

Cours de Programmation Orientée Objet JAVA

Demba SOW

*Docteur en
Mathématiques et en
Cryptologie*

L.A.C.G.A.A.

F.S.T. / U.C.A.D.



Plan du cours

1

- Introduction à Java

2

- Techniques de base du langage

3

- Les types primitifs

4

- Les structures de contrôle

5

- Programmation Orientée Objet

6

- Les tableaux

7

- Les chaînes de caractères

Plan du cours (suite)

8

- L'héritage

9

- Gestion des exceptions

10

- Les Swing GUI

11

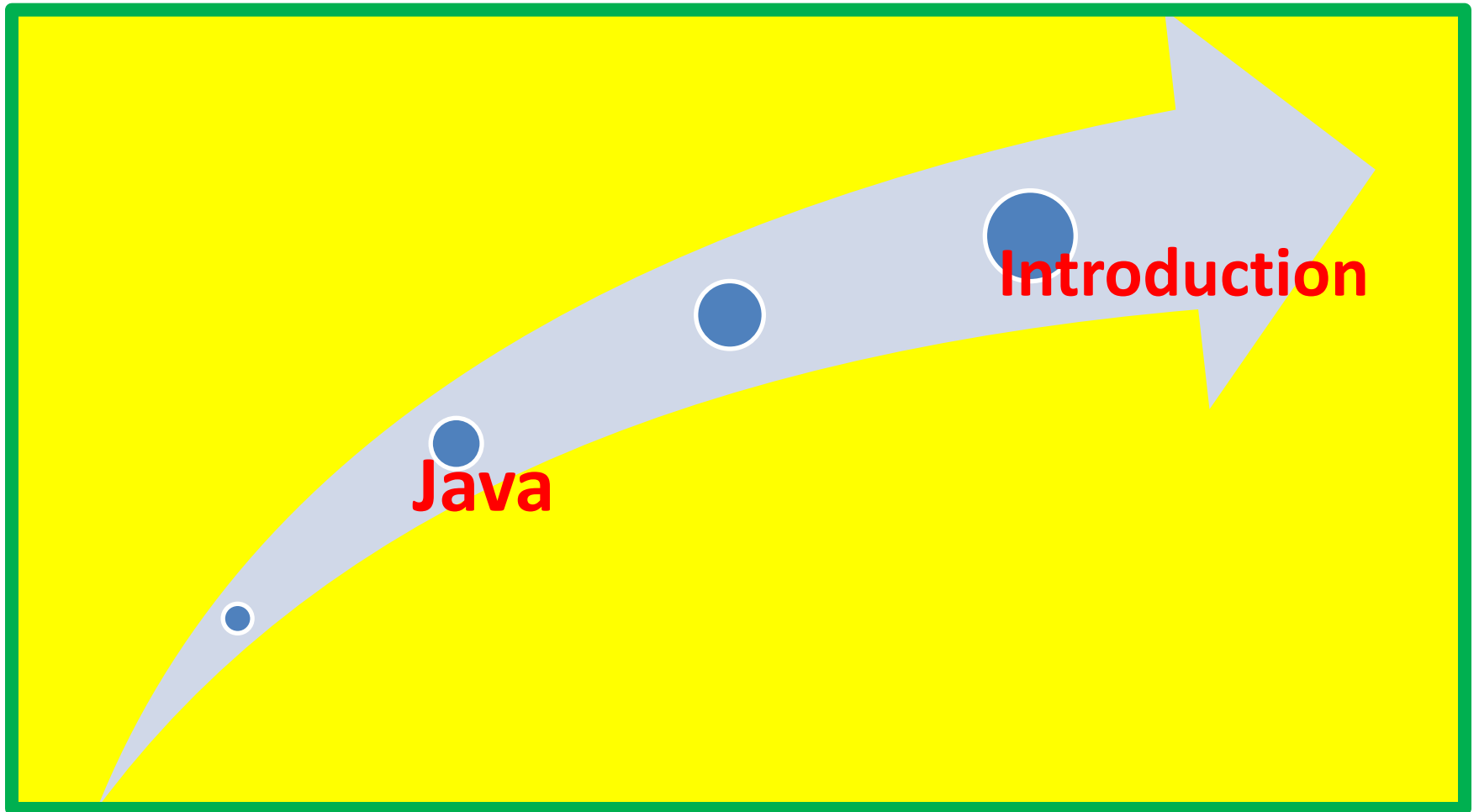
- Entrées Sorties

12

- Applets

13

- JDBC



Historique : Origines de Java

► 1990

- ✱ Internet très peu connu, **World Wide Web** inexistant .
- ✱ boom des PCs (puissance)
- ✱ Projet Oak de SUN Microsystems
 - *Langage pour la communication des appareils électroniques .*

