

# ANA2 - Lucky Summary

Sm!le42

9 mai 2021

## Table des matières

<b>1 Diagramme d'activité</b>	<b>1</b>
1.1 Transitions automatiques	1
1.2 Début et Fin	1
1.3 Conditions	2
1.4 Itérations (boucles)	2
<b>2 Classes et objets</b>	<b>2</b>
2.1 Classes	2
2.1.1 Visibilités	3
2.1.2 Attributs	3
2.1.3 Opérations	3
2.2 Objets	4
<b>3 Associations</b>	<b>4</b>
3.1 Associations 1-1 et 1-N	4

## 1 Diagramme d'activité

### 1.1 Transitions automatiques

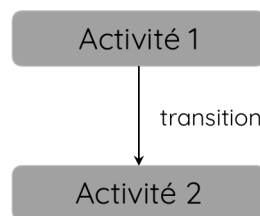


FIGURE 1 – Transitions automatiques

Une fois l'activité 1 terminée, on passe automatiquement à l'activité 2.

### 1.2 Début et Fin

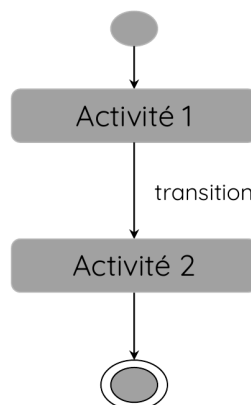


FIGURE 2 – Début et Fin

Le *rond gris* représente le **début** du programme, et le *rond gris entouré d'un cercle* représente la **fin** du programme (fin normale ou fin après erreur).

### 1.3 Conditions

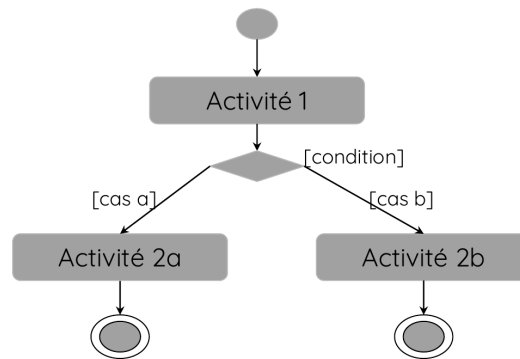


FIGURE 3 – Conditions

Arrivé au *losange* (=condition), il y a **deux possibilités**, soit **cas a**, soit **cas b**. (If, Then, Else)

### 1.4 Itérations (boucles)

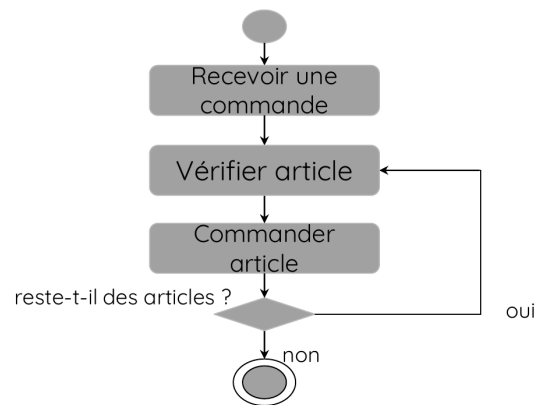


FIGURE 4 – Itérations

Les *instructions* seront exécutées **en boucle** tant que la *condition* (*losange*) est **vraie**. (For, While)

## 2 Classes et objets

### 2.1 Classes

```
1 //Exemple de classe Compte en Java
2 public class Compte {
3     //solde > decouvertMax
4     private int numero;
5     private int solde;
6     private int decouvertMax;
7
8     public Compte(int numero, int solde) {
9         //...
10    }
11
12    public int consulterSolde() {
13        //...
14        return int;
15    }
16 }
```

```

16
17 public void crediter(int somme) {
18     //...
19 }
20
21 public void debiter(int somme) {
22     //... (solde > decouvertMax)
23 }
24 }

```

Représentation UML de la classe Compte :

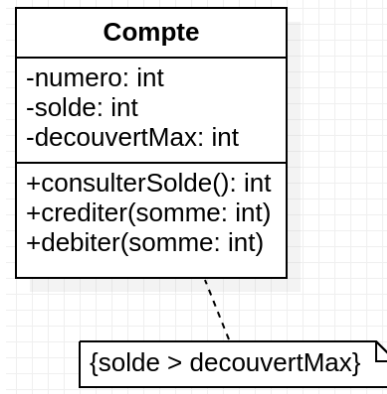


FIGURE 5 – Classes

### 2.1.1 Visibilités

Il existe 4 types de visibilité :

+	public
-	private
~	package
#	protected

1. **public +**  
La visibilité *public* signifie que les données sont accessibles depuis n'importe où.
2. **private -**  
La visibilité *private* signifie que les données ne sont accessibles que depuis la même classe.
3. **package ~**  
La visibilité *package* signifie que les données ne sont accessibles que depuis le même package.
4. **protected #**  
La visibilité *protected* signifie que les données ne sont accessibles que depuis le même package ou les sous-classes.

