

DOCUMENTATIONS

I Diagramme de classe

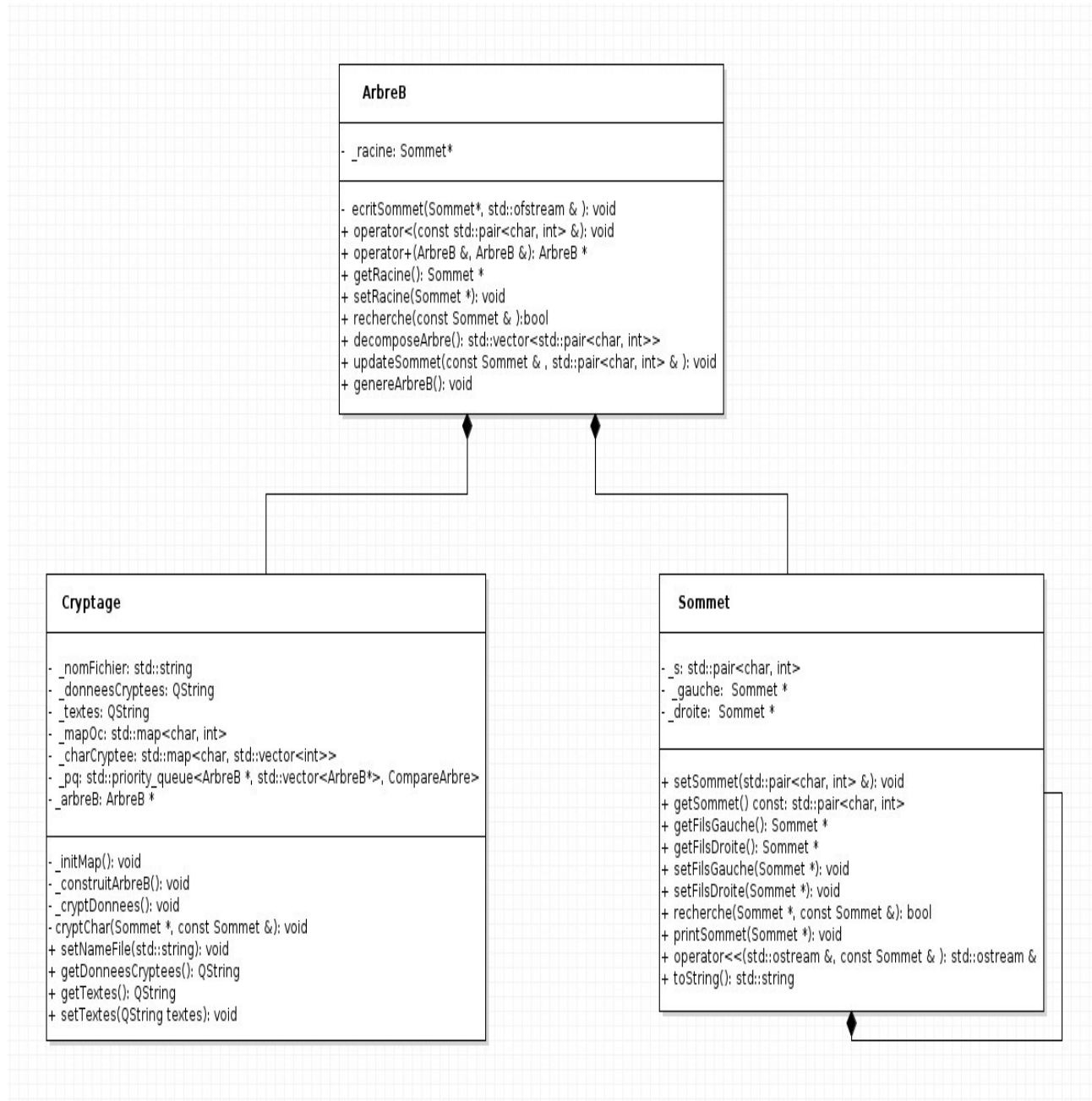


DIAGRAMME DE CLASSE

II Description des classes

1. Classe ArbreB

La classe ArbreB représente un arbre binaire composé par un sommet qui est la racine de l'arbre.

i. Attributs

- `_racine` : est la racine de l'arbre.