► 1993

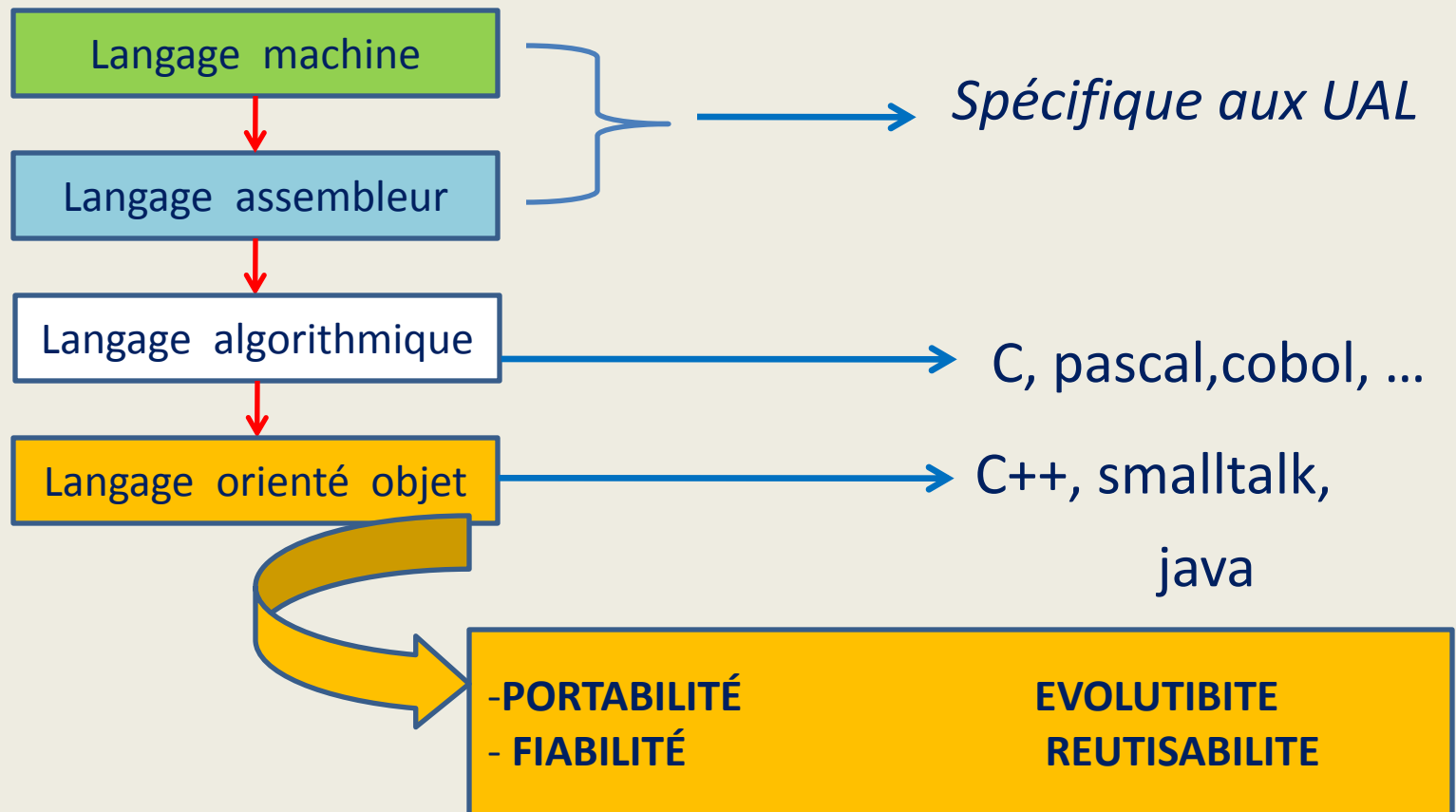
- ✱ mars : le NCSA lance MOSAIC, le premier navigateur internet
- ✱ (protocole http, langage html), le web décolle...
- ✱ été : Oak change d'orientation et s'adapte à la technologie internet

► 1995

- ✱ mai 1995 : projet *HotJava*, navigateur WEB, écrit par SUN en Java .

