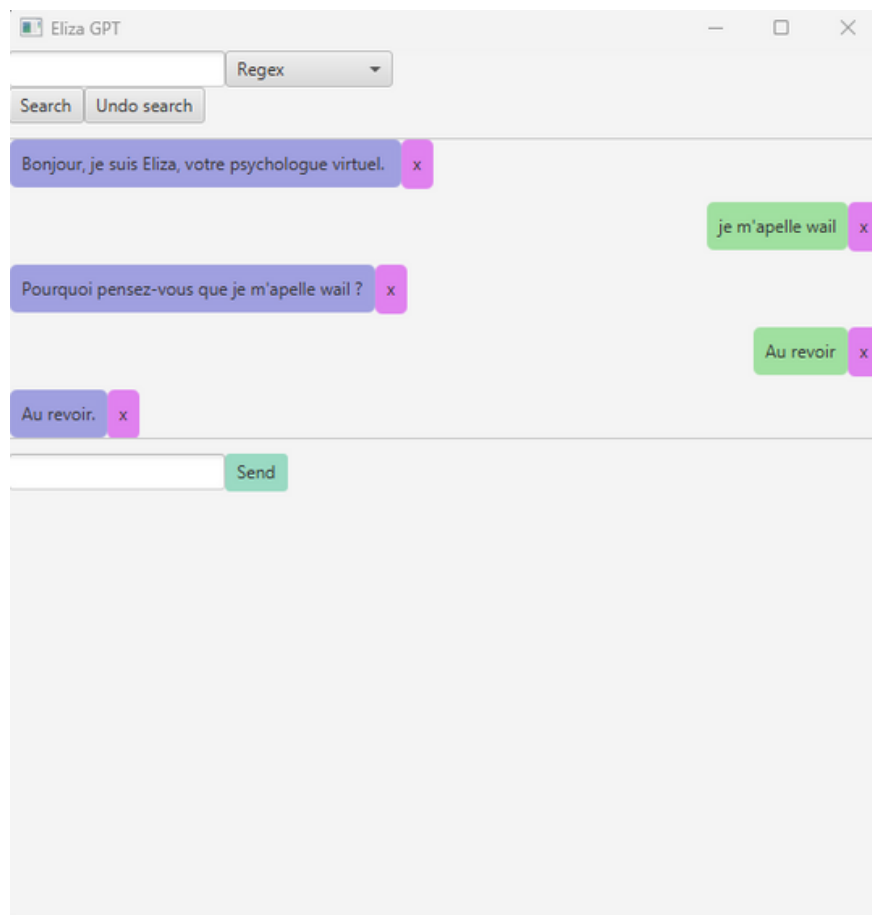


MINI PROJET ELIZA GPT

RAPPORT



SOMMAIRE

01. Présentation

02. Design patterns

03. Ethique

04. Tests

Présentation

Au fur et à mesure des tps nous avons développé une application avec un chat bot à partir d'un squelette.

TP1:

- Le TP1 nous a introduit à un squelette de projet en Java nommé ElizaGPT. Ce squelette comprenait trois fichiers essentiels :
 - App.java - responsable de l'initialisation de l'application.
 - MessageProcessor.java - Une class MessageProcessor avec des fonctions de traitement de messages. Une logique initiale pour les réponses d'un bot à un utilisateur. Une classe Verbe a été conçue pour la conjugaison des verbes, offrant un début de compréhension de la logique nécessaire pour élaborer des réponses cohérentes.
 - JfxView.java - gérant l'interface utilisateur à l'aide de JavaFX. Cependant, cette première itération impliquait une structure où la logique métier était partiellement imbriquée dans l'interface (par exemple, la génération de réponses se produisait dans la vue).

TP2:

- Après la phase initiale de développement, notre parcours s'est enrichi grâce au TP2, une étape cruciale où nous avons replongé dans l'univers de Git et découvert l'intégration continue ou nous avons implémenté dans notre Projet sur git un fichier GitLab CI.

