

Les bases de la programmation

Formateur : Nizar AYED - Juin 2017

nizar.ayed@upgrade-code.org

+33687574635

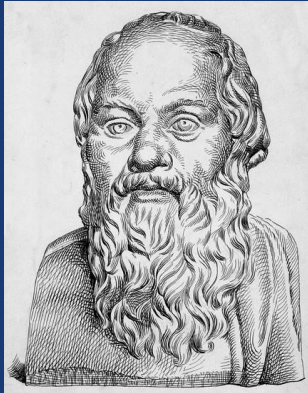
JOUR 1

Formation Développeur Web

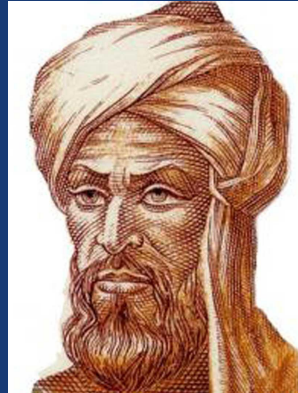
Introduction

Commençons par le commencement

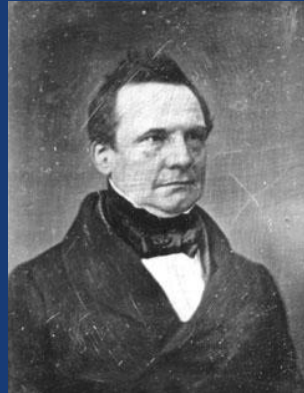
Quand l'homme a-t-il commencé à programmer ?



Socrate
Vème Siècle av. JC



Al Khawarizmi
780-850 ap. JC



Charles BABBAGE
1791-1871



Ada LOVELACE
1815-1852

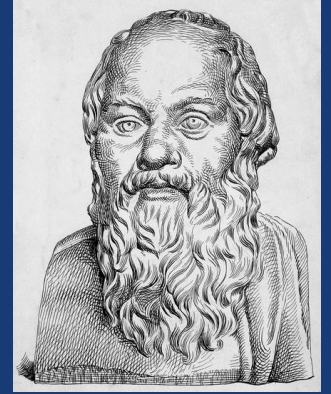
Ils ont tous contribué à l'essor de l'informatique actuelle!

Le test de Socrate

Le test des 3 passoires pour bloquer les rumeurs

- Test de vérité
- Test de bonté
- Test d'utilité

⇒ Ce test philosophique est un exemple de raisonnement logique comparable au raisonnement avec les ordinateurs

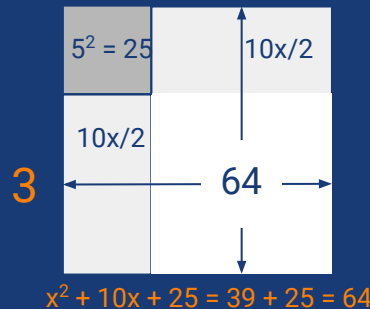
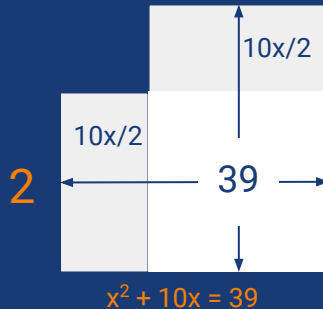
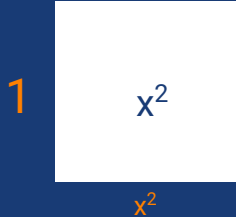


Socrate
Vème Siècle av. JC

Le 1er algorithme

- Les algorithmes ont existé depuis longtemps
- Mais c'est le premier à être formulé sous forme d'étapes séquentielle : Algorithme de résolution d'une équation de 2nd degré!
- Exemple :

