

Objectifs: Comprendre et mettre en oeuvre les notions de **variable**, **constante**, **données**, **types**, **instruction**, **programme**, **appel de fonction**.

Rappel : En iJava, `void algorithm()` est la fonction appelée lors de l'exécution d'un programme.

Exercice 1 : Parlons programmation

Décrivez de manière claire, ligne après ligne, le contenu du programme suivant.

```
1 class UnPremierProgramme extends Program {
2     void algorithm() {
3         String msg = "Salut_l'univers_!";
4         println(msg);
5         println("La_chaîne_ci-dessus_a_une_longueur_de_ " + length(msg) + "_caractères");
6     }
7 }
```

Exercice 2 : Le programme dont vous êtes le compilateur

Voici un programme :

```
1 class UnProgramme extends Program{
2     void algorithm() {
3         msg = "Ceci_est_un_message_plutôt_long_pour_ne_pas_dire_grand_chose
4         println(substring(msg, 7) + "... "
5     }
6 }
```

1. Ce programme compile-t-il ? Si non, corriger les erreurs.
2. Supposons maintenant les éventuelles erreurs corrigées. On remplace le contenu de la variable `msg` par la chaîne "Bref.". Le programme compile-t-il ? Si oui, indiquez ci-dessous ce que vous pensez qu'il va se produire à l'exécution.

Exercice 3 : Écriture d'un programme simple

Écrire un programme qui stocke le nombre entier 21 dans une variable, double la valeur de la variable, et enfin, affiche sa nouvelle valeur.

Exercice 4 : Drôles de types

Un programmeur un peu étourdi a oublié les types et déclarations de variables du programme suivant :

```
1 class Ttc extends Program {
2     void algorithm() {
3         ... prixHt = 99.99 ;
4         ... taxe = 0.196 ;
5         ... libelle = "tablette";
6         //Calcul du prix ttc
```

```

7 | ... prixTtc = prixHt * (1 + taxe);
8 | ... message = libelle + "_coûte_" + prixTtc + "_euros.";
9 | // Affichage
10 | print(message);
11 | }
12 | }

```

1. Complétez les déclarations manquantes concernant les types de variables
2. Que produit l'exécution de ce programme ?

3. Que faudrait-il changer dans `algorithm` si on souhaite désormais appliquer une tva réduite de 5.5% à un sac de pommes de terre à 6 euros hors taxes ?

Exercice 5 : Le programme dont vous faites la trace d'exécution

C'est la nouvelle année. Kazu a 3 projets : en priorité s'acheter un ordinateur, s'acheter une paire de rollers et aller au cinéma.

Ci-dessous, voici à gauche les étapes de Kazu dans sa quête et à droite le programme correspondant.

Kazu a prévu une enveloppe pour chacun de ces projets. Malheureusement, il n'a que 120€ pour l'instant, qu'il a mis intégralement dans l'enveloppe pour son ordinateur et c'est loin d'être suffisant ! Il décide donc d'aller souhaiter la bonne année à ses oncles :)

- Son oncle Shao lui donne 75 euros, ça sera de plus pour l'ordinateur.
- Chez son oncle Tetsuo, il reçoit un bon de 30 euros valable dans le magasin de sport uniquement, ils serviront pour les rollers.
- Kazu accepte sa proposition de sa soeur Yoko de mettre autant que lui pour l'ordinateur à condition qu'elle puisse s'en servir aussi.
- Chez son oncle Takeda, c'est 20 euros qu'il répartit en 2 parts égales entre le cinéma et les rollers.
- Kazu annonce à ses parents avec fierté les sommes dont il dispose pour chacun de ses projets.

```

1 | class Kazu extends Program{
2 |
3 |     void algorithm() {
4 |         //Déclaration des constantes
5 |         final int SHAO = 75;
6 |         final int TETSUO = 30;
7 |         final int TAKEDA = 20;
8 |         //Déclaration + init. des
           variables
9 |         int ordi = 120;
10 |        int rollers = 0;
11 |        int cinema = 0;
12 |        //Changements au fil de la
           quête de Kazu
13 |        ordi = ordi + SHAO;
14 |        rollers = rollers + TETSUO;
15 |        ordi = ordi * 2;
16 |        cinema = cinema + TAKEDA/2;
17 |        rollers = rollers + TAKEDA/2;
18 |        println("ordinateur:_:" + ordi);
19 |        println("rollers:_:" + rollers);
20 |        println("cinema:_:" + cinema);
21 |    }
22 | }

```

1. Faites le lien entre le texte et le programme.
2. Réalisez la trace d'exécution de cet algorithme.

Donnée(s)	ordi	cinema	rollers	Sortie(s)

Exercice 6 : Évaluation d'expressions diverses

Dans cet exercice, il s'agit d'évaluer diverses expressions en précisant leur type et leur valeur. Remplissez le tableau donné ci-dessus en donnant le résultat de l'évaluation des expressions.

