

Projet de programmation orienté objet en C++.

## 1 INTRODUCTION

---

Le projet est un backtester orienté objet qui simule des stratégies de trading sur des données historiques. L'objectif est de pouvoir tester plusieurs stratégies sur différents actifs, tout en suivant la gestion du portefeuille et le calcul des gains et pertes.

**Vous trouverez dans le dossier « doc » du projet une courte vidéo d'introduction au trading afin d'avoir une meilleure vision d'ensemble.**

## 2 CLASSES

---

### Candle

Rôle : Représente une bougie de marché avec open, high, low, close et date.

Interactions :

- Fournit les données aux stratégies (onCandle) pour générer des signaux.
- Sert aux positions pour calculer le PnL non réalisé et vérifier si SL/TP sont atteints.

+++++

### Position

Rôle : Représente une position ouverte sur un actif.

Interactions :

- Reçoit les prix du marché via Candle.
- Est gérée par le Portfolio qui décide quand ouvrir ou fermer.

+++++

### Portfolio

Rôle : Gestion globale du capital et des positions.

Responsabilités :

- Calcul du capital disponible et de la marge utilisée.
- Ouverture et fermeture de positions selon les signaux des stratégies.
- Calcul de la taille de position en fonction du RRR.
- Fermeture automatique de toutes les positions ouvertes en fin de backtest (closeAllPositions).

Interactions :

- Reçoit les signaux des stratégies.
- Crée et gère les positions de façon centralisée.
- Lit les Candle pour la valorisation du portefeuille.

+++++

### **IStrategy (interface)**

Rôle : Classe abstraite contenant des méthodes que toutes les stratégies doivent implémenter.

Interactions :

- Lit les Candles pour calculer les indicateurs.
- Lit le Portfolio pour récupérer le capital et le RRR afin de calculer la taille de position.
- Retourne un OrderRequest au Portfolio qui va décider de l'exécuter.

+++++

### **Stratégies concrètes**

Chaque stratégie hérite de IStrategy et implémente sa logique de décision :

- Bollinger : utilise les bandes de Bollinger pour détecter les cassures et générer BUY/SELL.
- Momentum : détecte la force du momentum pour entrer ou sortir d'une position.
- SMA : détecte les croisements de moyennes mobiles pour générer des signaux.

Interactions :

- Les stratégies n'exécutent jamais directement de trades. Elles renvoient des ordres au Portfolio, qui se charge de la taille et de l'ouverture/fermeture de positions.
- Les SL/TP sont calculés en fonction du prix d'entrée et du RRR, et la taille de position est ajustée en fonction du SL.

+++++

### **Backtester**

Rôle : Coordonne la simulation.

Déroulement de la simulation :

- Charge les données historiques.
- Parcourt les bougies une par une
- Passe la bougie aux stratégies (onCandle).
- Passe les ordres générés au portefeuille pour exécution.
- Met à jour les PnL non réalisés des positions ouvertes.
- À la fin, ferme toutes les positions restantes et calcule les résultats finaux (PerformanceReport).

+++++

### **MarketData**

Rôle : Fournit les données historiques ou en temps réel pour un ou plusieurs actifs.

Interactions :

- Alimente les stratégies avec les bougies pour prendre leurs décisions.
- Permet au Portfolio et aux Positions de connaître le prix actuel pour calculer PnL et marges.

+++++

### **Order**

Rôle : Représente un ordre concret envoyé au marché.

Interactions :

- Généré par le Portfolio à partir d'un OrderRequest.
- Permet de créer une Position LONG ou SHORT.