TP3:

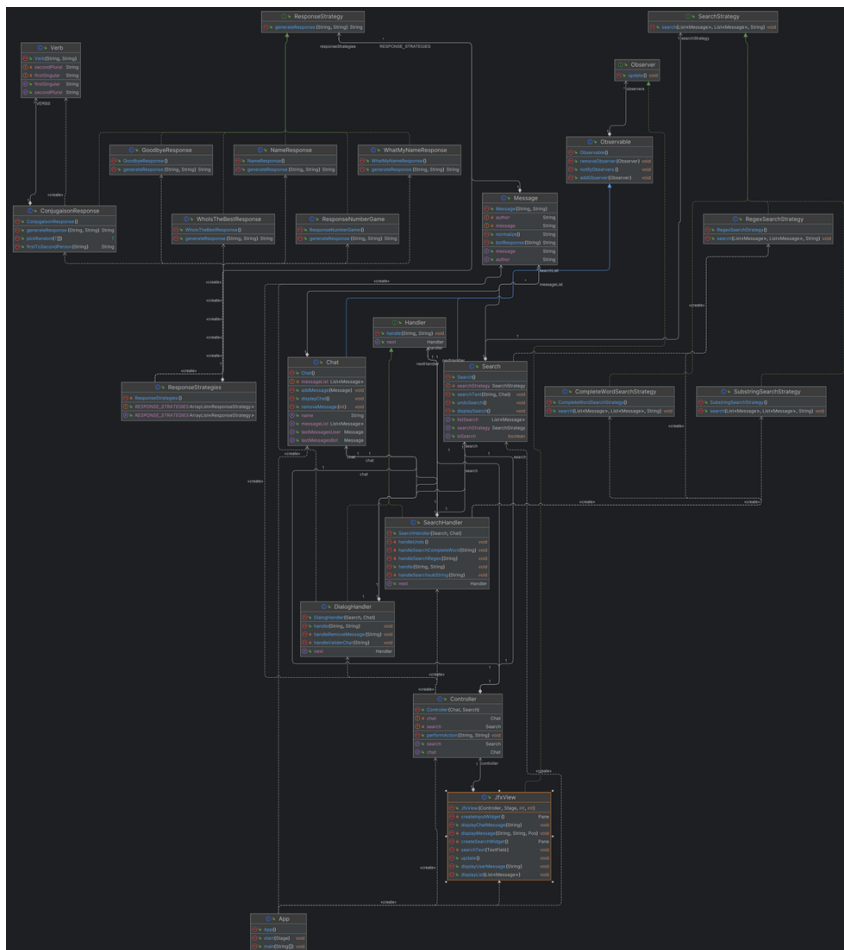
- Au cours du TP3, notre parcours s'est concentré sur l'amélioration de notre projet ElizaGPT, mettant l'accent sur l'application de design patterns, en particulier le pattern MVC, tout en étendant les fonctionnalités existantes. Nous avons entrepris un refactoring significatif en adoptant le pattern Modèle-Vue-Contrôleur (MVC). Puis nous avons mis en place Un pattern Chaîne de responsabilité, un pattern Fabrique, un pattern Stratégie. Puis nous avons ajouter des extensions à l'aide de l'utilisation de ces patterns.

Design patterns

Nous avons mis en place plusieurs principe et plusieurs pattern pour améliorer notre projet, voici des explications détaillées de ce que nous avons fait:

Pattern GRASP:

- Nous avons essayé le plus possible d'utiliser les principes Grasp pour mettre en place les fonctionnalités de chaque classe. Nous avons bien essayé de déterminer le rôle de chaque classe, quelle est sa responsabilité, quelle donnée il instancie. Mais nous avons eu du mal à avoir des classes avec le moins de dépendance possible. Voici un diagramme UML de notre projet



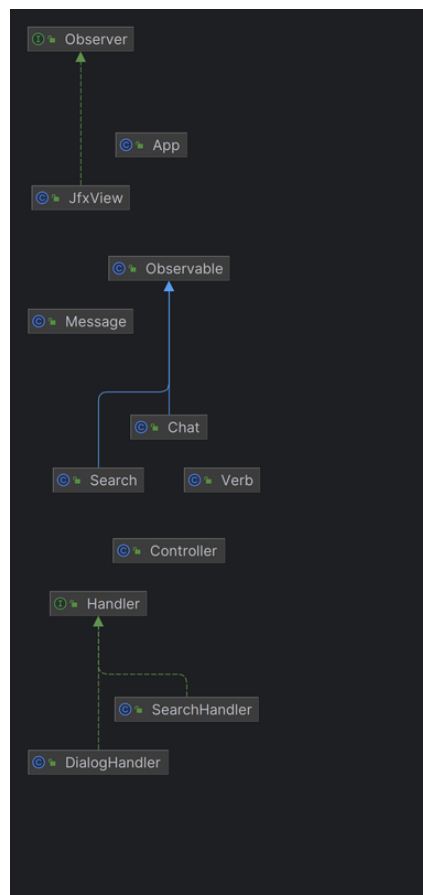
Comme on peut le voir sur se diagramme beaucoup de classe utilisent la classe chat car elle permet nous seulement d'afficher le dialogue entre l'utilisateur et le bot mais on fait aussi des filtres à partir de cette classe.

