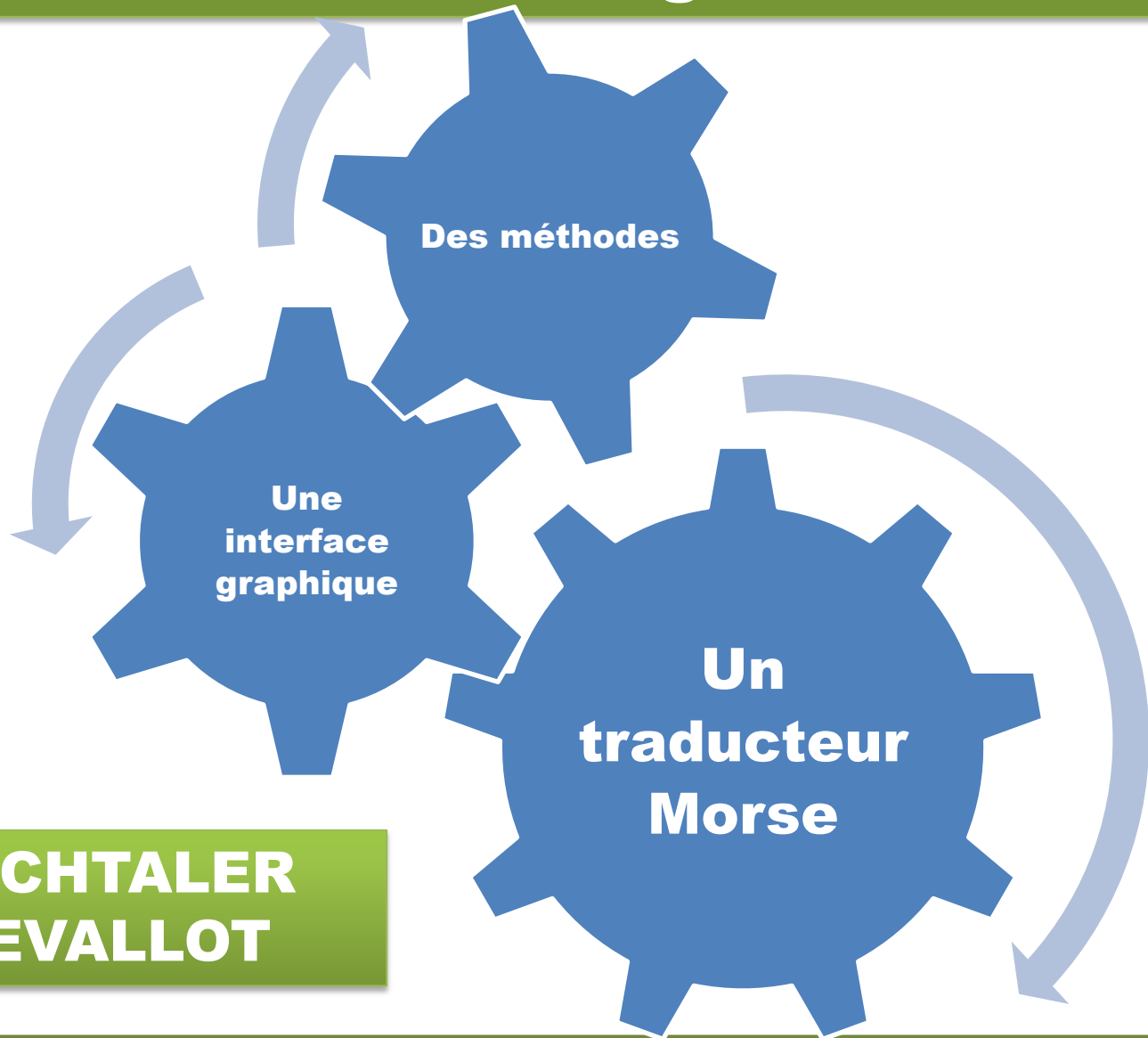


# Projet JAVA

## LPSIL Génie Logiciel



**Kévin MASCHTALER**  
**Pascal CHEVALLOT**

# SOMMAIRE

1. Analyse du besoin

2. La classe Morse

3. L'interface graphique

4. Feedback

# 1. ANALYSE DU BESOIN

Traduire un fichier texte en un fichier Morse et inversement :