Les générations des langages de programmation

■ *Plusieurs langages de programmation ont vu le jour :*

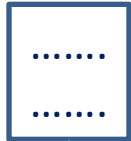


Java : un langage objet

- ➡ ***Imprégné du C++ mais améliorant ses insuffisances***
 - ✳ gestion automatique de la mémoire (Garbage Collector)
 - ✳ facilité de stockage des fichiers sur disque (sérialisation)
- ➡ ***Une gigantesque API (Application Programming Interface)***
 - ✳ une librairie de classes très importante (interface graphique, réseau, web, base de données, ...)
 - ✳ portabilité sans mesure
 - ✳ langage de plus en plus utilisé et évoluant rapidement.

La Machine Virtuelle java (JVM)

Code
source
coté
Dévelop
peur



Compilation en
PSEUDO-CODE
(byte-code)

java-compiler(javac)

Interprétation par la
Machine virtuelle Java pour
les
Systèmes d'exploitation

**Le pseudocode est
universel c'est à dire
indépendant du
compilateur et de
l'ordinateur où a lieu
cette compilation.**

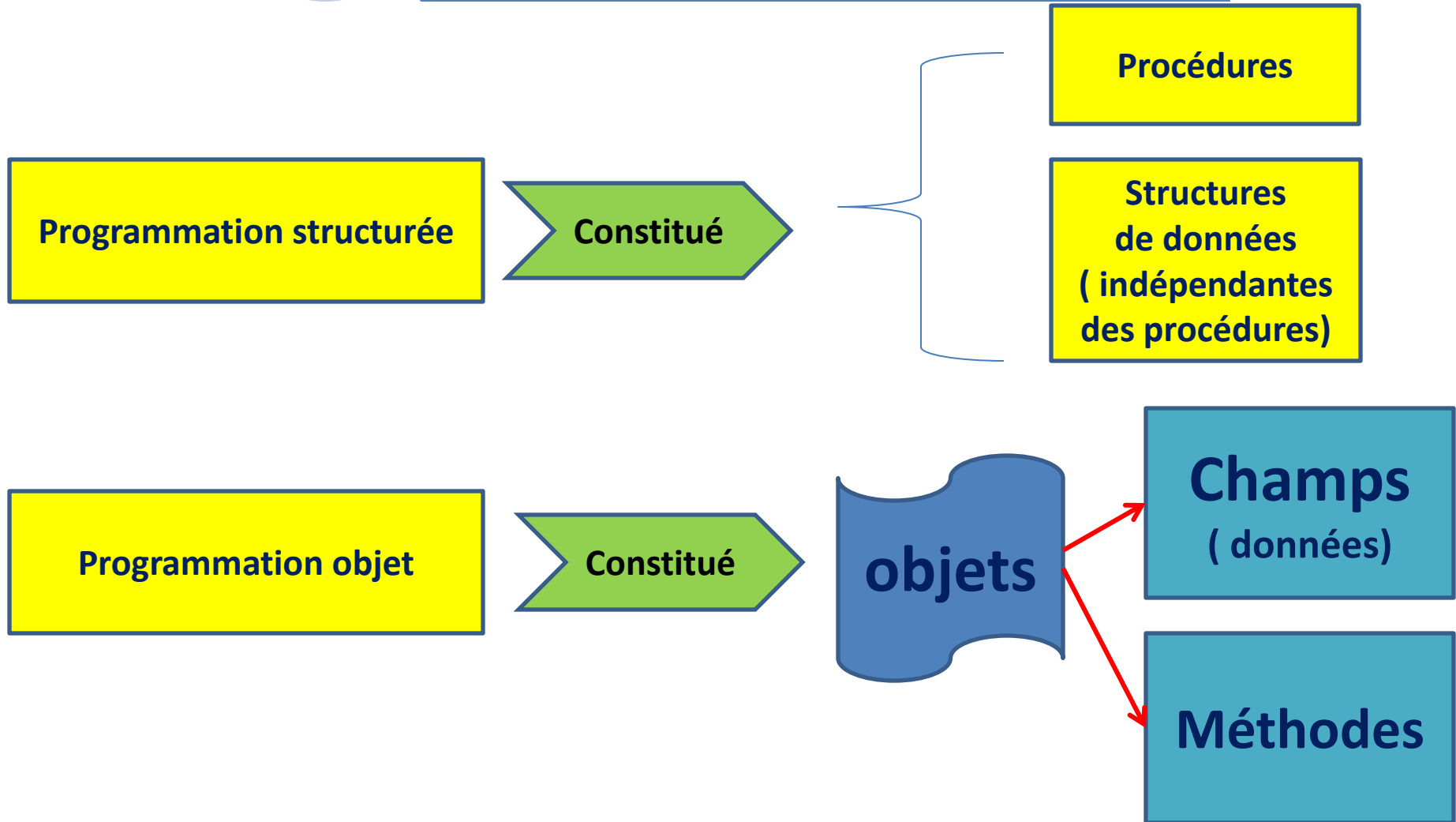
processeurs
coté
utilisateurs

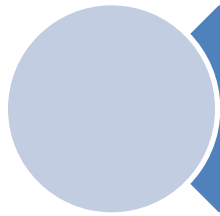
PC

Macintosh

UNIX

La Programmation Orientée Objet (P.O.O)



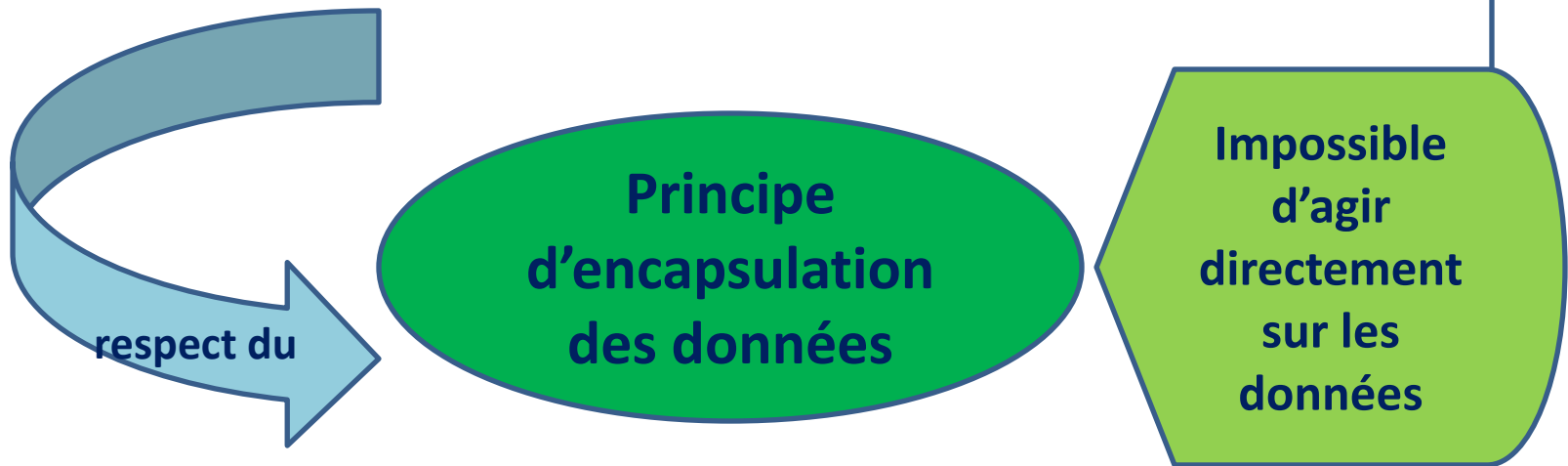


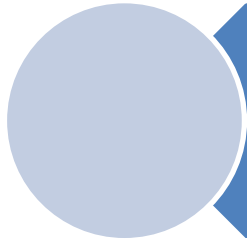
Concept de classe

- Le concept de **classe** correspond simplement à la **généralisation de type** que l'on rencontre dans les langages classiques. En effet, une classe n'est rien d'autre que **la description d'un ensemble d'objets ayant une structure de données commune et disposant des mêmes méthodes**.
- Les **objets** apparaissent alors comme **des variables d'un tel type de classe** (en P.O.O, on dit aussi qu'un objet est une *instance* de sa classe). Bien entendu, seule la structure est commune , **les valeurs des champs étant propres à chaque objet**. En revanche, les méthodes sont communes à l'ensemble des objets d'une même classe.

