Méthodologie

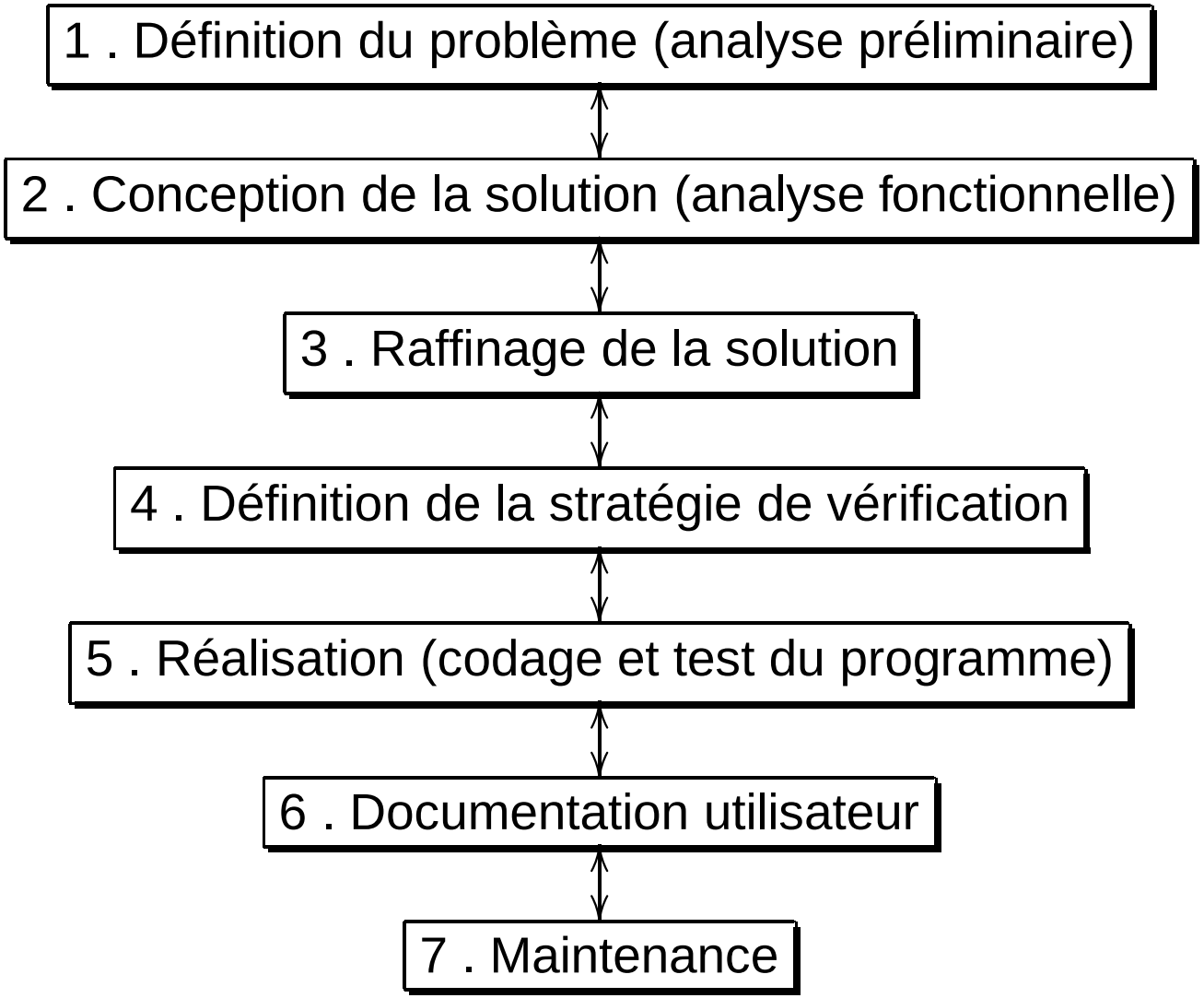
Florent Devin

Ecole Internationale des Sciences du Traitement de l'Information

Préambule

- Pas de méthode unique
- Nécessaire d'adopter une méthode
- Nécessaire d'adopter une seule méthode
- Permet de travailler en équipe
- Besoin de normaliser

