

Programmation I

Dr Khadim DRAME

Département d'Informatique
UFR des Sciences et Technologies
Université Assane Seck de Ziguinchor

26 mai 2021



- EC de l'UE **Algorithmique et Programmation 1**
- Déroulement des enseignements
 - CM : 8 séances de 1h30
 - TD : 8 séances de 1h30
 - TP : 8 séances de 1h30
- Évaluation
 - Contrôle continu (30%)
 - Examen final (70%)



Objectifs

- Objectif général
 - A la fin du cours, l'étudiant sera capable de **concevoir** et d'**implémenter** des programmes (en Pascal) et de les **exécuter**
- Objectifs spécifiques
 - 1 décrire les concepts fondamentaux de la programmation
 - 2 décrire la structure d'un programme impératif
 - 3 distinguer les différentes étapes de conception d'un programme
 - 4 concevoir des programmes permettant de résoudre des problèmes simples
 - 5 implémenter des programmes simples avec le langage Pascal
 - 6 compiler et exécuter des programmes codés en Pascal



Plan du cours

- 1 Introduction à la programmation
- 2 Objets, expressions et instructions de base
- 3 Structures conditionnelles
- 4 Structures itératives
- 5 Tableaux et chaînes de caractères
- 6 Sous-programmes



Introduction à la programmation

Dr Khadim DRAME

Département d'Informatique
UFR des Sciences et Technologies
Université Assane Seck de Ziguinchor

26 mai 2021



Plan

- 1 Algorithmique et programmation
- 2 Structure générale d'un programme en Pascal

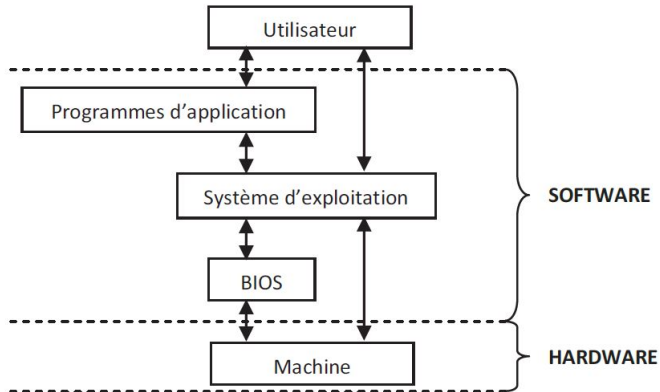


Définitions

- L'**informatique** : traitement *automatique* de l'*information*.
- Un **ordinateur** est une machine capable de traiter automatiquement l'information.
- L'ordinateur est constitué de :
 - composantes matérielles (**hardware**) : RAM, ROM, disque dur, écran, etc ;
 - composantes logicielles (**software**) : programmes stockés dans la mémoire de l'ordinateur qui assurent le fonctionnement du matériel.
- Un **système d'exploitation** est un ensemble de programmes capables de gérer l'ordinateur.



Composantes d'un ordinateur



Algorithme/algorithmique

Algorithme

- Un algorithme est une suite finie et ordonnée d'**instructions** dont l'exécution permet de résoudre un problème donné.
- Un algorithme est décrit dans un langage algorithmique ; il est conçu indépendamment d'un langage de programmation.



Algorithmique

L'algorithmique est la science qui étudie l'application des algorithmes en informatique.

Qualités d'un algorithme

- **Lisibilité**

- Aptitude à être compréhensible pour un développeur

- **Exactitude**

- Aptitude à fournir le résultat voulu

- **Robustesse**

- Aptitude à bien réagir dans des conditions anormales d'utilisation

- **Réutilisabilité**

- Aptitude à être réutilisé dans d'autres problèmes

- **Efficacité**

- Aptitude à utiliser de manière optimale les ressources



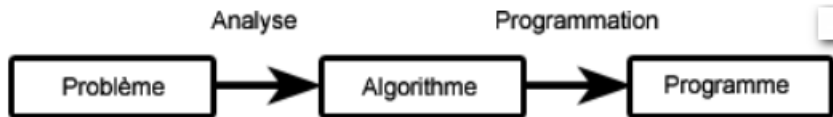
Programmation

Programme

Un programme est la réalisation d'un algorithme dans un langage compréhensible par la machine (langage de programmation).

Programmation

- La programmation est l'activité consistant à écrire des programmes dans un langage de programmation.
- Elle comprend 2 étapes : l'écriture d'un algorithme et sa traduction dans un langage de programmation.



Langage de programmation

- Un langage de programmation est un langage informatique, qui permet d'écrire un code source compréhensible par la machine.
- Un langage de programmation est défini très strictement par une norme spécifiant sa syntaxe et sa sémantique.

Exemples (langages de programmation)

Pascal, C, C++, Python, Java.



Composition d'un langage

Un langage de programmation est constitué de :

- un **alphabet** qui désigne les caractères utilisés par le langage ;
- un **vocabulaire** (lexique) qui désigne l'ensemble des mots du langage ;
- une **grammaire** (syntaxe) qui définit les règles de construction des déclarations et instructions ;
- une **sémantique** qui définit le sens donné aux instructions.



Langages de programmation

Familles de langages

Il existe un grand nombre de langages et styles de programmation regroupés en familles (paradigmes) :

- Programmation impérative (Pascal, C) ;
- Programmation objet (C++, Java) ;
- Programmation fonctionnelle (Lisp, Caml).

Niveau de langage

Selon leur proximité avec le langage machine, on distingue :

- les langages de **bas niveau** qui manipulent directement la mémoire et le processeur ;
- les langages de **haut niveau** (évolués) permettant d'utiliser des concepts plus abstraits.