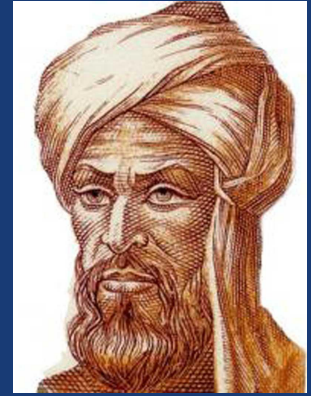
$$x^2 + 10x = 39$$



$$8 = 5 + x$$

Donc

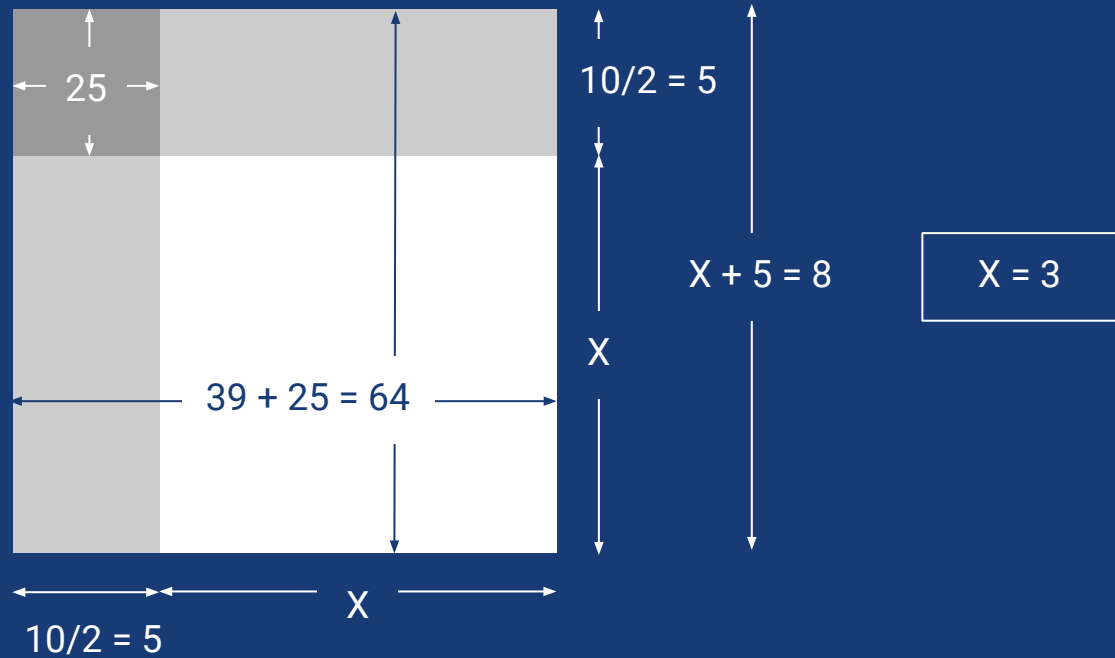
$$x = 3$$



Al Khawarizmi
780-850 ap. JC

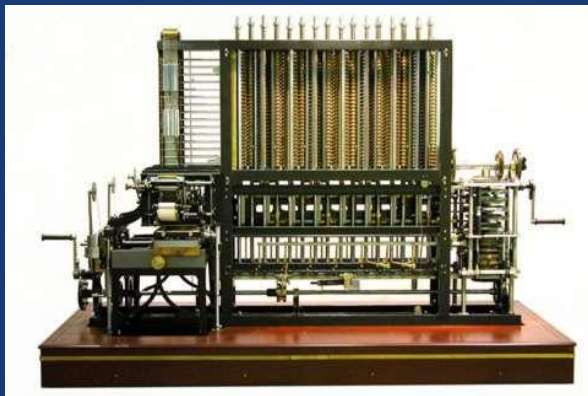
Etape par étape

$$X^2 + 10X = 39$$



Le premier calculateur

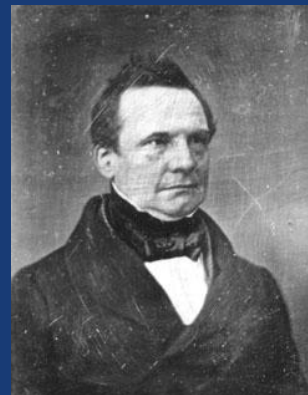
Une partie de la machine a été réalisée après sa mort par un de ses fils et était fonctionnelle vers 1905. La machine a depuis été construite (vers 1990), prouvant qu'une architecture d'ordinateur entièrement mécanique est possible.



Machine de Babbage



Cartes perforées pour
entrer les données



Charles BABBAGE
1791-1871

Et le premier programmeur (enfin, première...)

Son amour pour les mathématiques lui a permis de rédiger le premier algorithme pour la machine analytique de Babbage. Bien sûr, c'était sur papier et jamais il n'a été réellement exécuté.