Mais les classes on chacun leur rôle ce qui contribue à une forte cohésion, nous avons utiliser le polymorphisme une fabrication pure (class search par exemple) pour limité le fort couplage. Nous avons aussi fait des indirections (lié à des pattern de comportement) l'aide de l'utilisation de ces patterns.

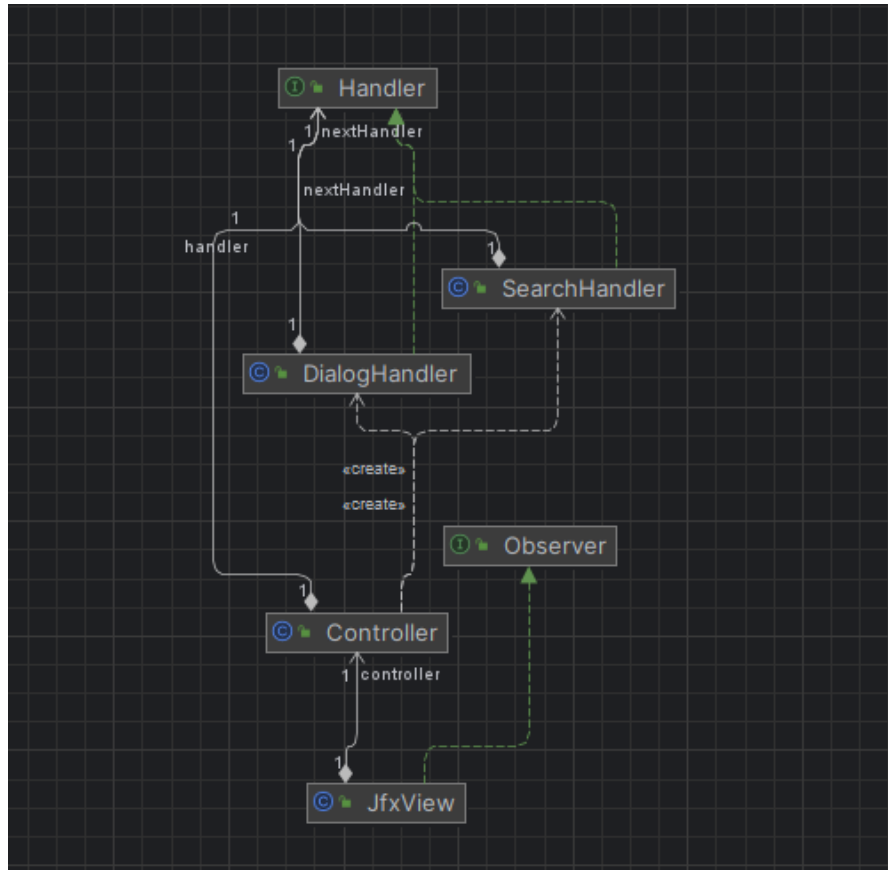
Pour la structure du Projet nous avons utiliser le pattern de conception MVC (Modèle-Vue-Contrôleur) en combinaison avec le modèle de conception Observer.

Pattern MVC:

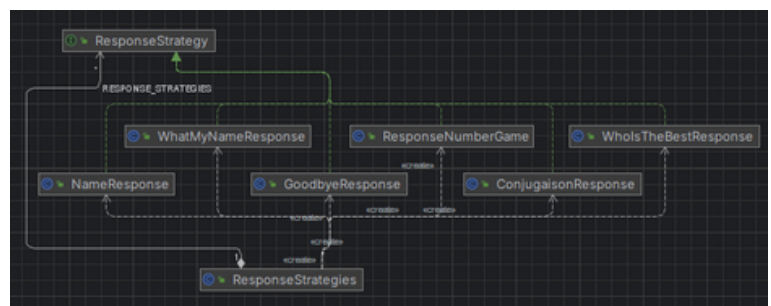
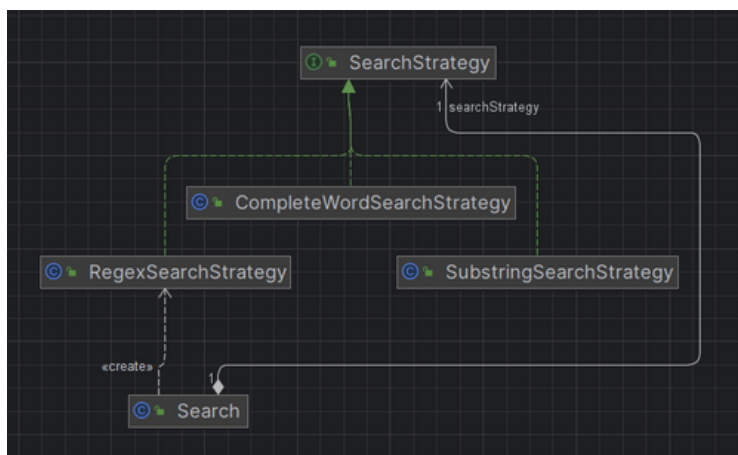
- **Modèle (Model)** : Le modèle représente les données et la logique métier. Dans notre code, les classes Message, Chat et Search joue le rôle de modèles. Par exemple, la classe Search contient la logique pour effectuer une recherche dans une liste de messages.
- **Vue (View)** : La vue est responsable de l'affichage des données à l'utilisateur et de la capture des actions de l'utilisateur. La classe JfxView joue le rôle de la vue. Elle contient le code pour créer l'interface utilisateur et afficher les messages de chat.
- **Contrôleur (Controller)** : Le contrôleur est utilisé pour gérer les actions de l'utilisateur et mettre à jour la vue et le modèle en conséquence. Dans notre code, la classe Controller joue le rôle du contrôleur. Elle reçoit les actions de l'utilisateur de la classe JfxView et effectue les opérations correspondantes sur les modèles.
- **Observer Pattern** : Le modèle Observer est utilisé pour maintenir la cohérence entre la vue et le modèle. Lorsque l'état du modèle change, la vue est automatiquement mise à jour. Dans votre code, la classe JfxView implémente l'interface Observer, ce qui signifie qu'elle est informée lorsque l'état du modèle change. Par exemple, lorsque la liste des messages de recherche dans la classe Search est mise à jour, la méthode update de la classe JfxView est appelée pour mettre à jour l'affichage.



Un des patterns que nous avons ajouter est le pattern handler (chaine de responsabilité). Chaque handler décide soit de traiter la demande, soit de la passer au prochain handler dans la chaîne. Nous avons décider avec le controller c'est a dire que l'utilisateur lorsqu'il fait une action ca passe par le controller qui va le faire passer sur tous les handler jusqu'au handler qui traite cette action.



Nous avons aussi mis en place le pattern strategy. De 2 manière : les réponses du bot et par rapport aux méthodes de recherche.



Ethique

Les intelligences artificielles conversationnelles, telles que ChatGPT et Bard, suscitent des préoccupations éthiques majeures. Ces technologies soulèvent des enjeux variés qui touchent à la fois les éditeurs de programmes et les utilisateurs.

Enjeux :

Confidentialité des données :

La confidentialité des données est le plus gros problème soulevé parmi les enjeux, les conversations avec ces IA impliquent la transmission de données sensibles qui sont à leurs tours stockés et protégées, mais surement réutiliser ce qui met une barrière pour la confidentialité des utilisateur.

La Transparence :

La transparence en intelligence artificielle implique la divulgation des applications, de la logique et des algorithmes utilisés. Bien que cruciale pour le contrôle public des décisions automatisées, une déclaration générique sur l'IA ne suffit pas à prévenir les risques. Les droits de propriété intellectuelle peuvent entraver l'accès à la structure des algorithmes. La complexité des modèles, notamment l'apprentissage profond, rend l'exercice de transparence difficile. Les algorithmes dynamiques et la nécessité de ressources pour détecter les biais posent des défis à la gouvernance des données.

Source : Lignes directrices sur l'intelligence artificielle et la protection des données

adopté par le comité de la convention pour la protection des personnes à l'égard du traitement des données à caractère personnel le 25 janvier 2019

Risques et bénéfices :

Risques pour la société :

- Les systèmes d'IA peuvent être vulnérables aux attaques malveillantes, compromettant ainsi la sécurité des données personnelles.

Source : Rapport du Sénat sur la cybersécurité et la confiance numérique (2018)

- L'automatisation alimentée par l'IA peut entraîner la suppression d'emplois, en particulier dans des secteurs tels que la production industrielle et les services.

