

Élaboration d'une calculatrice virtuelle

Soutenance du travail de maturité

Phuc-Thien Thomas Nguyen

Collège Rousseau

7 novembre 2014

- 1 Introduction (4 min)
 - Début, et plan (1 min)
 - Objectifs et méthodes de ce travail (3 min)

- 2 Contenu (11 min)
 - Quelques implémentations clés (8 min)
 - Nombres arbitrairement grands et exacts (3 min)
 - Expressions et analyseur syntaxique (5 min)
 - Test des implémentations (3 min)

- 3 Résultat/Conclusion (5 min)
 - Démonstration du produit final (3 min)
 - Bilan (2 min)



Objectifs et méthodes

Élaboration d'une calculatrice virtuelle



Objectifs et méthodes

Élaboration d'une calculatrice virtuelle qui devra être :

- Simple d'utilisation
- Originale
- Modulable/Organisée
- Fonctionnelle !



Objectifs et méthodes

Élaboration d'une calculatrice virtuelle qui devra être :

- Simple d'utilisation
- Originale
- Modulable/Organisée
- Fonctionnelle !

Ressources :

- Mathématiques : niveau du primaire au collège.
- Informatique : aisance dans la programmation.

→ Le plus important : ressources personnelles (concentration, autonomie, patience, persévérance).

Objectifs et méthodes

Élaboration d'une calculatrice virtuelle qui devra être :

- Simple d'utilisation
- Originale
- Modulable/Organisée
- Fonctionnelle !

Ressources :

- Mathématiques : niveau du primaire au collège.
- Informatique : aisance dans la programmation.

→ Le plus important : ressources personnelles (concentration, autonomie, patience, persévérance).

Langage : C++ (multi-plateforme, connu, orienté objet, efficace).



Nombres arbitrairement grands et exacts

Nombres arbitrairement grands et exacts

- Entiers : décomposition du nombre en chiffres, (40297 → [4] [0] [2] [9] [7]) : stockage dans des tableaux dynamiques (classe `vector` du C++).

Nombres arbitrairement grands et exacts

- Entiers : décomposition du nombre en chiffres, (40297 → [4] [0] [2] [9] [7]) : stockage dans des tableaux dynamiques (classe `vector` du C++).
- Nombres rationnels en valeurs exactes : stockage du numérateur et du dénominateur sous forme de deux nombres arbitrairement grands.

Nombres arbitrairement grands et exacts

- Entiers : décomposition du nombre en chiffres, (40297 \rightarrow [4] [0] [2] [9] [7]) : stockage dans des tableaux dynamiques (classe `vector` du C++).
- Nombres rationnels en valeurs exactes : stockage du numérateur et du dénominateur sous forme de deux nombres arbitrairement grands.
- Gestion des nombres complexes à coefficients rationnels.



Expressions et analyseur syntaxique

Expressions et analyseur syntaxique

- Idée : stocker les éléments d'une expression dans un tableau dynamique. Exemple : $2x + 3$ serait décomposé en $[2] [x] [+] [3]$.

Expressions et analyseur syntaxique

- Idée : stocker les éléments d'une expression dans un tableau dynamique. Exemple : $2x + 3$ serait décomposé en $[2] [x] [+] [3]$.
- Problème : on ne peut pas gérer de tableau dynamique contenant plusieurs types de variables.

Expressions et analyseur syntaxique

- Idée : stocker les éléments d'une expression dans un tableau dynamique. Exemple : $2x + 3$ serait décomposé en [2] [x] [+] [3].
- Problème : on ne peut pas gérer de tableau dynamique contenant plusieurs types de variables.
- Solution : collection hétérogène (classe de base + héritage + tableau dynamique d'objets de base + polymorphisme).

Expressions et analyseur syntaxique

- Idée : stocker les éléments d'une expression dans un tableau dynamique. Exemple : $2x + 3$ serait décomposé en [2] [x] [+] [3].
- Problème : on ne peut pas gérer de tableau dynamique contenant plusieurs types de variables.
- Solution : collection hétérogène (classe de base + héritage + tableau dynamique d'objets de base + polymorphisme).
- Règles pour l'analyse syntaxique (validité de la saisie, interprétation) et la gestion de parenthèses (récursivité).



Test des implémentations

Test des implémentations

- Exécuter les implémentations, et vérification des résultats.

Test des implémentations

- Exécuter les implémentations, et vérification des résultats.
- Distribution du programme à l'entourage pour qu'il soit testé et commenté.

Démonstration de PolyCalc

Après la conception d'Algèbra, est venue celle de PolyCalc...
Bibliothèque utilisée : FLTK : rapide, facile, très légère, compilation statique. Elle inclut OpenGL pour la représentation graphique.



Bilan

Bilan

- Buts du travail atteints : calculatrice fonctionnelle, facile à utiliser, proposant des fonctions originales, facilement améliorable.

Bilan

- Buts du travail atteints : calculatrice fonctionnelle, facile à utiliser, proposant des fonctions originales, facilement améliorable.
- Pour aller plus loin : améliorations possibles :
 - Gérer des fonctions non polynomiales.
 - Autres bases.
 - Optimisations.
 - Gérer les systèmes Android.

Bilan

- Buts du travail atteints : calculatrice fonctionnelle, facile à utiliser, proposant des fonctions originales, facilement améliorable.
- Pour aller plus loin : améliorations possibles :
 - Gérer des fonctions non polynomiales.
 - Autres bases.
 - Optimisations.
 - Gérer les systèmes Android.
- Expérience très satisfaisante et enrichissante : mise en pratique de mes compétences informatique, base pour un projet qui pourra être amélioré dans le futur.

- Fin -

Merci pour votre attention.