LO21 : Rapport de projet
Calculatrice à notation polonaise inversée

Agathe Oddon
Jean-Michel Tozzini

Printemps 2012

Introduction

Dans le cadre de notre UV LO21, nous devions réaliser la conception puis implémenter en C++ une calculatrice à notation polonaise inversée.

Le rendu final du projet est composé du présent rapport, du code du programme ainsi que de sa documentation Doxygen.

Table des matières

1 Conception **2**
- 1.1 Diagramme de classes . . . 2
  - 1.1.1 Types de données . . . 3
  - 1.1.2 Sauvegarde des données . . . 3
- 1.2 Diagrammes de séquences . . . 4

2 Implémentation **5**
- 2.1 Paramètres utilisateur . . . 5
- 2.2 Variable utilisateur . . . 5
- 2.3 Sauvegarde et restauration de contexte . . . 6
- 2.4 Historique . . . 6
- 2.5 Exceptions . . . 6
- 2.6 Méthodes de Constante et ses classes filles . . . 6

1 Conception

1.1 Diagramme de classes

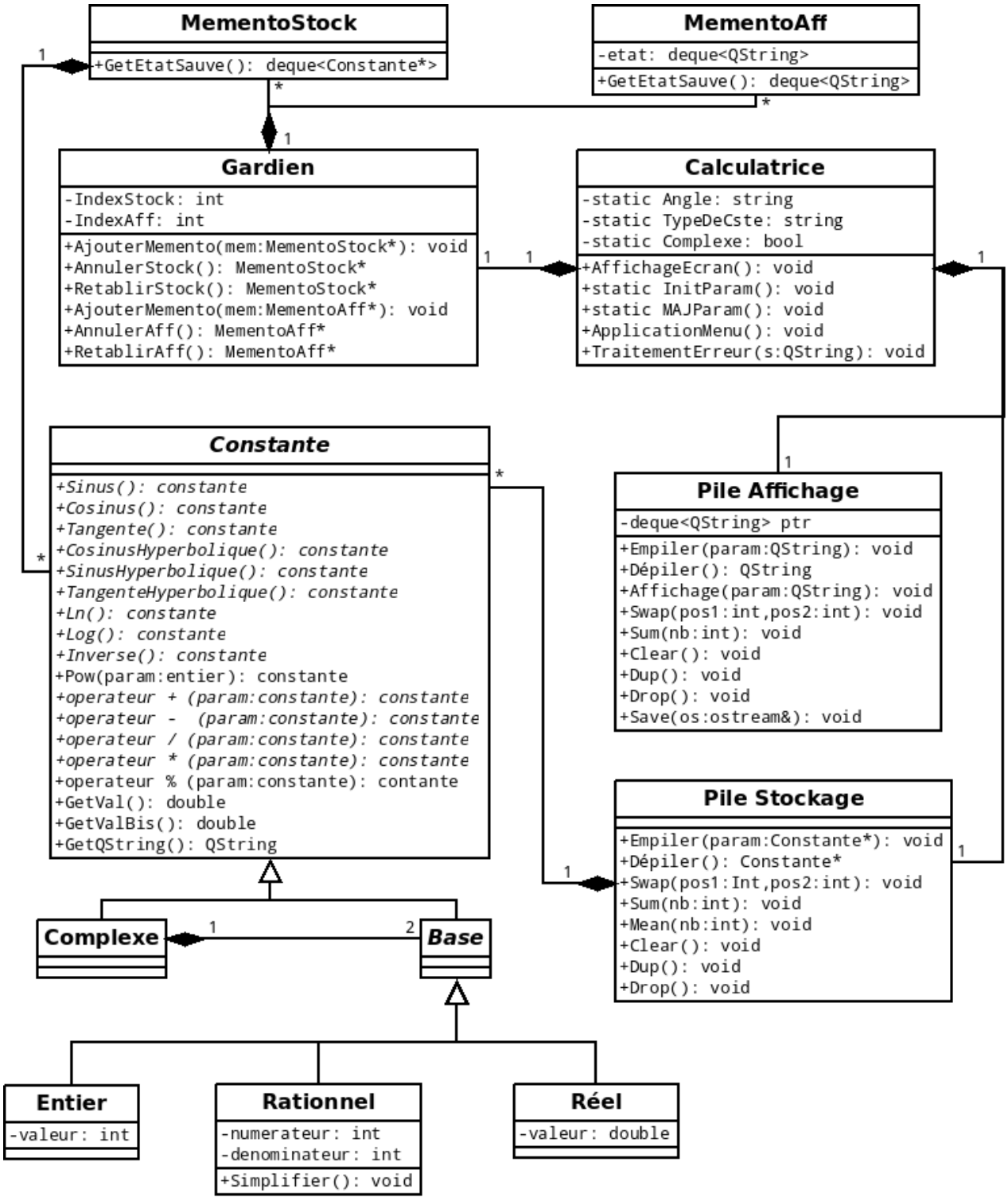

FIGURE 1 – Diagramme de classe de la Calculatrice à notation polonaise inversée

1.1.1 Types de données

Nous avons choisi de représenter les données manipulées par la calculatrice comme des objets de trois classes : Entier, Réel, Rationnel, Complexe et Expression.

La classe complexe est composée de deux objets de type Base, classe abstraite de laquelle dérivent Entier, Réel et Rationnel. Cela permet d'obtenir des complexes composés de deux attributs de types différents.

Les classes Base, Complexe et Expression dérivent de la classe abstraite et exclusive Constante.

L'utilisation de la classe mère abstraite Constante nous permet à la fois d'empiler des objets de type pointeur sur Constante, et de définir des méthodes polymorphes pour les opérateurs.

1.1.2 Sauvegarde des données

→ Pourquoi deux piles ?