Ada LOVELACE
1815-1852

Columns above which are written functions of the variable	Coefficients		Cards of the operations		Cards of the variables			
	Given	To be formed	No. of the Operations	Nature of the Operation	Columns on which operations are to be performed	Columns on which are to be inscribed the results of the operations	Indication of change of value in any column submitted to an operation	Results of the operations
$x^0 \dots\dots\dots 1V_0$	a	"	"	"	"	"	"	"
$x^1 \dots\dots\dots 1V_1$	b	"	"	"	"	"	"	"
$\cos^0 x \dots\dots 1V_2$	A	"	"	"	"	"	"	"
$\cos^1 x \dots\dots 1V_3$	B	"	"	"	"	"	"	"
$x^0 \cos^0 x \dots\dots 0V_4$	aA	1	\times	$1V_0 \times 1V_2 =$	$1V_4 \dots\dots$	$\begin{cases} 1V_0 = 1V_0 \\ 1V_2 = 1V_2 \end{cases}$	$1V_4 = aA$	coefficients of $x^0 \cos^0 x$
$x^0 \cos^1 x \dots\dots 0V_5$	aB	2	\times	$1V_0 \times 1V_3 =$	$1V_5 \dots\dots$	$\begin{cases} 1V_0 = 0V_0 \\ 1V_3 = 1V_3 \end{cases}$	$1V_5 = aB$	$\dots\dots\dots x^0 \cos^1 x$
$x^1 \cos^0 x \dots\dots 0V_6$	bA	3	\times	$1V_1 \times 1V_2 =$	$1V_6 \dots\dots$	$\begin{cases} 1V_1 = 1V_1 \\ 1V_2 = 0V_2 \end{cases}$	$1V_6 = bA$	$\dots\dots\dots x^1 \cos^0 x$
$x^1 \cos^1 x \dots\dots 0V_7$	bB	4	\times	$1V_1 \times 1V_3 =$	$1V_7 \dots\dots$	$\begin{cases} 1V_1 = 0V_1 \\ 1V_3 = 0V_3 \end{cases}$	$1V_7 = bB$	$\dots\dots\dots x^1 \cos^1 x$

Et l'histoire que l'on connaît

1947 : 1er ordinateur moderne - L'ENIAC (167m² pour 30 tonnes)

1971 : 1er processeur - Intel 4004 (comparable à l'ENIAC)

1975 : Création de Microsoft

1976 : Création de APPLE

1989 : Création du WWW (World Wide Web)

1993 : Lancement du Navigateur MOSAIC (actuellement Firefox)

2007 : Lancement de l'iPhone

2015 : 1ère Victoire de l'Intelligence Artificielle contre un humain au jeu de GO

En enfin, fût l'informatique moderne!

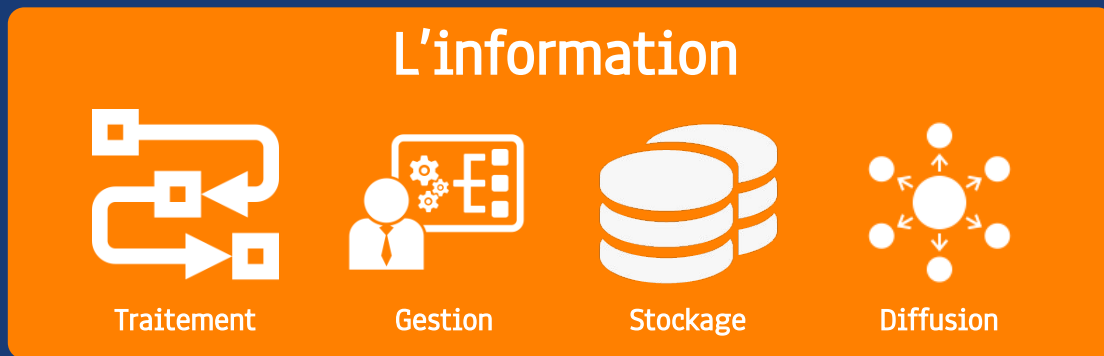
Sommaire

1. Système d'information
2. Architectures des SI
3. Approche agile
4. Pensée algorithmique
5. Programmation orientée objet
6. API / Services REST
7. Gestion des sources / GIT
8. Sublime text.

