

- [contenu](#)
- [menu](#)
- [navigation](#)
- [pied de page](#)

Projet : Calculatrice à notation polonaise inverse

[défilement haut](#)

- [Exercice](#)
- [Spécifications](#)
- [Notation Polonaise Inverse](#)

[défilement bas](#)



Masquer le plan



Afficher le plan

Spécifications

Attention

Les spécifications présentées dans ce document présentent le fonctionnement de la calculatrice demandé. Vous pouvez les personnaliser à condition de ne pas diminuer la complexité du projet. En cas de doutes sur certaines adaptations, demandez l'avis de votre chargé de TP.

Ces spécifications vous laissent volontairement des choix de conception, tant fonctionnels que conceptuels et technologiques.

Quels que soit les choix et adaptations que vous ferez, vous prendrez garde de les exposer et justifier dans le document rendu avec le projet.

Type de données

La calculatrice peut manier différents types de constante :

- Entière : suite de chiffres
- Rationnelle : deux entiers séparés par le caractère '/'
- Réelle : suite de chiffres pouvant accepter une virgule

- Complexe : constituée de deux constantes (Entière, Rationnelle ou séparées par le caractère '\$')
- Expression : expression entourée du caractère ' ' à évaluer ultérieurement

Gestion des données

- Les constantes sont stockées dans une pile au fur et à mesure de l'exécution
- Les X derniers éléments de la pile sont affichés à l'écran (où X est paramétrable)

Évaluation d'une expression

Une expression est composée d'opérateurs et de constantes séparés par des espaces. Une expression est évaluée lorsque l'utilisateur appuie sur la **ENTRÉE**. Dans le cas d'une constante ou d'une expression bornée par le caractère ' ' et stockée dans la pile) l'évaluation s'opère à l'aide de l'opérateur **EVAL**.

- Les éléments constituant une expression sont évalués un par un, de la gauche vers la droite
- L'évaluation d'une constante provoque son empilement
- L'évaluation d'un opérateur binaire provoque le dépilement de deux dernières constantes de la pile avec lesquels sera effectuée l'opération. Le résultat de l'opération est empilé. L'évaluation d'un opérateur unaire fonctionne de la même façon en ne dépilant qu'une unique constante.

Si la pile ne contient pas assez de constantes pour évaluer un opérateur, l'opérateur est laissé intact et la calculatrice indique une erreur. Les éléments de l'expression en cours n'ayant pas encore pu être évalués (dont l'opérateur) sont laissés sur la ligne de commandes pour correction.

- Lorsqu'un opérateur différent de l'opérateur **EVAL** dépile une constante ou une expression, l'opérateur ainsi qu'une éventuelle seconde constante sont laissés à la constante ou expression.
- L'évaluation d'une constante ou d'une expression se réalise comme toute autre expression. La constante est dépilée, l'évaluation revient à entrer le caractère ' ' de la constante ou de l'expression sur la ligne de commandes.

Afin d'évaluer une expression, la calculatrice utilisera des modes paramétrables par l'utilisateur :

- Type de constante (entière, rationnelle ou réelle)
- Unité des degrés (radian ou degré)
- Utilisation des complexes (oui ou non)

Remarque

Les modes « *type de constante* » et « *complexe* » sont présents a de simplifier le projet. Plusieurs fonctionnements sont envisageables

- Utiliser ce mode à l'évaluation des constantes pour ne créer q des constantes du mode choisi.
- Utiliser ce mode uniquement à l'évaluation d'un opérateur. I constantes sont alors converties dans le mode sélectionné. (prendra garde d'avertir l'utilisateur lorsqu'une partie du conte est perdue par la conversion.
- Ne pas utiliser ce mode, le type de constante le plus riche est util par défaut pour résoudre chaque opérateur.

Quel que soit le fonctionnement choisi, la pile doit avoir la possibilité contenir des constantes de type différent (que ce soit par changement de mode en cours d'utilisation ou en reconnaiss différentes entrées de l'utilisateur).

Une attention particulière sera portée à la relation entre les constant rationnelles et réelles. Une constante rationnelle est par défaut pl riche qu'une constante réelle. Par exemple la constante $1/3$ contie plus d'information que la constante 0,3333.