1.2 Diagrammes de séquences

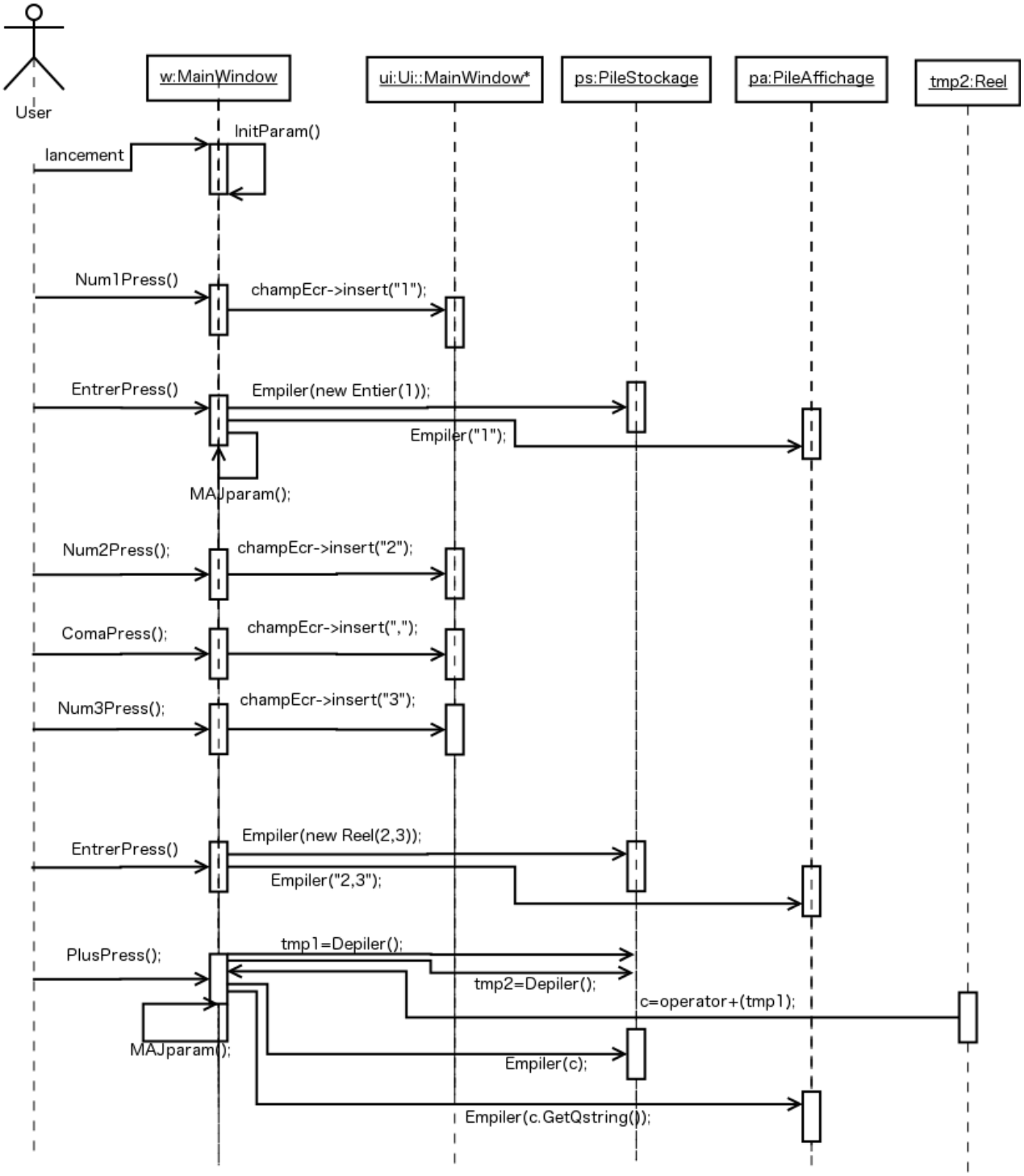

FIGURE 2 – Diagramme de séquence décrivant le lancement de la fenêtre, la saisie de "1", "2,3" , l'appui sur le bouton "+" par l'utilisateur, puis l'addition des deux valeurs.

