

Rapport de Projet : Simulation de la Presse Française

Introduction:

Le projet "Simulation de la Presse Française" a été conçu pour modéliser et simuler les événements et la propriété des médias dans le paysage médiatique français. L'objectif principal est de fournir une application interactive permettant de gérer des entités telles que des médias et des personnes, tout en simulant des événements comme des rachats ou des publications. Ce projet repose sur une architecture modulaire et utilise des concepts avancés de programmation orientée objet, notamment le patron de conception Observateur.

L'application est entièrement basée sur une interface console, avec une expérience utilisateur intuitive et quelque peu esthétique. Ce rapport détaille les réalisations, la structure du projet, les fonctionnalités clés et les difficultés rencontrées au cours du développement.

Réalisations

1. Chargement des Données

Le projet utilise des fichiers .tsv pour initialiser les données. Ces fichiers contiennent des informations sur les médias, les personnes physiques, les organisations et leurs relations. Un module dataloader a été développé pour parser ces fichiers et charger les données dans des objets Java.

Fonctionnalité Clé : dataloader

Rôle : Lire les fichiers .tsv et convertir les données en objets Java.

Points Importants :

Gestion des erreurs de formatage pour garantir l'intégrité des données.

Validation des données pour éviter les doublons ou les incohérences.

Chargement des entités dans des collections pour un accès rapide et efficace.

Cette fonctionnalité garantit que l'application dispose d'une base de données initiale cohérente et prête à être utilisée pour les simulations.

2. Modélisation des Entités

Les entités principales du projet incluent :

Media : Représente un média avec un identifiant et un nom.

PersonnePhysique : Représente une personne avec un identifiant et un nom.

Evenement : Représente un événement avec un type, une date et une description.

Fonctionnalité Clé : Classe Evenement

Rôle : Modéliser les événements tels que les rachats ou les publications.

Attributs :

type : Type de l'événement (e.g., "rachat", "publication").

date : Date de l'événement.

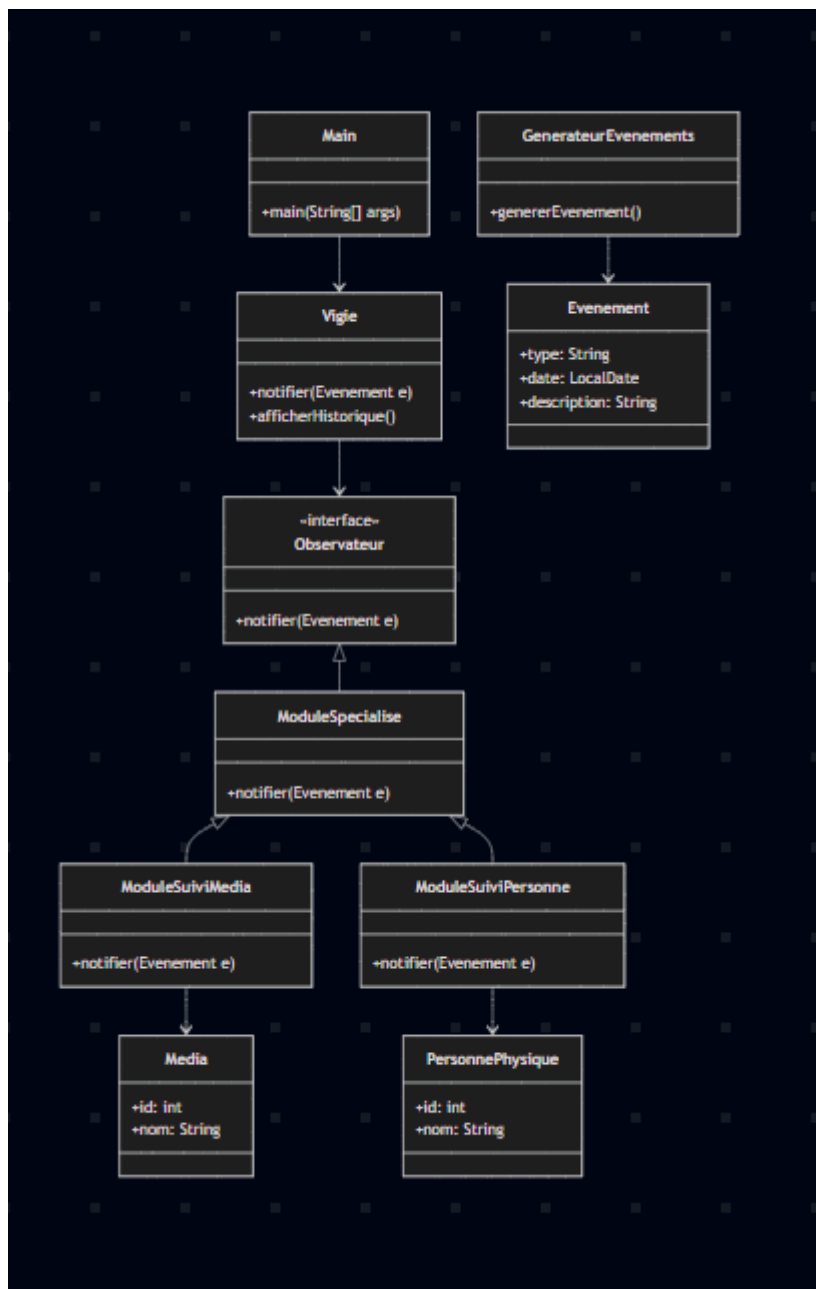
description : Détails de l'événement.