Cycle de vie d'un logiciel



Développement d'un logiciel/projet

- Ne pas se jeter sur la machine !
- Le codage n'est *qu'une* étape
- Mauvais étude des étapes précédente $\Leftrightarrow$ Mauvais programme $\Leftrightarrow$ Mauvaise note :-(
- Si impossibilité dans une étape $\Leftrightarrow$ remise en cause de l'étape précédente
- Remise en cause de l'étape précédente $\Leftrightarrow$ remise en cause de l'étape précédente
- Changement dans une étape $\Leftrightarrow$ Propagation des changements aux sous étapes

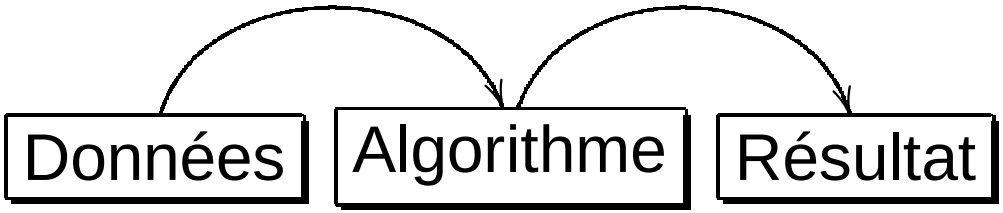
Analyse préliminaire

- Nécessaire d'avoir la compréhension totale d'un problème
- Une définition trop vague $\Leftrightarrow$ Développement difficile
- Les parties obscures doivent être clarifiées
- Définition correct du problème $\Leftrightarrow$ développement plus rapide
- Si problème
  - Revoir votre partie
  - Changer la façon d'aborder le problème

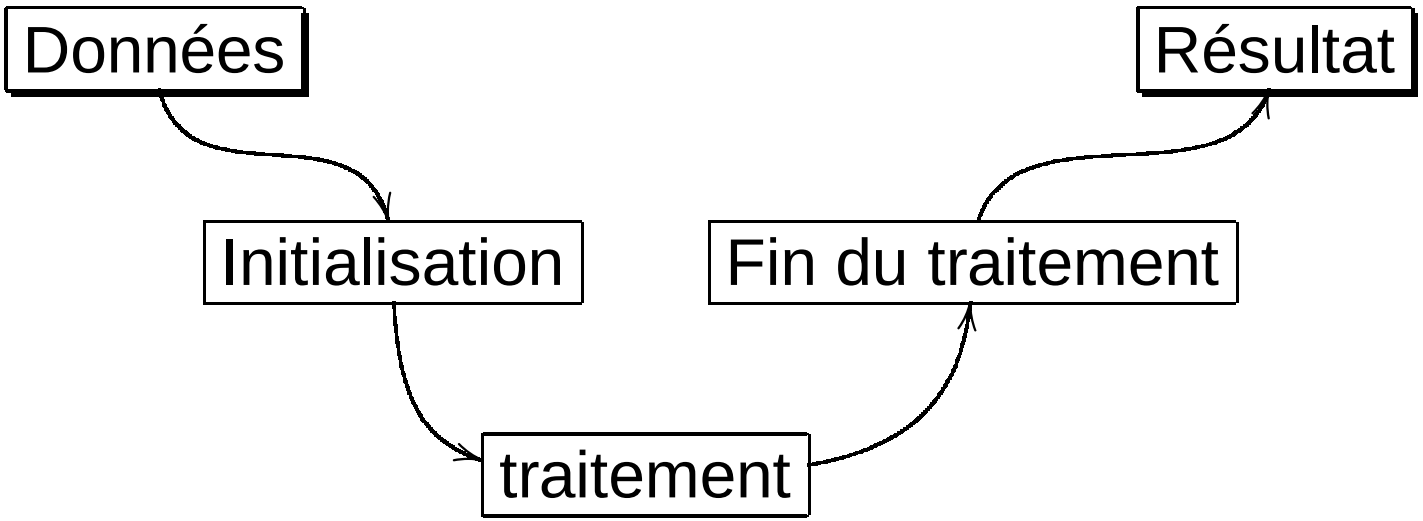
Analyse fonctionnelle

- Utilisation de la technique “diviser pour régner”
  - Analyser le problème
  - Décomposer en sous problèmes (plus simple à résoudre)
  - Décomposer les sous problèmes en sous problèmes
  - Représentation schématique des décompositions $\Leftrightarrow$ structure général du programme.

