur choisit un nombre aléatoire entre 1 et 1024.

On désire maintenant modifier le programme de sorte à obliger l'utilisateur à jouer intelligemment. Ainsi, si l'utilisateur propose un nombre inférieur ou égal à une précédente proposition qui avait déjà été annoncée comme étant trop petite, l'ordinateur affichera "Trop petit, je te l'ai déjà dit". De la même façon, si l'utilisateur propose un nombre supérieur ou égal à une précédente proposition qui avait déjà été annoncée comme étant trop grande, l'ordinateur affichera "Trop grand, je te l'ai déjà dit".

2. Modifiez votre programme de sorte à tenir compte de ces nouvelles règles.

En connaissant les bornes de l'intervalle initial, il est possible de limiter le nombre maximal de propositions par l'utilisateur (indice : il faut calculer un logarithme de base 2 pour trouver cette limite).

3. Modifiez votre programme en conséquence. Vous afficherez le nombre restant d'essais avant chaque saisie de l'utilisateur.
4. Modifiez votre programme de sorte à laisser à l'utilisateur le choix des bornes initiales de l'intervalle.