Les systèmes d'information

Définition

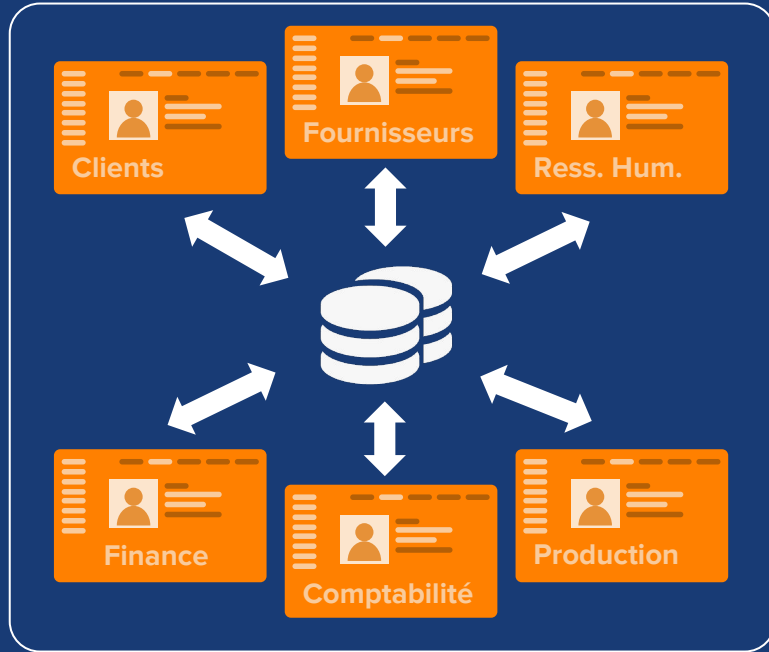
Un **Système d'Information** est tout **ensemble identifiable** de ressources tant matérielles, humaines et autres, d'une organisation, et ayant pour rôle de **traiter**, **gérer**, **stocker** et **diffuser** l'information que ce soit du milieu extérieur de l'organisation vers cette dernière, que ce soit dans l'organisation et enfin de l'organisation vers l'extérieur.



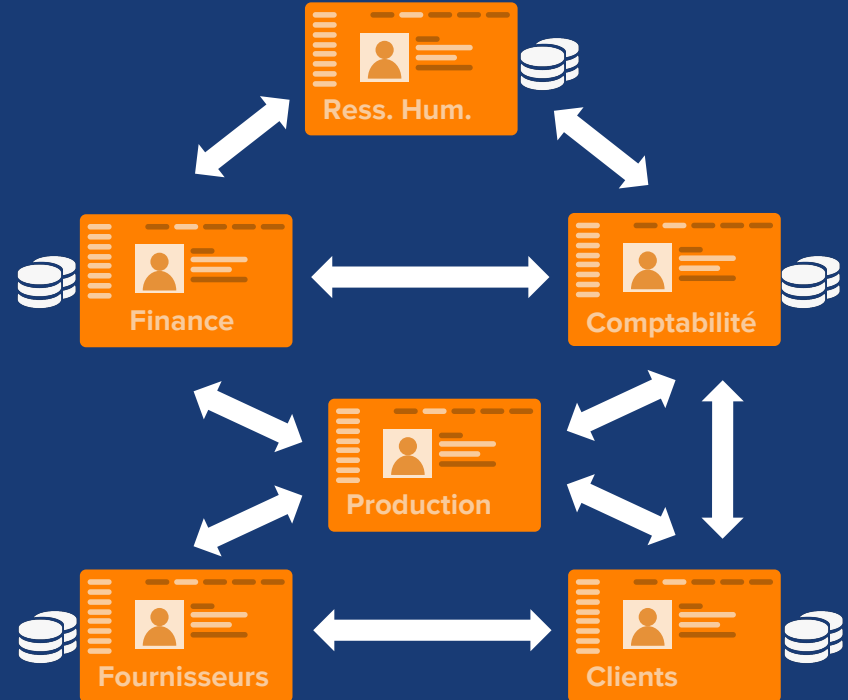
Les différents types de SI

ERP ou PGI

(Enterprise Resources Planning)
(Progiciel de Gestion Intégré)



Plusieurs Systèmes Interfacés entre eux



Les SI Techniques



Les métiers du SI



Le Directeur des SI

ou Directeur de l'Organisation et des SI
ou Directeur des SI et du Numérique

Il est en charge de tous l'ensemble des composants de systèmes d'information. Il veille à l'application de la stratégie numérique de l'entreprise, au bon fonctionnement des SI et au déploiement de celle-ci