ii. Méthodes

- `ecritSommet` : elle écrit chaque sommet de l'arbre dans le flux passé en argument.
- `operator+` : elle fusion deux arbres binaire et retourne l'arbre fusionné.
- `operator<` : ajoute un sommet dans l'arbre binaire.
- `recherche` : cherche le sommet passé en argument dans l'arbre et retourne vrai si le sommet est dans l'arbre sinon faux.
- `decompose` : elle décompose l'arbre et retourne tous les feuilles de l'arbre dans un vecteur
- `generateArbre` : génère un fichier Graphviz contenant les relations de tous les sommets de l'arbre binaire.
- `updateSommet` : met à jour le sommet passé en argument dans l'arbre binaire si le sommet est une feuilles d'arbre.

2. Classe Sommet

Cette classe représente un sommet de l'arbre binaire , elle composé par une paire de valeur une lettre et son nombre d'occurrence , un pointeur vers son fils de gauche et vers son fils de droite.

i. Attributs

- `_s` : représente une lettre et son nombre d'occurrence.
- `_*gauche` : pointeur sur le fils de gauche.
- `_*droite` : pointeur sur le fils de droite.

ii. Méthodes

- `getSommet` : retourne la paire de valeur qui est une lettre et sa fréquence.
- `getFilsGauche` : retourne le fils de gauche.

- getFilsDroite : retourne le fils de droite.
- setSommet: change la valeur du sommet.
- setFilsGauche: change le pointeur du fils de gauche -
setFilsDroite : change le pointeur du fils de droite.
- printSommet: affiche tous les sommets de manière récursive.
- operator<<: affiche la valeur d'un sommet dont la lettre et sa fréquence dans le flux standard.
- toString : retourne la description du sommet c'est à dire la lettre et son nombre d'occurrence en une chaîne de caractère.

3. Classe Cryptage

Cette classe crypte un fichier ou un text.

i. Attributs

- _nomFichier: nom du fichier à crypter.
- _donneesCryptees: données cryptées
- _textes: texte à crypter
- _mapOc: contient les lettres et leurs nombres d'occurrences.
- _charCryptee: contient le codage de chaque caractère.
- _pq: file de priorité pour créer l'arbre binaire. - _arbreB: arbre binaire.

ii. Méthodes

- _initMap: initialise l'attribut _mapOc. - _construitArbreB : construit l'arbre binaire
- _cryptDonnees: crypte tous les lettres.
- cryptChar: chiffre un caractère
- setNameFile: change le nom du fichier à crypter
- getDonneesCryptees: retourne les données chiffrées.
- getTextes: retourne le texte à chiffrer.
- setTextes : change le texte à chiffrer.

4. Classe Decryptage

Cette classe décrypte un fichier ou simple texte code.

I. Attributs

- _messageDechiffre: contient le texte ou fichier déchiffré.

ii. Methodes:

- decryptMessage: déchiffre un texte passé en argument.
- GetMessageDechiffre: retourne le texte déchiffré.

III GraphViz

GraphViz est un ensemble d'outils qui manipulent des graphes définis à l'aide de scripts suivant le langage DOT. Dans ce projet nous utilisons GraphViz afin de générer un arbre binaire sous format png pour afficher l'image générée dans une interface graphique. La méthode `ecritSommet` de la classe `ArbreB` écrit les relations entre les sommets de l'arbre binaire en respectant la syntaxe du langage DOT dans le flux passé en argument. Pour assurer l'unicité des sommets parents qui ont été obtenu en fusionnant deux autres sommets, nous ajoutons la lettre du sommet du fils gauche dans le sommet parent.

IV Interface graphique

L'interface est composée d'un bouton qui charge un fichier local, un champ de texte qui permet de saisir un texte à crypter, un bouton qui crypte un texte saisi et deux champs de texte immuable qui affichent le texte à chiffrer et chiffré.