Exemple théorique



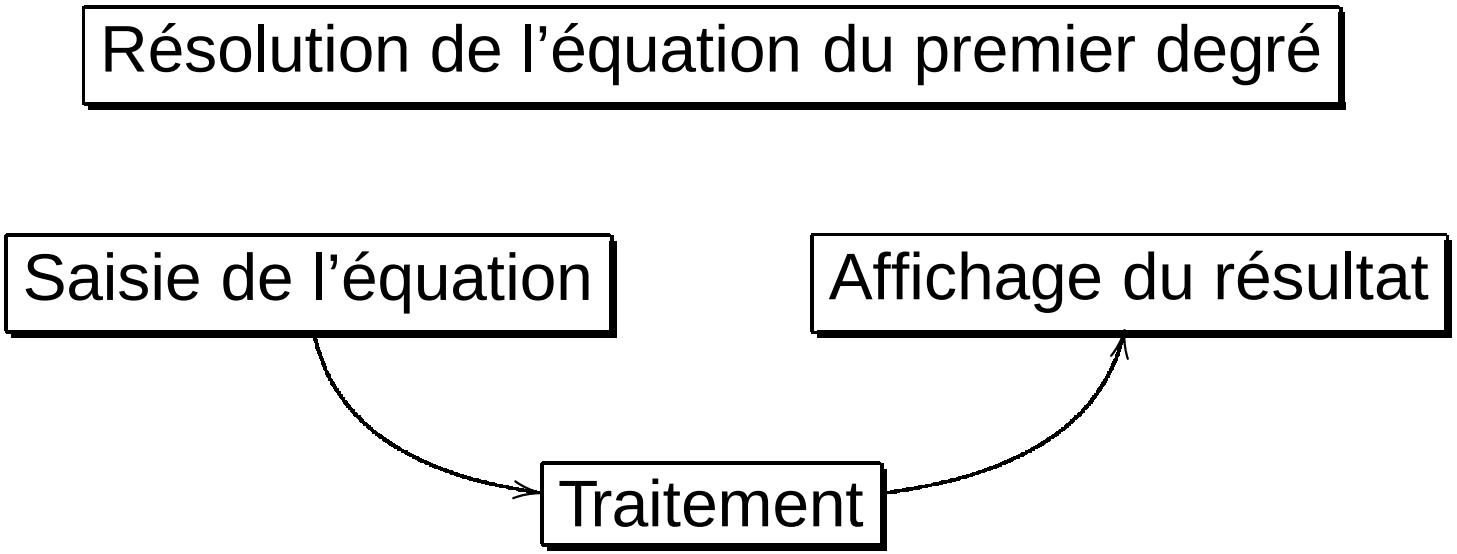
Exemple théorique



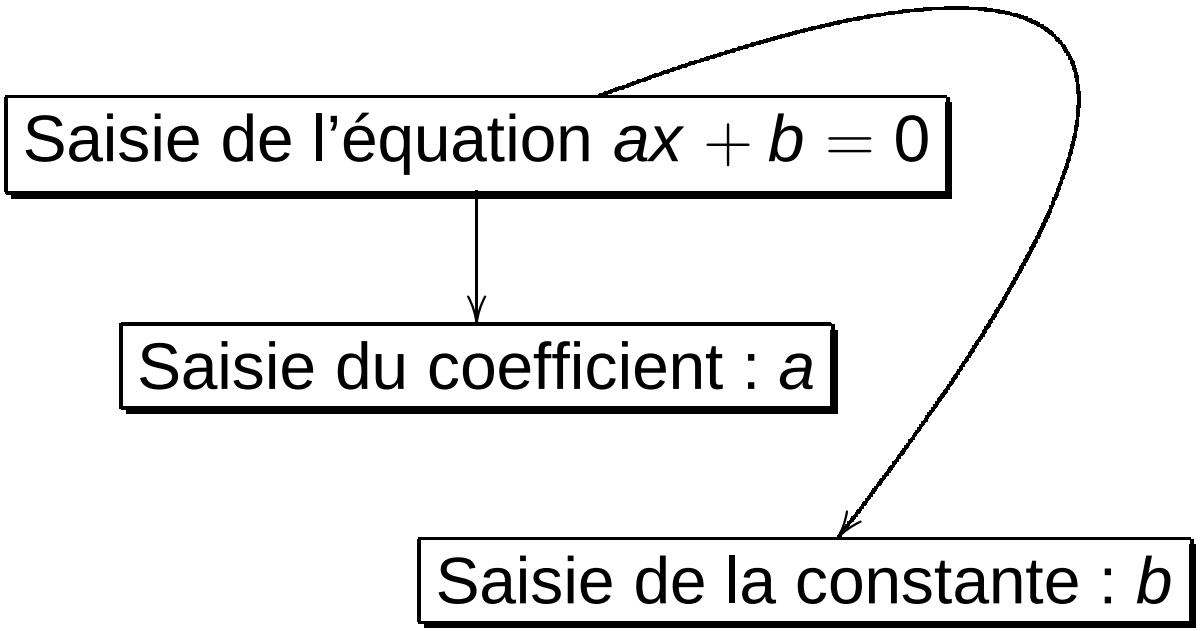
Exemple concret

Résolution de l'équation du premier degré

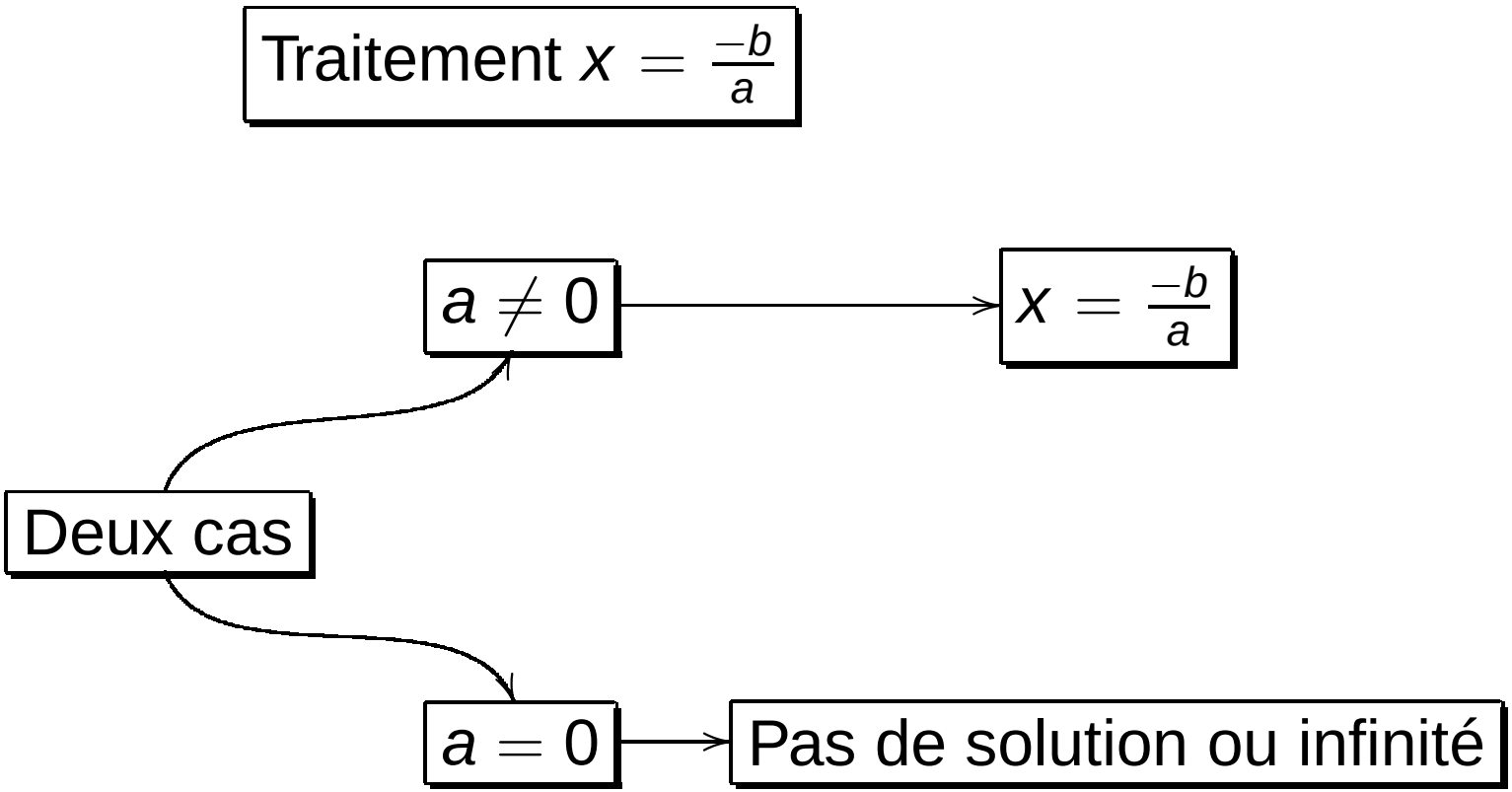
Exemple concret



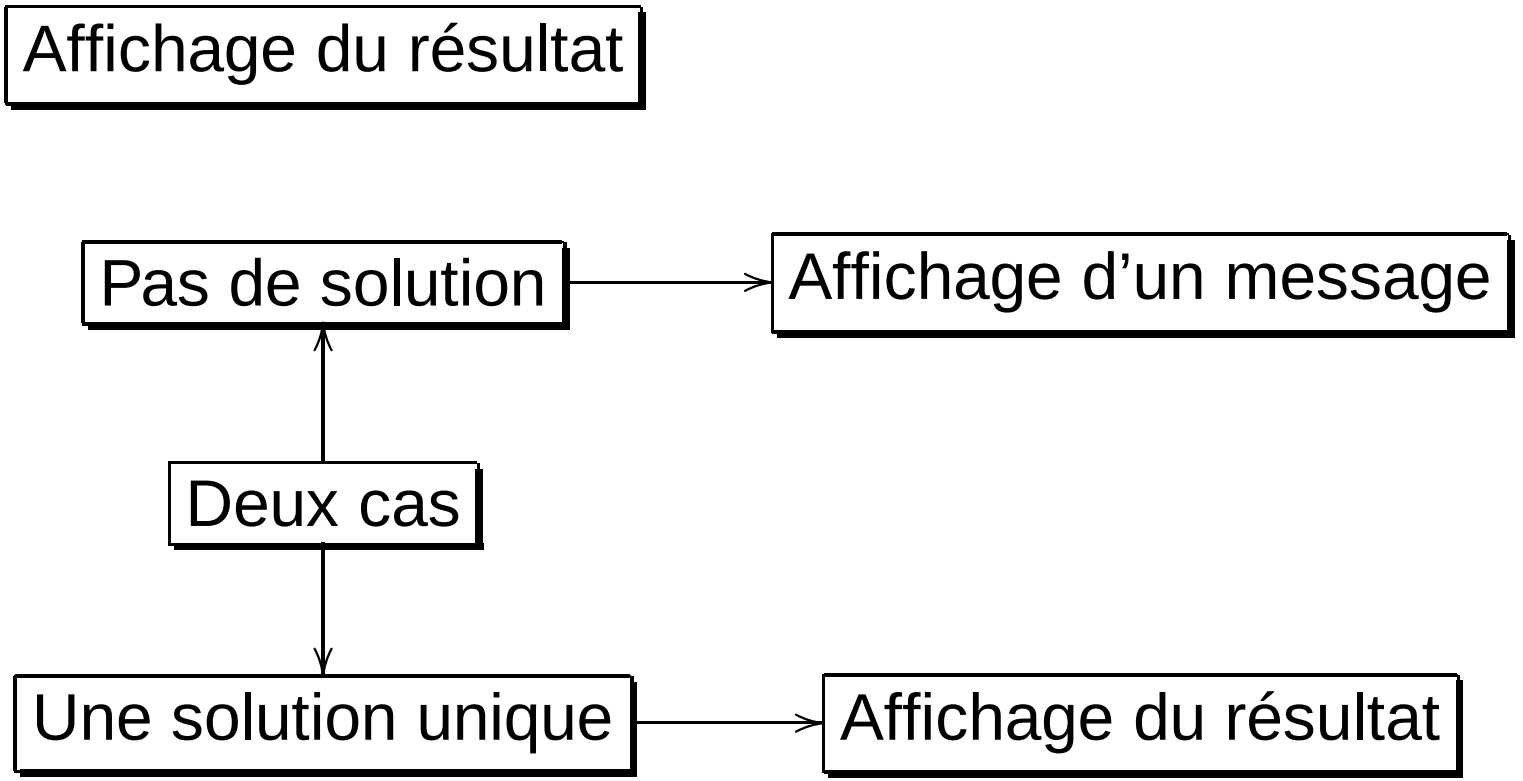
Exemple concret



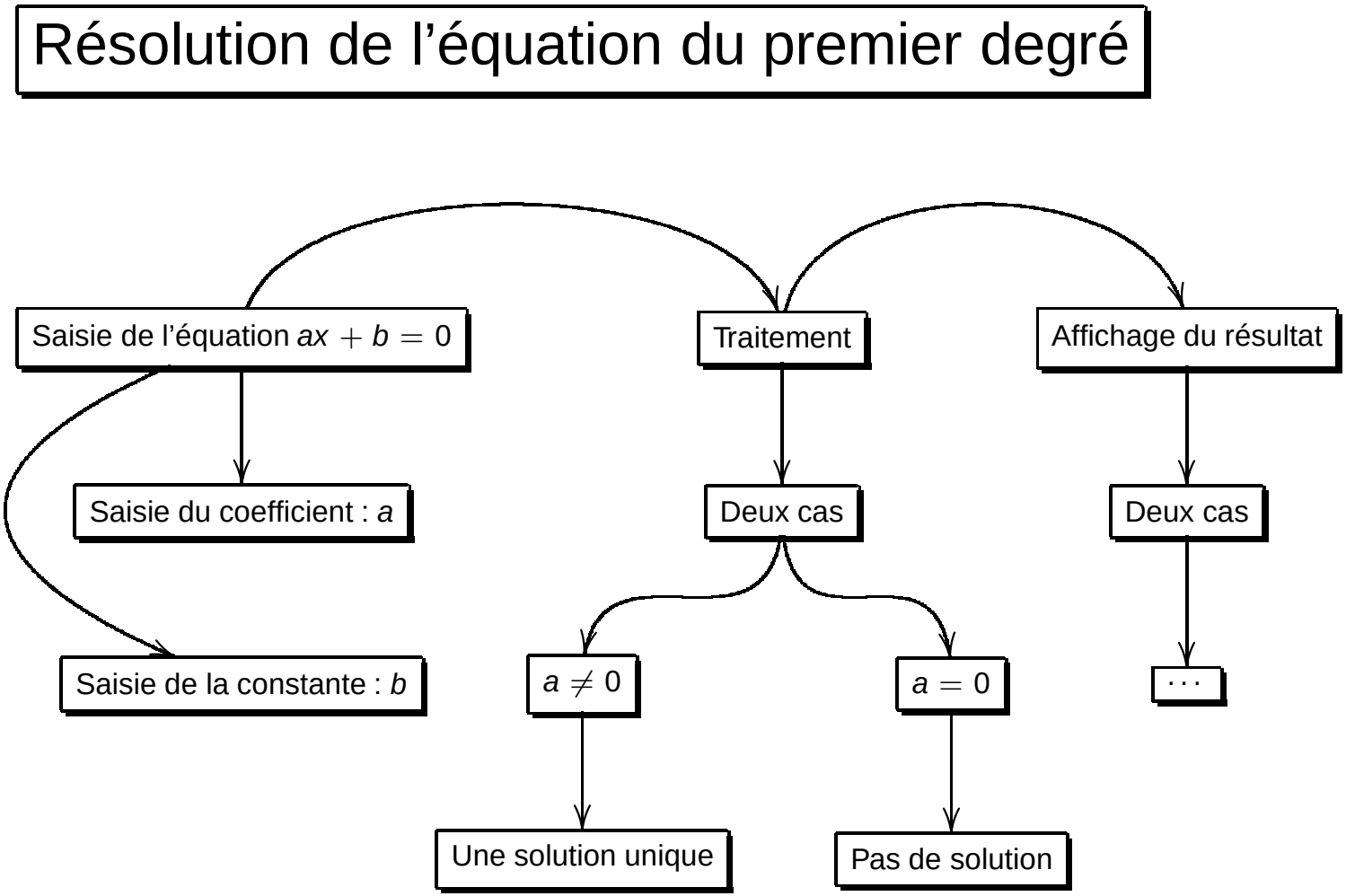
Exemple concret



Exemple concret



Exemple concret



Remarques

- Plusieurs façons de décomposer un problème
  - Saisie de l'équation et Traitement/affichage de la solution
- Selon le temps envisager plusieurs solutions
- Peut être bon de penser à factoriser des traitements