Le Directeur de programmes

Un programme est un ensemble de projets de SI dans une même division de l'organisation. Son rôle est de :

- coordonner les différents projets entre eux,
- assurer les moyens pour leur réussite
- assurer leur suivi



Le Directeur de projet

Son rôle est de :

- coordonner les différents acteurs du projet issue du métier et de l'IT
- assurer les moyens pour la réussite
- assurer leur suivi (budget, planning...)
- assurer la livraison des développement, leur installation, l'assistance utilisateur, le support et la maintenance

Les métiers du SI (suite)



Business Analyst

Il est chargé de :

- Recueillir les besoins
- Spécifier les solutions fonctionnelles
- Suivi des développements
- Recette (tests) des développements
- Assister les utilisateurs
- Gérer les réclamations et les améliorations



Le développeur

Il est chargé de :

- Spécifier la solution technique
- Programmer la solution
- Mener les tests unitaires
- Corriger / Améliorer la solution



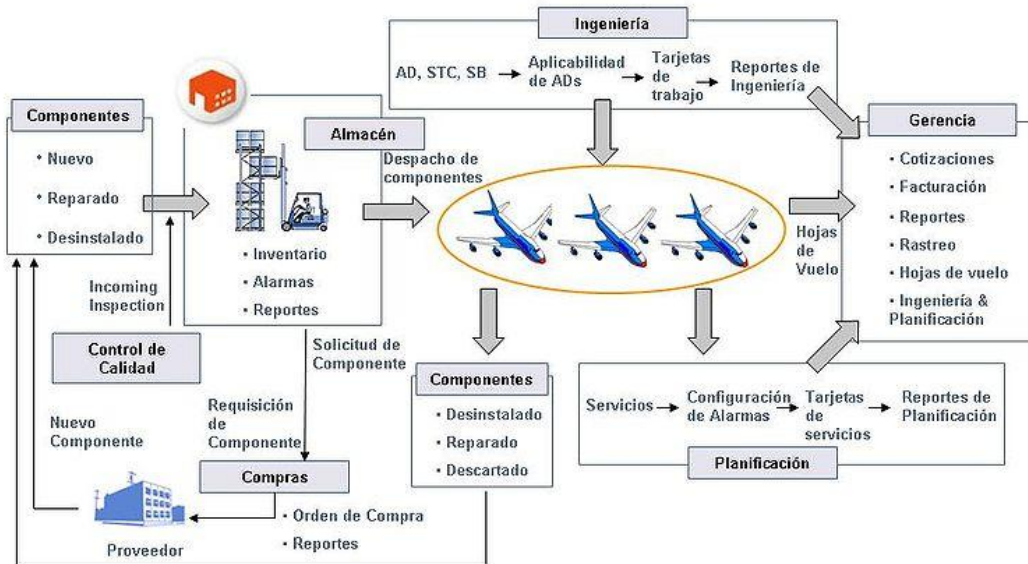
Architecte SI

Son rôle est de :

- définir les composants du système
- les interfacier
- coordonner les équipes logicielles avec les équipes matérielles