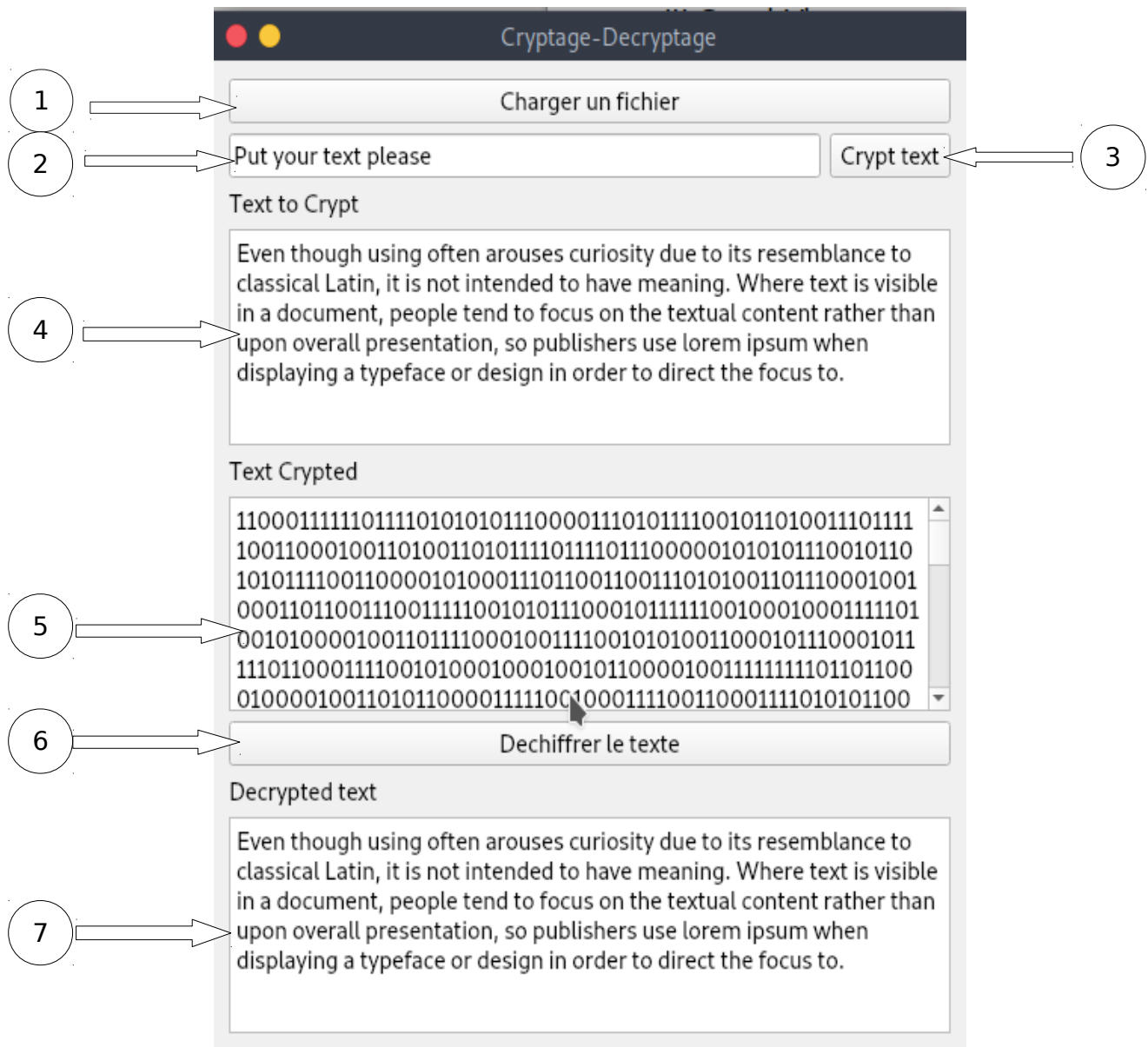
Une fois le fichier charger ou le texte saisi le programme ouvre une autre interface pour afficher l'arbre binaire généré.

INTERFACE DE CRYPTAGE / DECRYPTAGE

1. Bouton qui charge un fichier à crypter et affiche l'arbre généré
2. Champ de texte
3. Bouton qui permet de crypter le texte contenu dans le 2 et affiche l'arbre généré
4. Champ de texte contenant le texte chargé depuis un fichier.
5. Champ de texte contenant le texte chiffré.

6. Button pour déchiffrer le texte.

7. Champ de texte contenant le texte déchiffré.



V Installation

Les procédures de l'installation sont dans le fichier README.md du projet. (Pour mieux gérer les caractères, nous avons utilisé des textes en anglais).