	Expression iJava	Type	Evaluation
a	3 * 2		
b	2 * 1.5		
c	'A' + 1		
d	length("Programmation")		
e	"conca" + "té" + "nation"		
f	12/7+5		
g	9%3		
h	charAt("chaîne",1)		

Exercice 7 : Permuter les valeurs de deux variables

Le but de cet exercice est d'écrire un algorithme qui échange le contenu de deux variables de type chaîne chaîne1 et chaîne2. Ainsi, si l'on dispose de deux variables chaîne1 et chaîne2, à la fin de l'algorithme, le contenu de la variable chaîne1 doit être celui contenu initialement par chaîne2 et réciproquement.

```

1 class Permutation extends Program {
2     void algorithm() {
3         String chaîne1 = "premier";
4         String chaîne2 = "deuxième";
5         // A COMPLÉTER par le(s) instruction(s) permutant les valeurs des deux variables
6
7
8
9
10        println("Après_échange,_chaîne1=_ " + chaîne1 + "_et_chaîne2=_ " + chaîne2);
11    }
12 }
```

Exercice 8 : Manipulations de fonction sur les chaînes de caractères

Dans cet exercice, nous utiliserons la fonction substring(String s, int start, int end) → String qui permet de copier une sous-chaîne grâce à un indice de début (start) et de fin d'extraction (end). ATTENTION: le caractère à l'indice end n'est pas copié!

Remplissez le tableau donné ci-dessus en donnant le résultat de l'évaluation des expressions.

	Expression iJava	Évaluation
a	substring("Automne", 6, 7)	
b	substring("INFORMATIQUE", 1, 5)	
c	substring("hiver", 1, 1)	
d	substring("hiver", 3, 6)	
e	substring("Unité de valeur", 0, 1) + substring("Unité_de_Valeur", 9, 10)	
f	substring("Informatique", 2, 3, 4)	
g	substring("Prénom", (length("PRENOMS") + 2)/4, 3)	
h	substring("substring(\"OUI ou NON\", 8, 3)\", 11, 14)	

Prolongements (pour s'entraîner)

Exercice 9 : Corriger les erreurs d'un programme

Trouver et corriger les 7 erreurs dans le programme qui suit :

```

1 class Programme7Erreurs {
2     algorithm() {
3         int nombre = "202"
4         int double;
5         double = nombre * 2;
6         Println(double);
7     }
```

Exercice 10 : Comprendre un programme

Parfois les programmes sont malheureusement peu clairs, mais il faut tout de même parvenir à les comprendre. Que fait le programme ci-dessous ?

```

1 class ProgrammeMystere extends Program {
2     void algorithm() {
3         String s1 = "12345";
4         String s2 = "9876";
5         println(charAt(s2,0)-'0'+charAt(s1,length(s1)-1)-'0');
6     }
7 }
```

Exercice 11 : Pâques ou pas cap ?

L'algorithme donné ci-dessous permet de déterminer la date de Pâques d'une année donnée.

```

1 void algorithm() {
2     int a, b, c, d, e, f, g, h, i, k, l, m, result, year;
3     year = ... ;
4     a = year % 19;
5     b = year / 100;
6     c = year % 100;
7     d = b / 4;
8     e = b % 4;
9     f = (b + 8) / 25;
10    g = (b - f + 1) / 3;
11    h = (19 x a + b - d - g + 15) % 30;
12    i = c / 4;
13    k = c % 4;
14    l = (32 + 2 x e + 2 x i - h - k) % 7;
15    m = (a + 11 x h + 22 x l) / 451;
16    result = 22 + h + l - 7 x m;
17    println(result);
18 }
```

Faites la trace d'exécution pour 2008 puis pour l'année 2013.