Architectures des SI

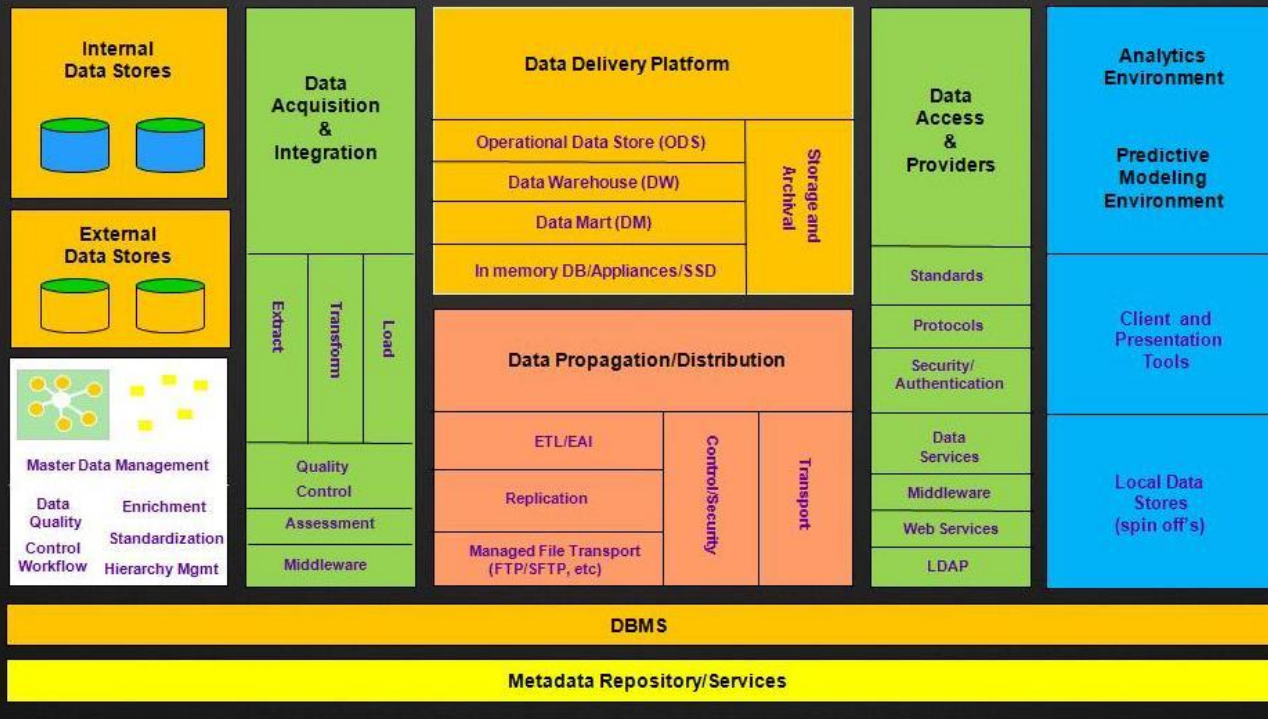
Architecture Métier



Copyright : Domaine public

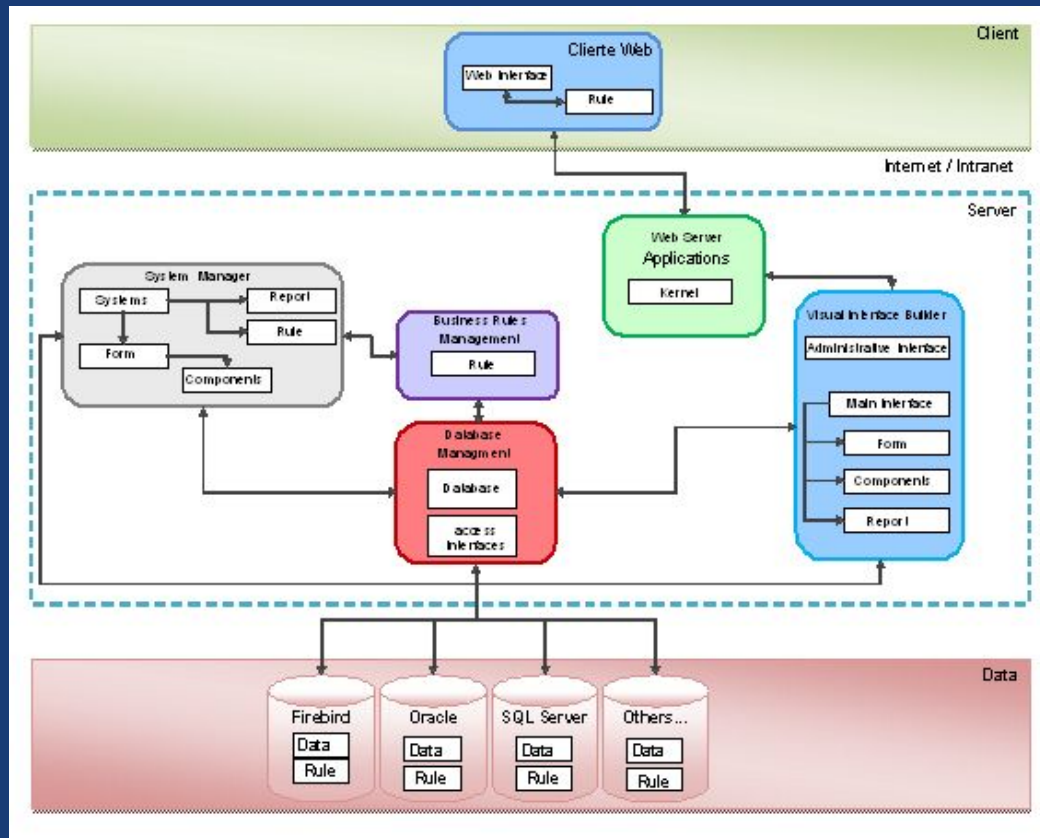
Architecture des informations

Data Architecture Reference Model



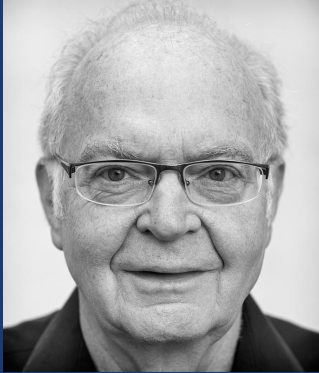
Copyright : Domaine public

Architecture logicielle



Démarche Créative

De l'idée au concret



'We have seen that computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty.'

'Nous avons vu que la programmation informatique est un art, car il s'applique les connaissances accumulées au monde, car il nécessite des compétences et de l'ingéniosité, et surtout parce qu'il produit des objets de la beauté.'

- Knuth, la programmation Informatique comme un art, CACM, 1974.
<http://www.paulgraham.com/knuth.html>

La pensée conceptuelle



Concevoir

En général, avoir une idée ferait suite à un besoin non adressé, un problème ou bien une frustration. Ce serait opportun d'analyser la source de ce problème pour bien le formuler et trouver l'idée d'une solution



Prototyper

Une fois le problème est bien formulé nous procédons à sa résolution grâce à la pensée algorithmique. Et grâce au code informatique, il est possible d'implémenter sa solution assez rapidement du moment que nous savons la "coder"!



Tester

Quand le prototype est prêt, on l'éprouve. Si la conception est bien réalisée, le test sera réussi. Et si ce n'est pas le cas, nous revenons à l'étape conception pour corriger et nous répétons la boucle jusqu'à finir par un prototype fonctionnel et staisfaisant

