Calculatrice_CLI_version_2

1 - ALGORITHMES

ALGORITHME CalculatriceAST DEBUT // INITIALISATION BIN_OPS ← mapping types AST → fonctions binaires UNARY_OPS ← mapping types AST → fonctions unaires CONSTANTS ← mapping noms → valeurs constantes ans ← None // Mémoire dernier résultat

```
// MODE LIGNE DE COMMANDE
SI arguments > 1 ALORS
    expression ← concaténer arguments[1:]
    résultat ← safe_eval(expression)
    AFFICHER résultat
SINON
    // MODE INTERACTIF REPL
    AFFICHER instructions
    TANT QUE VRAI FAIRE
        SAISIR input ← input(">>> ")
        input ← TRIM(input)

SI input VIDE ALORS CONTINUER

SELON input
            CAS ("q", "quit", ":exit"): QUITTER
            CAS ":ans": AFFICHER valeur ans
            AUTRE:
                résultat ← safe_eval(input)
                ans ← résultat // Mémorisation
                AFFICHER "= {résultat}"
        FIN SELON
    FIN TANT QUE
FIN SI
```

FIN

// SOUS-ALGORITHME safe_eval ALGORITHME safe_eval(expression) DEBUT // VALIDATION SÉCURITÉ allowed_chars ← "0123456789+-*/.()% pi e ans " SI ∃ caractère ∉ allowed_chars ALORS LANCER ERREUR "Caractères non autorisés" FIN SI

```
// PARSING SYNTAXIQUE
arbre ← ast.parse(expression, mode='eval')
```

```
// ÉVALUATION RÉCURSIVE
RETOURNER eval_ast(arbre)
```

FIN

// SOUS-ALGORITHME eval_ast (RÉCURSIF) ALGORITHME eval_ast(noeud) DEBUT SELON type(noeud) CAS ast.Expression: RETOURNER eval_ast(noeud.body)

```
CAS ast.BinOp:
        gauche ← eval_ast(noeud.left)
        droite ← eval_ast(noeud.right)
        type_op ← type(noeud.op)

SI type_op ∉ BIN_OPS ALORS
            LANCER ERREUR "Opérateur non supporté"
        FIN SI

RETOURNER BIN_OPS[type_op](gauche, droite)

CAS ast.UnaryOp:
        opérande ← eval_ast(noeud.operand)
        type_op ← type(noeud.op)

SI type_op ∉ UNARY_OPS ALORS
            LANCER ERREUR "Opérateur unaire non supporté"
        FIN SI

RETOURNER UNARY_OPS type_op

CAS ast.Constant:
        SI noeud.value ∈ (int, float) ALORS
            RETOURNER noeud.value
        SINON
            LANCER ERREUR "Constante non numérique"
        FIN SI

CAS ast.Name:
        SI noeud.id ∈ CONSTANTS ALORS
            RETOURNER CONSTANTS[noeud.id]
        SINON SI noeud.id = "ans" ET ans ≠ None ALORS
            RETOURNER ans
        SINON
            LANCER ERREUR "Variable inconnue"
        FIN SI

AUTRE:
        LANCER ERREUR "Type de noeud non supporté"
FIN SELON
```

FIN