BASES DE LA PROGRAMMATION

STRUCTURES DÉCISIONNELLES

PRENDRE DES DÉCISIONS

- Il est fréquent, dans un programme, de devoir prendre une décision en fonction du contexte
- Plusieurs *structures décisionnelles* permettent d'exécuter différentes actions
  - l'instruction `if`
  - l'instruction `if/else`
  - l'instruction `if/else if`
  - l'instruction `switch`

INSTRUCTION IF (SI)

- Si une certaine condition spécifiée est *vraie*, alors on exécute l'action A et ensuite on continue le programme
  - si la condition est fausse, l'action A est ignorée, on passe directement à la suite
- La condition doit pouvoir s'évaluer à un **booléen** (type `boolean` en Java) :
  - vrai (`true`) ou faux (`false`)

```
SI (condition est vraie)
    Action A
// suite du programme
```

```
SI (le feu est rouge)
    S'arrêter
// suite du programme
```

PROBLÈME

LES COMMERCIAUX REÇOIVENT UN SALAIRE DE BASE DE 400 € PAR SEMAINE. CEUX QUI FONT AU MOINS 10 VENTES SUR LA SEMAINE ONT UNE PRIME DE 250 €.

ANALYSE

- Chaque commercial va recevoir les 400 € quoi qu'il arrive
- **SI** le commercial fait au moins 10 ventes, il reçoit 250 € en plus
  - cette condition implique un *branchement*, un détour dans le programme : on va ajouter les 250 € si la condition « au moins 10 ventes » est vraie
  - c'est une instruction `if`

ALGORITHME

- Initialiser le salaire à 400, la prime à 250 et le quota de ventes pour la prime à 10
- Récupérer le nombre de ventes du commercial sur la semaine
- **SI** le nombre de vente est au moins égal au quota
  - ajouter la prime au salaire
- Afficher le salaire du commercial

INSTUCTION IF EN JAVA

- La syntaxe en Java est donnée ci-dessous
  - la condition doit être entre parenthèses
  - elle doit être une expression booléenne (V/F)
  - les accolades { } permettent de délimiter le bloc de code à exécuter quand la condition est vraie

```
if (condition) {
  // ce bloc n'est exécuté que si condition est V
}
// suite du programme
```

IMPLÉMENTATION DE L'ALGORITHME EN JAVA

EXEMPLE D'IMPLÉMENTATION

```java
public static void main(String[] args) {
  // Initialisation des variables
  int salaire = 400;
  int prime = 250;
  int quota = 10;

  // Récupérer le nombre de ventes
  System.out.print("Combien de ventes avez-vous conclues cette semaine ? ");
  Scanner clavier = new Scanner(System.in);
  int nbVentes = clavier.nextInt();
  clavier.close();

  // Test : prime ou pas ?
  if (nbVentes >= quota) {
    salaire = salaire + prime;
  }

  // Affichage
  System.out.println("Votre salaire est de : " + salaire + " €");
}
```

OPÉRATEURS RELATIONNELS

- Pour écrire une condition, il est fréquent d'utiliser des **opérateurs relationnels**, qui permettent de comparer des valeurs :
  - `<`, `>`, `<=`, `>=`
  - égalité : `==` (et non pas un seul `=`)
  - différence : `!=`
- Ex :
  - `if (salaire >= 3000) { … }`
  - `if (temperature == 0) { … }`
  - `if (note != noteMaximale) { … }`

INSTRUCTION IF/ELSE (SI/SINON)

- On veut spécifier ce qui se passe **dans les 2 cas** (V ou F)
- Si la condition est *vraie*, alors on exécute l'action A ; sinon on exécute l'action B ; ensuite on continue le programme dans tous les cas
  - **UNE** branche du IF/ELSE exactement est exécutée

```
SI (condition)
    Action A (branche VRAI)
SINON
    Action B (branche FAUX)
// suite du programme, arrivé à ce point on est certain que soit A soit B a été exécutée
```

```
SI (argent liquide disponible)
    Payer en liquide
SINON
    Payer par CB
// suite du programme
```

PROBLÈME

LES COMMERCIAUX REÇOIVENT UN MESSAGE DE FÉLICITATIONS S'ILS ONT REMPLI LE QUOTA DE 10 VENTES MINIMUM SUR LA SEMAINE ; SINON, ILS SONT INFORMÉS DU NOMBRE DE VENTES QU'IL MANQUAIT

ANALYSE

- Un message sera toujours affiché au commercial, mais le contenu du message est conditionné par le fait qu'il a réussi à faire 10 ventes ou pas
- On a donc une décision à prendre : on affiche les félicitations **OU BIEN** le nombre de ventes manquantes
  - cela pourra être implémenté avec un IF/ELSE