Pensée Algorithmique

Démarche pédagogique : La pensée algorithmique



Résoudre un problème

Dans notre système éducatif, grâce aux mathématiques, nous apprenons à formuler les problèmes sous forme d'équations. Par la suite, nous appliquons un "Algorithme" pour les résoudre.

D'où vient cet algorithme ?

C'est un schéma de résolution que des penseurs précurseurs ont mis en place lorsqu'ils se sont attaqués à ces équations.

La pensée algorithmique

C'est une approche séculaire et date depuis Socrate et a été décrite par Platon. Elle a été officialisée par "Al Khawarizmi" en décrivant la méthode de résolution d'une équation du 2nd degré.

En quoi cela consiste ?

D'abord, bien formuler le problème, ensuite, le diviser en sous-problèmes (bien formulés à leur tour), pour bien les analyser et détecter les schémas simple de résolution avant de finir par construire la l'algorithme solution en combinant et factorisant les différents schémas simples.

Cette approche est utilisé pour enseigner à la fois les bases du code informatique et aussi apprendre à devenir *créatifs* et *innovants* en apprenant à résoudre soi même les problèmes de la vie au quotidien.

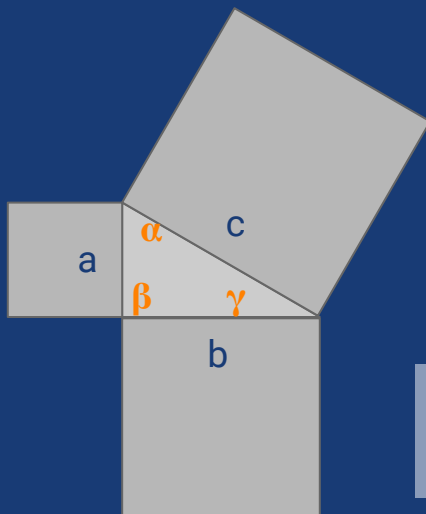


Formuler

Abstraction : *nom féminin*

- *Opération intellectuelle qui consiste à isoler par la pensée l'un des caractères de quelque chose et à le considérer indépendamment des autres caractères de l'objet.*

