INITIATION AU NUMÉRIQUE

COURS D'INTRODUCTION

PLAN

- Qu’est-ce que programmer?
- Concepts et principe de fonctionnement d'un programme informatique.
- Types et exemples de langages de programmation.
- Environnement de programmation.

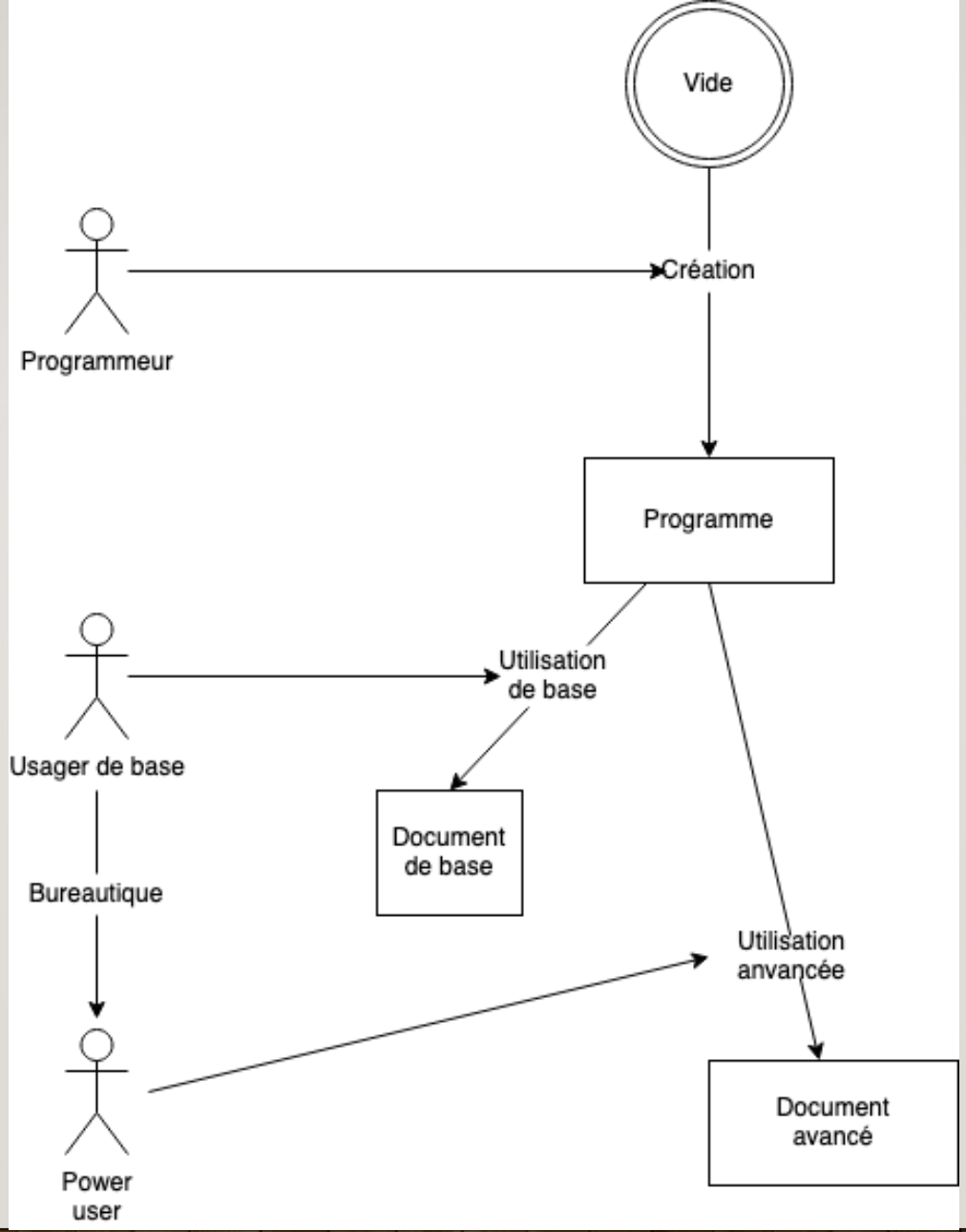
QU'EST-CE QUE PROGRAMMER?

PROGRAMMER - DÉFINITION

- Programmer, c'est partir de zéro, d'une page blanche, et de composer du code jusqu'à ce qu'un ordinateur exécute un travail précis.

- La principale fonction d'un programmeur est de réfléchir de façon logique et de **construire** une démarche, comme une construction de blocs Légo.

- **Un programmeur n'est pas un super utilisateur**



CONCEPTS ET PRINCIPE DE FONCTIONNEMENT D'UN PROGRAMME INFORMATIQUE

LANGAGE - DÉFINITION

- Un ordinateur ne peut comprendre que des zéros et des uns (0 ou 1).
- Un compilateur ou un interpréteur est un programme qui **traduit** du texte en 0 et 1 exécutable par l'ordinateur.
- Un langage est une **syntaxe textuelle** à respecter pour qu'un compilateur ou un interpréteur comprenne notre texte et puisse le traduire:
  - Liste de mots reconnus par le compilateur.
  - Ordonnancement des mots reconnu par le compilateur.

INSTRUCTION

- Définition:
  - Ordre donné à un ordinateur pour exécuter une tâche précise.
  - Étape minimaliste qui ne peut pas se diviser en sous-tâche.
- Exemples:
  - Additionner 2 valeurs.
  - Ouvrir un fichier.
  - Comparer 2 valeurs.
  - Afficher un message.