Utilisation :

Les événements sont générés par le module GenerateurEvenements et propagés aux modules spécialisés.

Cette modélisation permet de capturer les interactions complexes entre les entités et de les représenter de manière structurée.

Diagramme de classe (fait sur Mermaid chart). Désolé de la lisibilité la capture d'écran n'as pas de haute résolution.



3. Gestion des Événements

Un générateur d'événements (GenerateurEvenements) a été implémenté pour simuler des événements tels que des rachats ou des publications. Ces événements sont propagés aux modules spécialisés via le patron de conception Observateur.

Fonctionnalité Clé : GenerateurEvenements

Rôle : Générer des événements aléatoires ou basés sur des entrées utilisateur.

Points Importants :

Génération dynamique d'événements avec des descriptions personnalisées.

Intégration avec les modules spécialisés pour notifier les entités concernées.

Fonctionnalité Clé : Vigie

Rôle : Centraliser la gestion des notifications et des alertes.

Méthodes :

notifier(Evenement e) : Propager un événement à tous les observateurs enregistrés.

afficherHistorique() : Afficher l'historique des alertes reçues.

Ces fonctionnalités assurent une gestion efficace des événements, tout en maintenant une traçabilité complète grâce à l'historique.

4. Interface Console

Une interface console a été développée pour permettre à l'utilisateur d'interagir avec l'application. Elle inclut :

Un menu esthétique en ASCII art que j'ai d'abord essayé de réaliser moi même mais que j'ai trouvé dur à faire. J'ai donc utilisé copilot(chatgpt) pour cette partie du rendu pour avoir quelque chose d'un minimum esthétique.

Des options pour ajouter des entités, simuler des événements et afficher l'historique des alertes.

Fonctionnalité Clé : Menu Principal

Rôle : Offrir une navigation intuitive à l'utilisateur.

Options :

Ajouter un média surveillé.

Ajouter une personne surveillée.

Simuler un événement.

Afficher l'historique des alertes.

Quitter l'application.

L'interface console a été codée avec des séparateurs visuels et un logo ASCII pour améliorer l'expérience utilisateur. Chaque option du menu est associée à une fonctionnalité clé, permettant une interaction intuitive (chacun ses standards).

Structure du Projet

Organisation des Fichiers

Le projet est organisé en plusieurs packages :

core : Contient les classes principales comme Vigie, GenerateurEvenements et Observateur.

model : Contient les entités comme Media, PersonnePhysique et Evenement.

modules : Contient les modules spécialisés comme ModuleSuiviMedia et ModuleSuiviPersonne.

util : Contient des utilitaires comme dataloader pour le chargement des données.

tout ces packages sont dans le src et le package data est à [part](#). Il contient les différents tableaux de médias, personnes morales et physique etc..

Cette organisation modulaire facilite la maintenance et l'extensibilité du projet.

Difficultés Rencontrées

1. Gestion des Données

Le chargement des données à partir des fichiers .tsv a nécessité une attention particulière pour gérer les erreurs de formatage et les données manquantes. Des validations supplémentaires ont été mises en place pour garantir l'intégrité des données. Au début du projet le fichier média.tsv s'était mal chargé et n'affichait rien, ce qui m'a fait perdre beaucoup de temps à chercher pourquoi certaines fonctionnalités ne voulaient pas marcher.

2. Implémentation du Patron Observateur

L'implémentation du patron de conception Observateur a posé des défis, notamment pour garantir que les modules spécialisés reçoivent les notifications appropriées. Une attention particulière a été portée à la synchronisation des notifications.

3. Esthétique de l'Interface

Créer une interface console esthétique tout en maintenant une lisibilité et une simplicité a demandé plusieurs essais infructueux de ma part et finalement le recours à l'IA. Le logo ASCII et les séparateurs visuels ont été ajoutés pour améliorer l'expérience utilisateur.

4. Gestion des Exceptions

Assurer une gestion robuste des exceptions, notamment pour les entrées utilisateur, a été un aspect critique pour éviter les plantages de l'application. Des messages d'erreur clairs et informatifs ont été intégrés.

Conclusion:

Ce projet a permis de modéliser efficacement les interactions et les événements dans la presse française. Grâce à une architecture modulaire et à l'utilisation de concepts avancés, l'application est extensible et maintenable. Les défis rencontrés ont été surmontés avec succès, et le résultat final est une application fonctionnelle et un peu esthétique.

Ce projet constitue une base solide pour des extensions futures, telles que l'intégration d'une interface graphique(que je n'avais pas la foie d'implémenter malheureusement) ou l'ajout de nouvelles fonctionnalités comme l'analyse statistique des événements.