(Larousse)



$$a^2 + b^2 = c^2 \dots$$

$$\alpha + \beta + \gamma = 180^\circ \dots$$

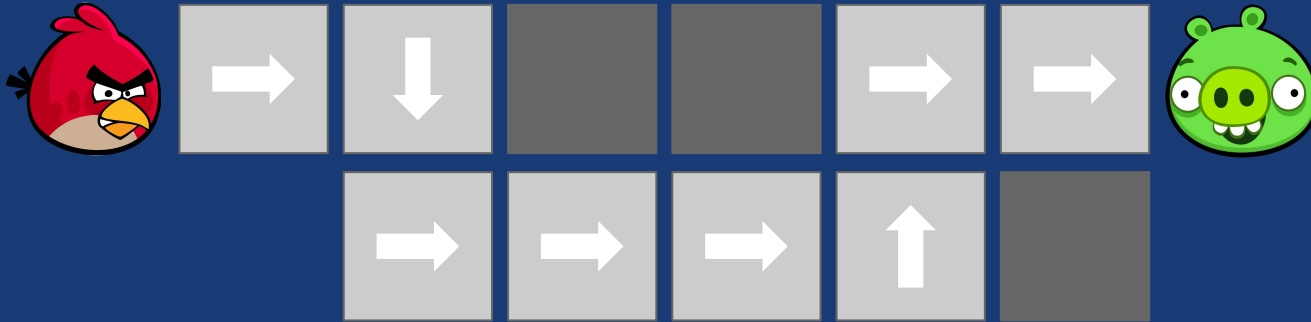
Et ainsi de suite

Les mathématiques nous aident à l'abstraction, mais pas à la résolution.

Séparer



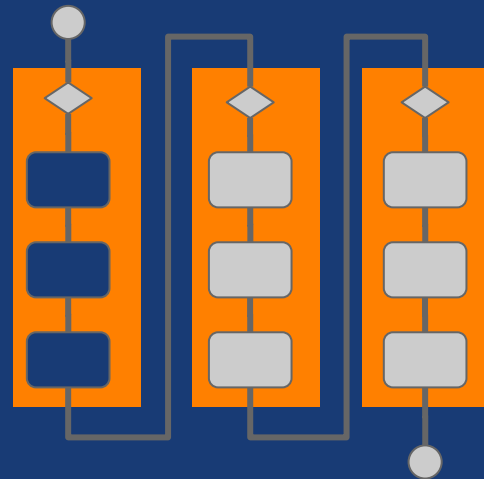
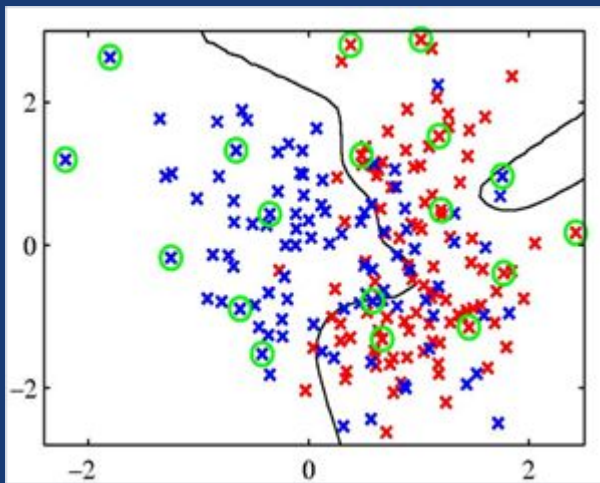
- Diviser les problèmes, données ou processus en sous-ensembles élémentaires faciles à résoudre et à gérer



Analyser



- Analyse des données et découverte des tendances ou bien des aspects réguliers dedans
- Découverte des modèles et des schémas répétitifs des tâches pour les factoriser





Résoudre

- Développement des instructions à exécuter pour chaque tâche élémentaire
 - Assemblage des différentes tâches
 - Factorisation et édition de l'algorithme final
-
- Cas concret :
 - Le raisonnement par la récurrence
 - Calcul du factoriel ou bien la somme des nombres ... etc
 - Exemple d'une seule instruction à exécuter par agent : Tri à bulles

Création d'un jeu

Au choix : Astéroïd, Geometry Dash, Formule 1...



<http://meo.codimeo.com>