

# Mini projet programmation Orientée Objets

---

*Simulateur de trading (v2)*  
*1AINFO*

## **Objectif**

Il s'agit de développer une application permettant à son utilisateur de faire une simulation de trading en bourse (vente et achat d'actions) manuellement ou via des algorithmes.

## **Description détaillée**

L'application va permettre à un trader (humain ou algorithmique) de commencer une simulation de trading à partir d'une date donnée et pendant un certain nombre de jours. Le trader commence la simulation avec un solde de départ et prend chaque jour un certain nombre des décisions d'achat et/ou de vente d'actions. Le trader peut décider de ne plus faire de transactions (décision autre que VENDRE ou ACHETER), ceci indique à l'application qu'il faut passer au jour suivant (si la date de fin de la simulation n'a pas été atteinte).

Le prix de vente ou d'achat d'une action est déterminé en fonction de la date courante de la simulation à partir d'un historique de prix journaliers fourni sous forme d'un fichier texte structuré. En effet, seule la bourse a accès à l'historique complet des prix journaliers et les traders sont ainsi obligés de faire des requêtes pour prendre connaissance des prix. Bien évidemment, les décisions prises par les traders sont seulement basées sur les prix du jour et des jours précédents puisqu'ils ne doivent avoir aucun accès aux prix futurs.

Quand il s'agit d'un trader humain, l'application interagit avec lui via les entrées/sorties standards (clavier et console) pour qu'il puisse donner ses ordres et consulter les prix journaliers. Cependant, un algorithme de trading devrait être capable d'acheter ou de vendre les actions de manière autonome et

automatique. A la fin de la simulation, les gains ou les pertes sont affichés en termes de valeur monétaire et de pourcentage.

### **Besoins fonctionnels**

L'utilisateur devrait être capable de :

- Lancer une nouvelle simulation en spécifiant :
  - o la date du jour
  - o le montant à investir (solde de départ)
  - o mode de décision (manuel ou via un algorithme).
  - o Spécifier le nombre de jours de la simulation.
- Voir les détails du portefeuille à la date du jour (affichage à la fin de la journée)
  - o Montant restant et actions en possession.
  - o La valeur totale du portefeuille (solde + prix actuel des actions en possession).
  - o Montant et pourcentage des gains/pertes.
- Lorsque le trader est humain, lui permettre de voir les prix journaliers d'une action (ou de toutes) pour une période donnée.
- Accéder aux résultats des anciennes simulations (date, trader, montant de départ, date départ, nombre de jours, %gains) enregistrées dans un fichier.

### **Besoins non fonctionnels**

- Ergonomie
- Scalabilité et performances prévisibles (complexité de structure de données et d'algorithmes).
- Extensibilité, configurabilité et réutilisabilité
- Robustesse (garantir que les éventuels traders respectent les contraintes métier).
- Application des bonnes pratiques de la POO

### **Critères d'évaluation et remarques**

- Couverture des besoins fonctionnels.

- Démonstration de l'extensibilité par substitution/ajout de traders et de structures de données pour le stockage de l'historique.
- Effort employé pour respecter les besoins non fonctionnels.
- Justification de la conception.
- Discussion des différentes approches en termes de complexité et de temps de réponse en fonction de la taille des données (historique).
- La qualité/complexité des algorithmes de décision/trading n'est pas prioritaire et est considérée comme secondaire par rapport aux besoins fonctionnels et non fonctionnels.