1. Décomposition de la traduction :

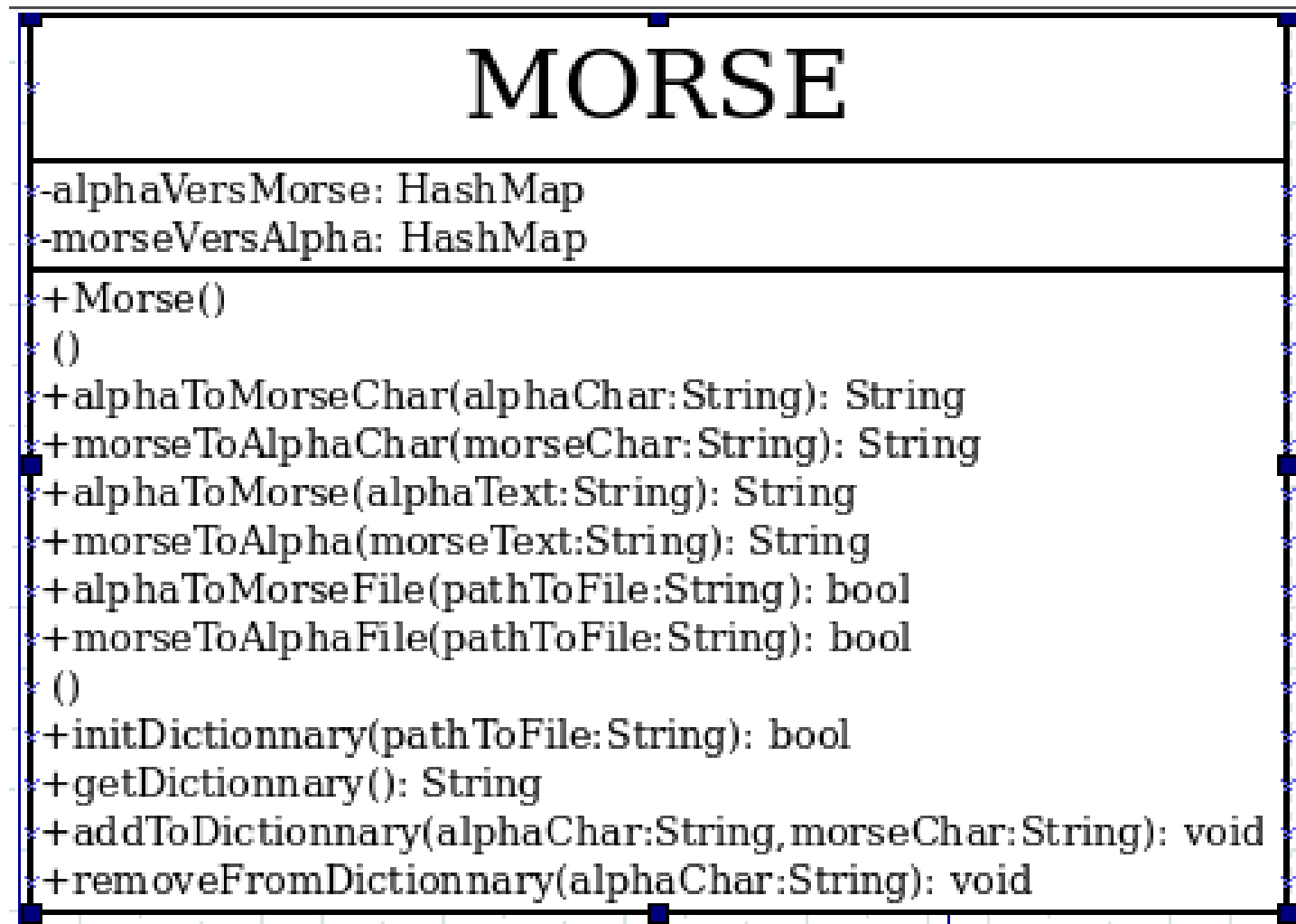
Texte > Ligne > Caractère > Dictionnaire

2. Lecture / Ecriture d'un fichier texte

3. Ergonomie utilisateur

## 2. La classe Morse

### 2.1 Diagramme de classe



## 2. La classe Morse

### 2.2 Pourquoi utiliser la classe HashMap ?

```
import java.util.HashMap;  
private HashMap<String,String> alphaVersmorse;  
private HashMap<String,String> morseVersalpha;
```

- Association clé / valeur
- Tri / Parcours rapides et optimisés
- Simplicité d'implémentation