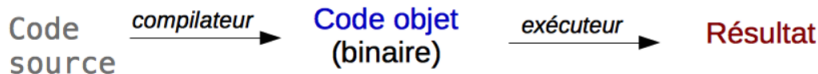
Compilation

Compilation

La compilation consiste à traduire le code source d'un programme en un programme exécutable par la machine (code objet ou programme objet).

Compilateur

Un compilateur est un programme informatique qui transforme un code source (généralement écrit dans un langage de programmation) en un code objet (écrit dans un langage cible de bas niveau).



Langage compilé/langage interprété

Langage compilé

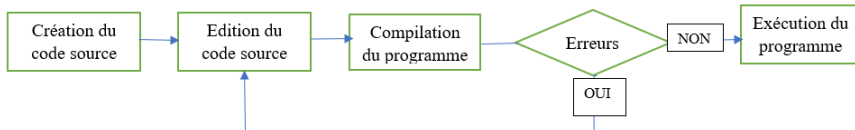
- Un langage est dit compilé lorsqu'il s'appuie sur un compilateur qui transforme le code source du programme en code machine et produit ainsi un fichier exécutable par l'ordinateur.
- Exemples de langages compilés : Pascal, C, C++.

Langage interprété

- Un langage est dit interprété lorsqu'il s'appuie sur un interpréteur qui exécute le code instruction par instruction.
- L'interpréteur ne produit pas de fichier exécutable, il traduit les instructions une à une et les exécute à la volée.
- Exemples de langages interprétés : Bash, Python, Lisp.

Cycle de développement d'un programme

- 1 Écrire le code source du programme ;
- 2 Éditer éventuellement le code source ;
- 3 Compiler le programme ;
- 4 Exécuter et tester le programme.



- La mise en place d'un programme nécessite l'usage de plusieurs outils :
 - un **éditeur** de texte pour la saisie et l'édition du code source ;
 - un **compilateur** (ou **interpréteur**) pour la traduction du code ;
 - un **debbuger** qui aide le développeur à analyser les bugs (erreurs).
- Ces outils sont généralement intégrés dans un logiciel appelé **Environnement de Développement Intégré** (EDI ou IDE en anglais).

Exemples d'EDI : **Lazarus**, DevPascal, Turbo Pascal.

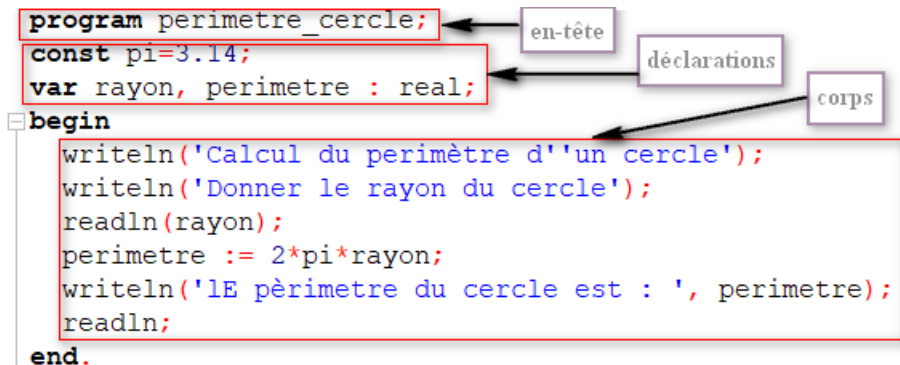


Plan

- 1 Algorithmique et programmation
- 2 Structure générale d'un programme en Pascal



Structure d'un programme en Pascal



Structure d'un programme en Pascal

Un programme Pascal est composé de :

- un **en-tête** constitué du mot clef **program** suivi du **nom** (identificateur) du programme ;
- une partie dite **déclaration** qui comprend :
 - la déclaration des **constantes** ;
 - la déclaration des **variables** ;
- le **corps** du programme :
 - Il est constitué d'instructions (partie exécutable) ;
 - il est compris entre les mots clés **begin** et **end** et se termine par un point.



Structure d'un programme en Pascal

Remarques

- L'en-tête et le corps du programme sont **obligatoires**.
- Les déclarations sont **optionnelles** et dépendent des besoins du développeur.
- Les déclarations et les instructions terminent par un point virgule (sauf exceptions).

Commentaire

- Les commentaires sont utilisés pour expliquer le code.
- Ils sont placés entre accolades ({ et }) ou entre (* et *).
- Ils ne sont pas exécutés, ils sont ignorés par le compilateur.
- Les commentaires sont **optionnels**.

Exemple de programme en Pascal

```
1 program max2;{programme cherchant le maximum entre
   deux nombres}
2 var
3   x, y : real;
4 begin
5   write('Donner deux nombres ');
6   readln(x,y);
7   if x>=y then
8     write('Le maximum est ',x)
9   else
10    write('Le maximum est ',y);
11   readln();
12 end.
```

Listing 1 – Exemple de programme Pascal



A retenir

- Un algorithme est une suite finie et ordonnée d'instructions pour résoudre un problème donné.
- Un algorithme est indépendant d'un langage de programmation.
- Un programme est la traduction d'un algorithme dans un langage de programmation.
- Il existe différentes familles de langages de programmation.
- L'ordinateur ne comprend que le langage binaire.
- La compilation permet de traduire un programme écrit dans un langage de programmation en un programme en langage machine.
- Un programme comprend trois parties : une entête, des déclarations et un corps.



Un programme est réalisé pour effectuer un ensemble de **traitements** sur des **données**

- quels **types** de données utilisés ? comment fournir ces **données** au programme ?
- quels **types** de données retournés ? comment communiquer le(s) **résultat(s)** du programme ?
- comment ces **traitements** sont-ils effectués ?