Par défaut, un opérateur retourne une constante de même type que mode sélectionné. Certaines exceptions sont possibles à condition les justifier dans le document rendu (l'opérateur MEAN retourne u constante réelle ou rationnelle, les opérateurs trigonométriqu peuvent retourner des constantes réelles quel que soit le type constante d'entrée, etc.).

Interface

- La calculatrice possède un afficheur et un clavier
- L'affichage du clavier est une option paramétrable par l'utilisateur
- La calculatrice peut être utilisée à la souris ou directement au clavier
- Toutes les options de paramétrages sont définies à travers une i simple et agréable à utiliser.

Sauvegarde du contexte

Au démarrage de l'application, l'état de la pile et les paramétrage calculatrice lors de la dernière exécution sont récupérés.

Fonctions annuler et rétablir

La calculatrice dispose des fonctions « *annuler* » et « *rétablir* ». Ces f peuvent être appelées par un menu ou par les raccourcis Ctrl-Z et Ctrl-Y.

Opérateurs binaires

Les types entre parenthèses sont les types disponibles pour l'utilisation de l'opérateur :

- $+$, $-$, $*$, $/$ (entier, rationnel, réel, complexe)
- POW : puissance (entier, rationnel, réel)
- MOD : modulo (entier)
- SIGN : inversion du signe (entier, rationnel, réel, complexe)

Opérateurs unaires

Les types entre parenthèses sont les types disponibles pour l'utilisation de l'opérateur :

- SIN : sinus (entier, rationnel, réel)
- COS : cosinus (entier, rationnel, réel)
- TAN : tangente (entier, rationnel, réel)
- SINH : sinus hyperbolique (entier, rationnel, réel)
- COSH : cosinus hyperbolique (entier, rationnel, réel)
- TANH : tangente hyperbolique (entier, rationnel, réel)
- LN : logarithme népérien (entier, rationnel, réel)
- LOG : logarithme décimal (entier, rationnel, réel)
- INV : inverse (entier, rationnel, réel)
- SQRT : racine carrée (entier, rationnel, réel)
- SQR : fonction carré (entier, rationnel, réel, complexe)
- CUBE : fonction cube (entier, rationnel, réel, complexe)
- ! : factorielle (entier)
- EVAL : évaluation d'une expression (Expression)

Opérations sur la pile

La pile peut être manipulée à l'aide des opérateurs suivants.

Opérateur binaire :

- SWAP : inversion de l'élément numéro x et l'élément numéro y de la pile (x et y sont les arguments de l'opérateur) (entier)

Opérateurs unaires :

- SUM : somme des x premiers éléments de la pile (où x est l'argument) (entier)

- MEAN : moyenne des x premiers éléments de la pile (où x est l'arg (entier))

Opérations sans arguments :

- CLEAR (ou Ctrl - C) : vider la pile
- DUP (ou Entrée sans aucune expression sur la ligne) : duplique le 1 élément de la pile
- DROP (ou Retour en arrière sans aucune expression sur la ligne) : supprime le premier élément de la pile

Options

Pour compléter votre projet, vous choisirez au minimum une des fonctionnalités suivantes :

- Commande utilisateur : vous commencerez par enrichir les expressions avec la possibilité d'ajouter une ou deux variables (formatage prédéfini à inclure dans l'expression). Ces expressions s'exécuteront comme un opérateur (un opérateur binaire). Vous ajouterez ensuite un nouvel opérateur binaire prenant une chaîne de caractères et une expression dynamique en argument. L'utilisation de la chaîne appellera l'expression dynamique enregistrée. Vous ferez en sorte que les réglages utilisateurs soient persistants d'une exécution à l'autre de la calculatrice.
- Onglets : vous placerez votre pile dans une collection STL de votre choix. La commande Ctrl-N ouvrira un nouvel onglet avec une pile vide. Cette pile sera ajoutée à la collection. La commande Ctrl - T ouvrira un nouvel onglet en dupliquant la pile courante. La commande Ctrl-W supprimera l'onglet courant.
- Log : vous implémenterez une classe LogMessage, contenant un message et un degré d'importance du message de log. Une seconde classe LogHandler manipulera des LogMessage pour écrire les messages de log sur un fichier dans un fichier dédié.