## 2. La classe Morse

### 2.3 Décomposition des fonctions

#### Les fonctions de traduction :

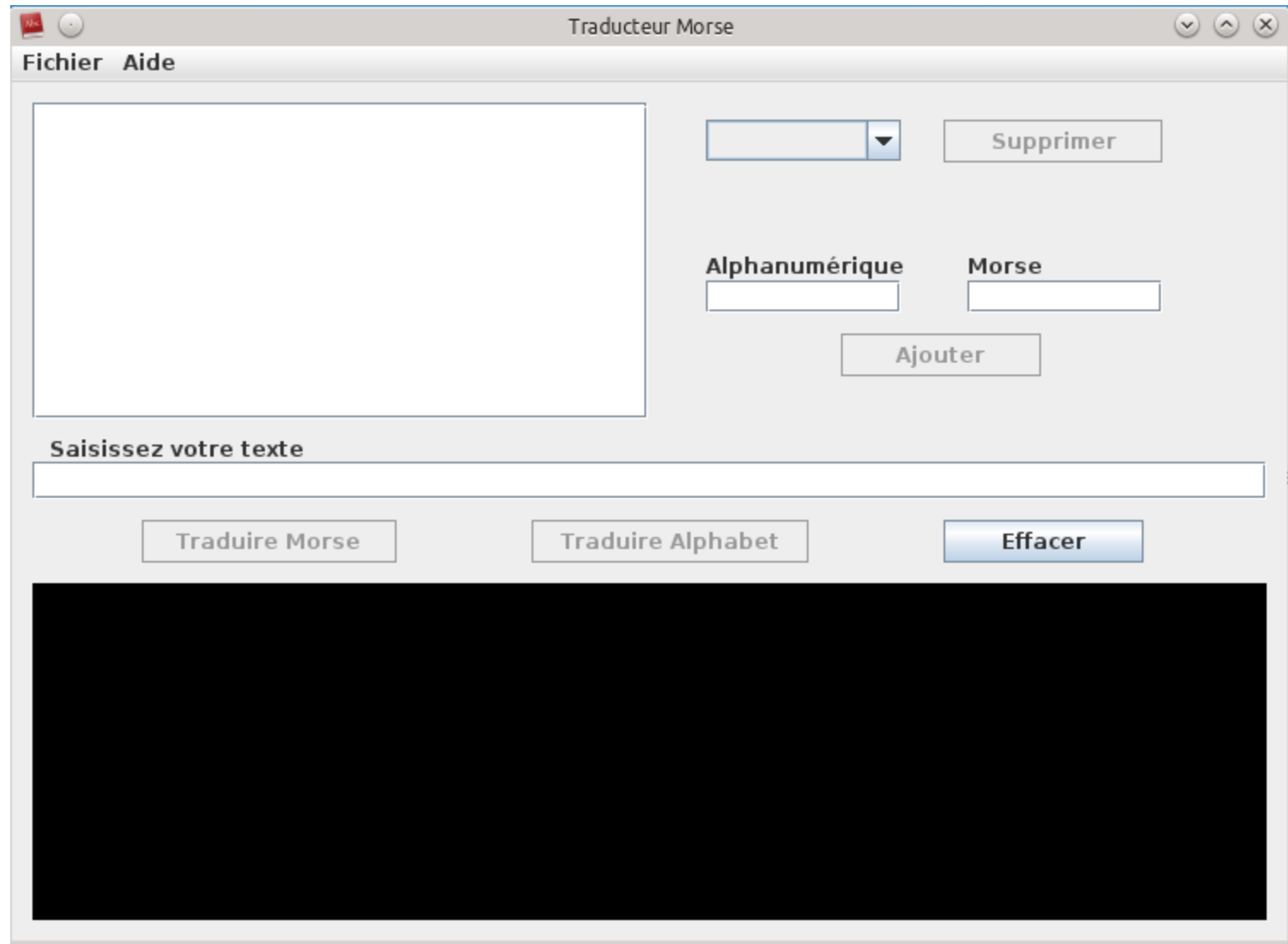
- Traitement des caractères : **String** *alphaToMorseChar()*
- Traitement des chaînes de caractères : **String** *morseToAlpha()*
- Traitement des fichiers : **boolean** *alphaToMorseFile()*

#### La gestion du dictionnaire :

- Initialisation du dictionnaire : **boolean** *initDictionnaire()*
- Parcours du dictionnaire : **String** *getDictionnaire()*
- Ajout / Suppression du dictionnaire : **void** *addToDictionnaire()*

*Utilisation d'une classe outils Tools pour lecture / écriture fichiers*

### 3. L'interface graphique



## 4. Feedback

1. Apprentissage de Java et interface graphique en Java

2. Gestion de projet : Git / GitHub, versioning, documentation

3. Choix de solution



# Merci beaucoup pour votre attention

***Kévin MASCHTALER*** – Licence Professionnelle Génie Logiciel

***Pascal CHEVALLOT*** – Licence Professionnelle Génie Logiciel