La P.O.O: l'encapsulation

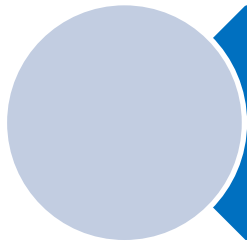
Les méthodes agissent exclusivement
sur les champs de l'objet





JRE (Java Runtime Environment)

- Le **JRE** contient uniquement ***l'environnement d'exécution de programmes Java.***
- *Le **JDK** contient lui même le **JRE**. Le **JRE** seul doit être installé sur les machines où des applications java doivent être exécutées.*
- Depuis sa version 1.2, **Java** a été renommé **Java 2**. Les numéros de versions **1.2** et **2** désignent donc la même version.
- Le **JDK** a été renommé **J2SDK** (***Java 2 Software Development Kit***) mais la dénomination **JDK** reste encore largement utilisée.
- Le **JRE** a été renommé **J2RE** (***Java 2 Runtime Edition***).
- **Trois éditions de Java existent :**
 - ***J2ME** : Java 2 Micro Édition ·*
 - ***J2SE** : Java 2 Standard Édition ·*
 - ***J2EE** : Java 2 Entreprise Édition ·*
- **Sun** fourni le **JDK**, à partir de la version 1.2, sous les plateformes ***Windows, Solaris*** et ***Linux.***
- On parle actuellement de **Java 7** et **Java 8**.



Java 2 en trois éditions différentes

J2ME(Java2 Micro Édition)



Applications sur environnement limité

- systèmes portables
- systèmes de navigation embarqués

J2SE(Java2 Standard Édition)



-Applications

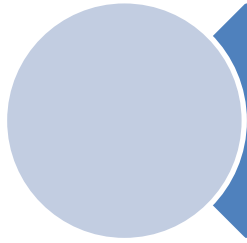
-Applet

J2EE(Java2 Enterprise Édition)



-API pour applications d'entreprise
(accès bases de données)

- EJB(composants métiers)
- JSP(Java Server Pages)
- Servlet (HTML dynamique)



L'API

L'**A.P.I.** (**A**pplication **P**rogramming **I**nterface) est un ensemble de classes utilisables par le programmeur .

Un programme JAVA est constitué d'un certain nombre de classes :

- des classes **prédéfinies** de l'API (environ un millier de classes) (Application Programming Interface) .
- des classes **définies** par l'utilisateur

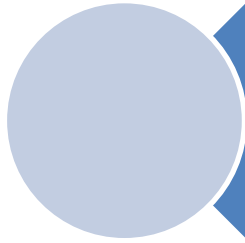
Un programme utilise
au maximum
les fonctionnalités
présentes dans l'A.P.I.

PROGRAMME

classes définies
pour ce
programme

API

classe : type de
données
+ fonctions pour
le manipuler



Applications et Applets

- Tout programme est soit une application soit une applet

APPLICATION

Les applications autonomes sont des programmes exécutés sous le contrôle du système d'exploitation. Les programmes **C**, **C++** ... sont des applications de ce type.

APPLET

Les applets sont exécutées lors de la visualisation par un navigateur **WEB** d'une page **HTML**.

Applications: la console vs G.U.I

**Programme à interface
console**

L'utilisateur fournit des infos au clavier sous forme de lignes de texte. Le programme décide du séquençement des opérations. L'opérateur est sollicité au moment voulu.

**Programme à interface
graphique : GUI
(Graphical User Interface)**

L'interaction programme opérateur se fait essentiellement via des *composants* graphiques. C'est la programmation événementielle: le programme réagit à des événements provoqués par l'utilisateur.