CODE - DÉFINITION

- Suite d'instructions (voir diapo précédente) qui, prises dans leur ensemble, exécutent un travail utile.

CODE - EXEMPLE

1. Ouvrir un fichier;
2. Lire une valeur;
3. Ajouter la valeur à un total;
4. Recommencer à l'étape 2 jusqu'à la fin du fichier;
5. Afficher le total;

PRODUCTION D'UN CODE

- Pour produire du code, il faut:
  1. Réfléchir à ce qu'on veut faire.
  2. Diviser notre démarche en étapes (algorithmique).
  3. Traduire nos étapes en instructions qui respectent le langage choisi.
  4. Compiler le code (pour certains langages).
  5. Exécuter le code.

ENTRÉES/SORTIES

- Pour la majorité des programmes, l'exécution nécessite d'interagir avec des éléments externes:
  - Utilisateur (clavier, souris, écran).
  - Fichier (ouvrir, lire, écrire, enregistrer, fermer).
  - Carte réseau (transmission de messages, réception de messages).
- La majorité des langages offrent des services qui permettent de créer des instructions pour interagir avec les éléments externes.

TYPES ET EXEMPLES DE LANGAGES DE PROGRAMMATION

LANGAGES COMPILÉS

- Les langages compilés possèdent des programmes qui traduisent le texte respectant sa syntaxe en fichier exécutable.
- Ce fichier est "final" en ce sens qu'on a plus besoin du code initial pour le faire fonctionner après compilation.
- La compilation est faite pour une seule plate-forme (Windows ou Mac ou Linux). Il faut indiquer la plate-forme de destination.
- Pour exécuter sur plusieurs plates-formes, il faut recompiler.

LANGAGES COMPILÉS - EXEMPLES

- C
  - Un des plus vieux langages encore utilisé.
  - Développé fin 1960/début 1970 pour programmer UNIX.
- SmallTalk
  - Un des premiers langages orientés-objets.
  - Un des premiers langages à utiliser l'affichage graphique.

LANGAGES COMPILÉS - EXEMPLES

- Objective-C
  - Basé sur le C et le Smalltalk
  - Utilisé surtout sur les Macs.
- C#
  - Version Microsoft de Java.

LANGAGES INTERPRÉTÉS

- Langages qui sont traduits dynamiquement (“live”) lors de l’exécution.
- Nécessitent un interpréteur pour la plate-forme utilisée.
- Peuvent être distribués indépendamment de la plate-forme:
  - Le code est universel.
  - L’interpréteur est adapté à la plate-forme.

LANGAGES INTERPRÉTÉS - EXEMPLES

- Bash
  - Série de commandes exécutables dans un terminal sur les plates-formes UNIX (Linux, Mac, Solaris).
- PowerShell
  - Série de commandes exécutables dans un terminal sur les plates-formes Windows (maintenant disponible pour Linux et Mac).
- html, xml, xhtml
  - Langage utilisé dans les pages web.

LANGAGES INTERPRÉTÉS - EXEMPLES

- Javascript:
  - Un des langages de scripts interprétés les plus utilisés.
  - À l'origine → sous-langage de Java utilisé pour faire des pages web dynamiques du côté client; syntaxe très limitée.
  - Par la suite → le langage a évolué fortement compte tenu de sa grande utilisation.
  - Maintenant → langage complet totalement indépendant de Java, utilisable:
    - dans des pages web (avec un interpréteur intégré au navigateur);
    - dans des programmes indépendants (avec un interpréteur spécialisé).

LANGAGES INTERPRÉTÉS - EXEMPLES

- Python :
  - Créé par Guido van Rossum (fin des années 1980; début1991).
  - Principale caractéristique : lisibilité du code en forçant l'indentation.
  - Supporte la programmation structurée, fonctionnelle et orienté-objet.
  - Contient de très nombreuses librairies.

- Source : https://en.wikipedia.org/wiki/Python_(programming_language)

LANGAGES HYBRIDES (COMPILATION JIT)

- Langages ayant un mode mixte:
  - Le code subit une transformation semblable à la compilation, mais destinée à un interpréteur.
  - L'interpréteur exécute le code semi-compilé, en fonction de la plate-forme.
- Exemple : Java
  - Basé sur le C et le Smalltalk
  - Développé pour faire des programmes réseaux.

ENVIRONNEMENT DE PROGRAMMATION

LIBRAIRIES

- Code exécutable conçu et distribué par divers programmeurs, pour faciliter la production de programme.
- Des programmeurs composent du code pour des problèmes récurrents, afin que d'autres programmeurs n'aient pas à toujours "réinventer la roue".
- Lors de la production d'un programme, un programmeur rend les librairies disponibles à son environnement de développement puis il lie les librairies nécessaires et il les utilisent sans avoir à les coder.

IDE

- Acronyme de Interactive Development Environnement.
- À l'origine → les programmeurs devaient :
  - Programmer dans des éditeurs de texte brut;
  - Compiler avec un compilateur spécialisé (si langage compilé);
  - Exécuter, corriger la syntaxe puis déboguer manuellement.
- Maintenant → les IDE intègrent les outils permettant de tout faire.

IDE

- Dans ce cours, nous utiliserons:
  - Geany pour débuter le développement
    - Très simple et léger.
  - PyCharm de JetBrains pour le développement avancé
    - Plus lourd à faire fonctionner;
    - Possède un débogueur avancé.

QUESTIONS?

- Merci de votre attention