- Besoin de définir des interfaces pour transmettre des données d'un composant à un autre
- Décomposition optimale s'acquiert avec l'expérience !

Résumé

- Analyse fonctionnelle
  - les structures de données utilisées ;
  - la structure fonctionnelle du projet ;
  - éventuellement une planification ;
  - les fonctionnalités du projet.

Conception détaillée

- À partir du squelette obtenu de l'analyse fonctionnelle
- Détail de chacun des traitements
- Création d'algorithme pour chaque partie non triviale

Vérification

- Étape fondamentale
- Validation de l'ensemble des algorithmes trouvés
- Spécification explicite des interfaces des composants (entrée/sortie)
- Faisable sur un petit projet
- Difficile sur un gros projet $\Leftrightarrow$ plan de vérification obligatoire pendant le codage

Vérification descendante

- Commence par le programme principal
- Vérifie en descendant de les fonctions comme le ferait l'exécution
- Chaque étape doit être vérifiée
  - Validation de l'ensemble des possibilité à un instant t
  - Génération de l'ensemble des réponses possibles
  - Continue l'exécution avec chacune des réponses possible

Vérification ascendante

- Procédé inverse à la vérification descendante
- Pas incompatible
- Vérification ascendante
  - Validation individuelle de chacune des fonctions
  - Vérification de l'interaction entre les composants (correcte, faisable, sans risque, ...)
  - Permet de vérifier un module
- Utilisation des deux méthodes en fonction du contexte

Codage

- Utilise le travail précédent
- Qu'une étape parmi d'autre !
- Résolution de problème matériel uniquement
- Il ne doit plus y avoir de problème de conception

Coopération

- Nombre d'intervenant $\Leftrightarrow$ communication
- Adoption d'un style de programmation unique
  - À l'école c'est obligatoire
- Commenter correctement les codes
  - pour mieux expliquer ce que vous avez fait
  - pour vous souvenir de ce que vous aviez fait
  - mieux vaut taper les commentaires puis le code (pas l'inverse)

Documentation

- Normes de l'école $\Leftrightarrow$ documentation technique
- Réalisation d'une documentation utilisateur
- Réalisation d'une documentation d'installation
- Réalisation d'une documentation de type *README*

Maintenance

- Étape après la livraison
- Amélioration, ajout de fonctionnalités
- Correction de bug ne devrait pas en faire parti !