ALGORITHME

- Initialiser le quota de ventes à 10
- Récupérer le nombre de ventes du commercial sur la semaine
- **SI** le nombre de vente est supérieur ou égal au quota
  - afficher "Félicitations ! Vous avez rempli le quota."
- **SINON**
  - calculer le nombre NB de ventes manquantes
  - et afficher "il vous manque NB ventes pour atteindre le quota"

INSTUCTION IF/ELSE EN JAVA

- La syntaxe reprend celle du IF simple, on ajoute juste la branche FAUX en utilisant le mot-clé ELSE
  - attention de ne pas oublier les parenthèses pour la condition
  - et les 2 paires d'accolades pour délimiter les 2 blocs

```
if (condition) {  // la condition doit s'évaluée à un booléen (V/F)
  // ce bloc est exécuté si condition est true
} else {
  // ce bloc est exécuté si condition est false
}
// avec un IF/ELSE, L'UNE des 2 branches est TOUJOURS exécutée
// suite du programme
```

IMPLÉMENTATION DE L'ALGORITHME EN JAVA

EXEMPLE D'IMPLÉMENTATION

```java
public static void main(String[] args) {
  int quota = 10;

  // Récupérer le nombre de ventes
  System.out.print("Combien de ventes avez-vous conclues cette semaine ? ");
  Scanner clavier = new Scanner(System.in);
  int nbVentes = clavier.nextInt();
  clavier.close();

  // Le IF/ELSE est nécessaire parce qu'on n'affiche
  // pas le même message en fonction du nombre de ventes
  if (nbVentes >= quota) {
    System.out.println("Félicitations ! Vous avez rempli votre quota.");
  } else {
    int nbManquantes = quota - nbVentes;
    System.out.println("Il vous manque " + nbManquantes + " ventes.");
  }
}
```

EXEMPLE D'IMPLÉMENTATION V2

```java
// Même chose mais on a ici conservé, en même temps,
// le traitement précédent sur le salaire
public static void main(String[] args) {
  int salaire = 400;
  int prime = 250;
  int quota = 10;

  Scanner clavier = new Scanner(System.in);
  System.out.print("Entrez votre nombre de ventes : ");
  int nbVentes = clavier.nextInt();
  clavier.close();

  if (nbVentes >= quota) {
    System.out.println("Félicitations ! Vous avez rempli votre quota.");
    salaire = salaire + prime;
  } else {
    int nbVentesManquantes = quota - nbVentes;
    System.out.println("Il vous manque " + nbVentesManquantes + " ventes");
  }

  System.out.println("Votre salaire est de : " + salaire);
}
```

INSTRUCTION IF/ELSE IF

- On peut utiliser une instruction IF/ELSE IF si **plus de deux alternatives** sont identifiées
- Si une certaine condition cond1 est *vraie*, alors on exécute l'action A ; sinon, si une deuxième condition cond2 est *vraie*, on exécute l'action B ; si toujours non, on exécute l'action C ; ensuite on continue le programme
  - dans tous les cas, UNE branche parmi les trois exactement est exécutée
  - on peut ajouter autant de ELSE IF qu'on veut, si 3 alternatives ne sont pas suffisantes

INSTRUCTION IF/ELSE IF

```
SI (cond1)
    Action A
SINON SI (cond2)
    Action B
SINON // Ce sinon "ramasse" tous les cas non prévus par les 2 conditions
    Action C
// suite du programme
```

```
SI (argent liquide disponible)
    Payer en liquide
SINON SI (CB disponible)
    Payer par CB
SINON
    Payer par chèque
// suite du programme
```

PROBLÈME

AFFICHER LA COULEUR CORRESPONDANTE À UN ÉLÈVE EN FONCTION DE LA NOTE D'UNE ÉVALUATION

- "rouge" entre 0 et 5
- "jaune" entre 6 et 10
- "vert" entre 11 et 15
- "vert+" entre 16 et 20

ANALYSE

- Un IF/ELSE simple ne suffit pas : on a 4 cas différents
- On va utiliser 3 conditions grâce à des IF/ELSE IF
- La 4ème alternative sera "ramassée" par le ELSE final

ALGORITHME

- Récupérer la note de l’évaluation
- **SI** la note est inférieure à 6
  - afficher `rouge`
- **SINON SI** la note est inférieure à 11
  - afficher `jaune`
- **SINON SI** la note est inférieure à 16
  - afficher `vert`
- **SINON** afficher `vert+`