### 2.1.2 Attributs

Les attributs d'une classe sont notés de la manière suivante :

Visibilité Type Nom ;

Ex :

private int numero ;

### 2.1.3 Opérations

Les opérations d'une classe sont notées de la manière suivante :

Visibilité Type de retour Nom (Parametres) { ... }

Ex :

public String getNom(int numero) { ... }

(Cette méthode prend donc un int en paramètre, et retournera un String)

## 2.2 Objets

Exemple d'objets de la classe **Compte** (vue plus haut) :

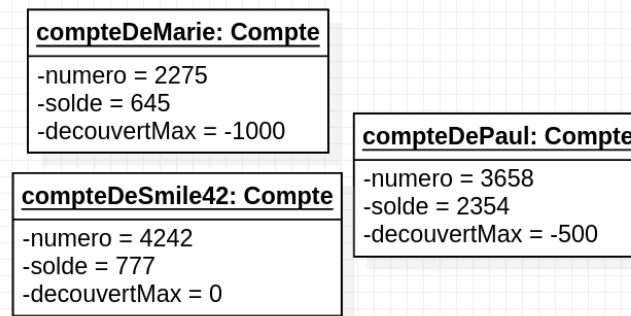


FIGURE 6 – Objets

### 1. Identité :

Permet d'identifier un objet.

### 2. État :

Caractéristiques de l'objet à un moment donné (Valeurs des attributs).

### 3. Comportement :

Ensemble des opérations qu'un objet peut exécuter ou subir.

- Create
- Read
- Update
- Delete

## 3 Associations

Connexions sémantiques durables entre des classes.

### 3.1 Associations 1-1 et 1-N

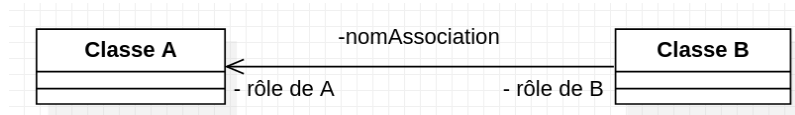


FIGURE 7 – Associations 1-1 1-N

Exemple d'associations (*Java* + *UML*) :

```
1 //Classe Client
2 public class Client {
3     private final String nom;
4     private final String prenom;
5     private final Compte compte;
6
7     public Client(String nom, String prenom, Compte compte) {...}
8     public String getNom() {...}
9     public String getPrenom() {...}
10    public Compte getCompte() {...}
11 }
```

```
1 //Classe Compte
2 public class Compte {
3     private final String numero;
```

```

4  private int solde;
5  private final Client titulaire;
6
7  public Compte(String numero, int solde, Client titulaire) {...}
8  public String getNumero() {...}
9  public int getSolde() {...}
10 public Client getTitulaire() {...}
11 }

```

Représentation UML de l'association des classes Client et Compte :

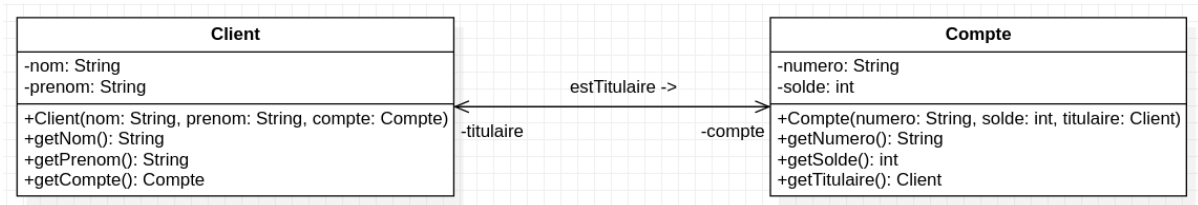


FIGURE 8 – Associations Client Compte (Exemple)

Ainsi, si on crée un Client "pierre" et qu'on lui attribue le compte "c1", on peut dire ceci :

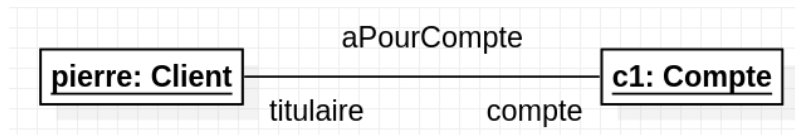


FIGURE 9 – Pierre : c1

- pierre a pour compte c1
- c1 joue le rôle de compte pour pierre
- pierre joue le rôle de titulaire pour c1
- le (un des) compte(s) de pierre est c1
- le (un des) titulaire(s) de c1 est pierre