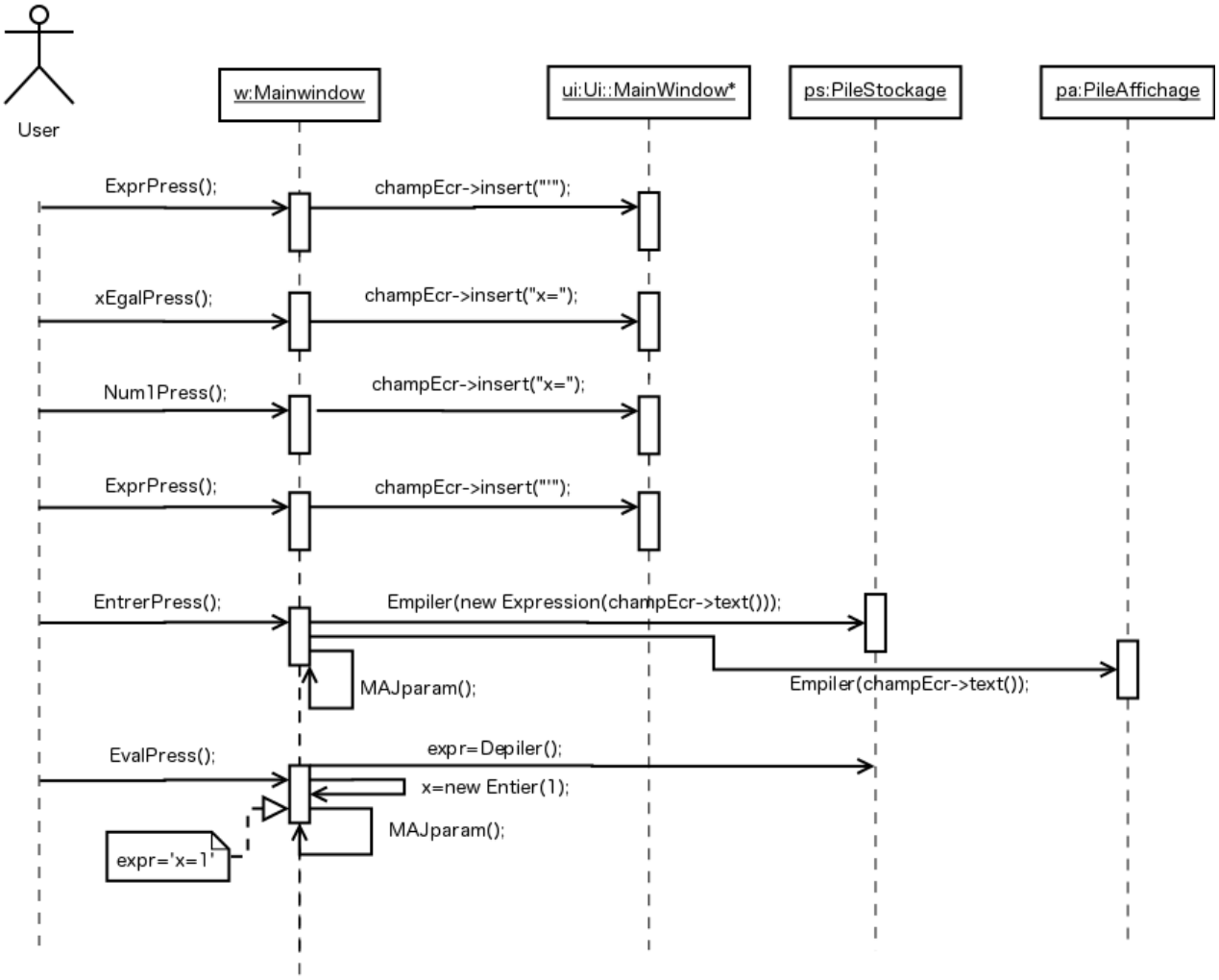

FIGURE 3 – Diagramme de séquence décrivant le lancement de la fenêtre, la saisie de 'x=1', puis l'évaluation de l'expression et la sauvegarde de la valeur 1 dans la variable x.

2 Implémentation

2.1 Paramètres utilisateur

L'utilisateur dispose d'un menu "Paramètres" lui permettant :
- De changer le type de constante - **EST-CE QUE CA ON LE SUPPRIME OU PAS ?**
- D'activer ou désactiver le clavier
- D'activer ou désactiver le mode complexe
- De choisir l'unité d'angle pour les fonctions cos, sin, tan, cosh, sinh et tanh

2.2 Variable utilisateur

L'initialisation de la variable utilisateur x se fait par l'intermédiaire d'une expression de la forme 'x=valeur'. Après évaluation (bouton "Eval"), cette expression initialise la variable x avec la valeur spécifiée, qui peut-être de type Entier, Rationnel, Réel ou Complexe. Ce comportement est décrit dans la figure 3.

La variable x sera ensuite accessible en tapant directement x ou par l'intermédiaire du bouton "x".

2.3 Sauvegarde et restauration de contexte

A l'ouverture du programme, une méthode est lancée permettant la restauration du contexte de la Calculatrice : il s'agit des piles, des valeurs des menu et de la valeur de la variable utilisateur si elle a été initialisée. Cette méthode interroge un fichier "param.txt" contenant les données du contexte. Si il s'agit de la première ouverture du programme, le fichier est initialisé avec des valeurs par défaut.

A chaque entrée de l'utilisateur ou modification des paramètres, une méthode est lancée pour mettre à jour le ficher.

2.4 Historique

2.5 Exceptions

2.6 Méthodes de Constante et ses classes filles

Pour les opérateurs utilisables avec plusieurs types de données, par exemple l'opérateur + peut-être utilisés avec les types Entier, Réel, Rationnel, Complexe et Expression, nous avons d'abord défini une méthode de Constante permettant d'appeler la bonne méthode de ses classes filles.

Toutes les méthodes ont ensuite dues être implémentées pour tous les types de données. Nous avons fait le choix de toujours renvoyer le type le plus riche. Par exemple, l'addition d'un entier et d'un réel renverra toujours un réel, de même la soustraction d'un rationnel et d'un complexe renverra un complexe. [1] Les méthodes qui n'acceptait pas le type de données, par exemple l'opérateur mod (modulo) n'accepte que les entiers, renvoient des exceptions.

1. Pour les opérations entre les types Rationnel et Réel, nous avons choisi de toujours renvoyer un Réel, méme si il ne s'agit pas du type le plus riche, du fait que nous ne pouvons pas convertir un Réel en Rationnel sans perdre d'information sur celui-ci.