INSTUCTION IF/ELSE IF EN JAVA

- La syntaxe est juste une évolution du IF/ELSE
  - de nouveau, il faut des accolades pour délimiter tous les blocs

```
if (cond1) {
  // ce bloc est exécuté si cond1 est true
} else if (cond2) {
  // ce bloc est exécuté si cond2 est true
} else {
  // ce bloc est exécuté dans tous les autres cas
  // on peut continuer avec autant de else if qu'on a besoin
}
// suite du programme
```

IMPLÉMENTATION DE L'ALGORITHME EN JAVA

EXEMPLE D'IMPLÉMENTATION

```java
public static void main(String[] args) {
  // Récupérer la note
  System.out.print("Entrez votre note sur 20 : ");
  Scanner clavier = new Scanner(System.in);
  int note = clavier.nextInt();
  clavier.close();

  // On déclare une variable intermédiaire qui va retenir la couleur
  String couleur;
  // Test : dans quelle fenêtre sonnes-nous ?
  if (note < 6) {          // 0-5
    couleur = "rouge";
  } else if (note < 11) { // 6-10
    couleur = "jaune";
  } else if (note < 16) { // 11-15
    couleur = "vert";
  } else {                 // 16-20
    couleur = "vert+";
  }

  // La variable "couleur" est maintenant correctement initialisée
  System.out.println("Votre couleur : " + couleur);
}
```

OPÉRATEURS LOGIQUES

- Parfois on doit combiner plusieurs expressions pour tester une condition
  - `ex:SI (feuEstVert OU pompiersEnIntervention)`
  - `OU` est ici un *opérateur logique* qui permet de combiner les deux expressions booléennes pour n'en faire qu'une
  - il y a deux autres opérateurs logiques : `ET` et `NON`

PROBLÈME

POUR OBTENIR UN PRÊT, ON DOIT GAGNER PLUS DE 30000 € PAR AN ET TRAVAILLER DEPUIS AU MOINS DEUX ANS

ALGORITHME

- Récupérer le salaire annuel
- Récupérer le nombre d’années d’emploi
- **SI** salaire est supérieur à 30000 **ET** il travaille depuis au moins deux ans
  - afficher que le prêt est accepté
- **SINON**
  - afficher que le prêt est refusé

LES OPÉRATEURS LOGIQUES JAVA

- **&&** (ET) : les deux conditions doivent être vraies
  - `(1 <= 2 && 4 != 5)` est VRAI
- **||** (OU) : l'une des deux au moins doit être vraie
  - `(3 == 4 || 6 > 1)` est VRAI
- **!** (NON) : on prend l'inverse du résultat
  - `!(2 == 3)` est VRAI

OPÉRATEURS LOGIQUES - EXEMPLES

```
if (feuEstOrange && vitesse >= 40 && distanceAuFeu < 15) {
  accelerer();
}

if (!possedeLaCle || coffreEstCoince) {
  EquiperHache();
  ForcerLeCoffre();
}
```

IMPLÉMENTATION DE L'ALGORITHME EN JAVA

EXEMPLE D'IMPLÉMENTATION

```java
public static void main(String[] args) {
  // Récupérer le salaire annuel
  Scanner clavier = new Scanner(System.in);
  System.out.print("Entrez votre salaire annuel : ");
  int salaire = clavier.nextInt();

  // Récupérer le nombre d'années d'emploi
  System.out.print("Entrez le nombre d'années d'emploi : ");
  int anneesEmploi = clavier.nextInt();
  clavier.close();

  // La condition du IF contient deux choses à évaluer
  // Les deux doivent être vraies pour que la condition soit vraie
  // Il faut donc un ET (&& en Java)
  if (salaire > 30000 && anneesEmploi >= 2) {
    System.out.println("Le prêt est accepté.");
  } else {
    System.out.println("Le prêt est refusé.");
  }
}
```

EXERCICE - CAFÉ-CRÈME

- Écrire un programme qui annonce si on n'a pas assez, juste assez, ou plus qu'il ne faut pour se payer un cagé-crème à 1 € :
  - demande combien on a de « pièces jaunes » de chaque type (1, 5, 10, 20, 50)
  - affichage en fonction du total calculé (3 types de messages possibles) :
    - `Il vous manque 25 cents pour prendre un café.`
    - `Vous avez exactement de quoi vous payer le café !`
    - `Il vous restera 15 cents après avoir pris votre café !`
- Un exemple de sortie est donné sur la diapo suivante

CAFÉ-CRÈME - EXEMPLE D'EXÉCUTION

```
Bonjour ! Je suis l'assistant de la machine à café. Dites-moi les pièces jaunes dont vous dispose

pièces de 1 cent ? 5
pièces de 5 cents ? 5
pièces de 10 cents ? 2
pièces de 20 cents ? 0
pièces de 50 cents ? 1

Vous avez exactement de quoi vous payer le café-crème !
```