Source : Rapport du Conseil d'Analyse Économique (CAE) sur l'intelligence artificielle et emploi (2019)

- Les algorithmes d'IA peuvent refléter des biais présents dans les données d'entraînement, ce qui peut conduire à des résultats discriminatoires. Par exemple, des biais liés au genre ou à la race peuvent être perpétués.

Source : Rapport du Conseil National du Numérique (CNNum) sur l'éthique de l'intelligence artificielle (2017)

Les cas de suspension de ChatGPT en Italie et le suicide lié à l'utilisation d'un robot conversationnel soulignent les conséquences potentiellement graves des défaillances éthiques des IA.

Bénéfices pour la société :

L'intelligence artificielle (IA) offre des avantages significatifs à la société, notamment dans les domaines de la santé, de l'éducation, de l'industrie, de l'économie, de la mobilité, de l'environnement et de la finance. Elle améliore les diagnostics médicaux, personnalise l'éducation, optimise les processus industriels, stimule l'innovation économique, favorise la mobilité autonome et contribue à la gestion environnementale. Cependant, des préoccupations éthiques, telles que la protection des données et l'impact sur l'emploi, nécessitent une réglementation appropriée pour garantir que l'IA profite de manière éthique et équitable à la société.

Mesures légales et techniques :

Législation sur l'IA :

La nécessité de réglementer le développement et l'utilisation des IA est soulignée par des experts, dont Elon Musk. Le respect du GDPR et d'autres normes de protection des données est crucial.

source : https://www.lemonde.fr/economie/article/2023/03/29/elon-musk-et-des-centaines-d-experts-reclament-une-pause-dans-le-developpement-de-l-ia_6167461_3234.html

Exploitation éthique des travailleurs :

Des préoccupations éthiques ont été soulevées quant à l'exploitation de travailleurs kényans dans le développement de ChatGPT. Les éditeurs doivent prendre des mesures pour garantir des conditions de travail équitables.

source : <https://www.journaldugeek.com/2023/01/19/chatgpt-est-il-devenu-plus-ethique-grace-a-l-exploitation-de-travailleurs-kenyans/>

Limites de l'Implémentation (Projet Scolaire) :

Dans le cadre d'un projet scolaire, des mesures simplistes peuvent être mises en place, telles que des avertissements sur l'utilisation éthique de l'IA. Cependant, il est crucial de reconnaître les limites de ces approches et de souligner la nécessité d'une réglementation plus approfondie dans des applications réelles.

Conclusion :

Les IA conversationnelles posent des défis éthiques significatifs, de la transparence à la protection des données. Les éditeurs doivent prendre des mesures légales et techniques pour atténuer les risques, tout en veillant à maximiser les bénéfices sociaux. Les références citées fournissent une base solide pour comprendre ces enjeux et guider la réflexion sur des solutions éthiques dans le domaine des IA conversationnelles.

Tests

Nos tests sont utiles pour s'assurer que les fonctionnalités de base de votre application fonctionnent correctement. Ils testent les fonctionnalités de toutes les classes pour vérifier si tout fonctionne sans problème.

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 71, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jacoco:0.8.10:report (report) @ ELIZA-GPT ---
[INFO] Loading execution data file C:\Users\adnan\Desktop\M1\MIF01 - genie logiciel\mif01\eli
[INFO] Analyzed bundle 'ELIZA-GPT' with 20 classes
[INFO]
[INFO] --- checkstyle:3.3.0:check (validate) @ ELIZA-GPT ---
[INFO] Début de la vérification...
Vérification terminée.
[INFO] You have 0 Checkstyle violations.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.909 s
[INFO] Finished at: 2023-12-17T23:51:48+01:00
[INFO] -----
[WARNING]
[WARNING] Plugin validation issues were detected in 2 plugin(s)
[WARNING]
[WARNING] * org.jacoco:jacoco-maven-plugin:0.8.10
[WARNING] * org.apache.maven.plugins:maven-resources-plugin:3.3.0
[WARNING]
[WARNING] For more or less details, use 'maven.plugin.validation' property with one of the va
[WARNING]

Process finished with exit code 0
```

Comme nous pouvons le remarquer 71 tests sont passés sans aucunes Failures ce qui est bon pour notre projet car cela indique que toutes les méthodes de nos classes fonctionnent sans aucun problème avec différentes valeurs/paramètres.

Nous avons aussi réalisé des tests manuelles pour notre interface graphique (View) comme par exemple tester les fonctionnalités d'ajout et de suppression de messages, ainsi que la récupération des derniers messages et du nom de l'utilisateur..