**🚀 Базовый чек-лист для твоего будущего бота:**

✅ Подключение к API Binance (Spot)

✅ Получение котировок (OHLCV) по таймфрейму (1h)

✅ Расчёт RSI (14), SMA (20), ATR (14), MACD (12,26,9)

✅ Логика входа:

* RSI < 30 и Price > SMA → BUY
* RSI > 70 и Price < SMA → SELL
  + подтверждение по MACD (DIF>DEA для BUY)
  + подтверждение по ATR (чтобы не входить на "вялых" рынках)

✅ Проверка RRR ≥ 1:2 перед входом

✅ Управление позицией:

* Один вход — один выход
* Проверка текущего состояния позиции
* Разделение капитала на % от депозита
* Проверка minNotional Binance

✅ Учёт комиссий:

* На BUY и SELL
* Расчёт **чистой прибыли PnL**

✅ Логирование:

* Время, цена, объём, RSI, SMA, MACD, ATR
* PnL после каждой сделки
* Комиссии
* Баланс после сделки

✅ Интерфейс:

* На первом этапе — лог в файл
* Будущая версия — уведомления в Telegram

✅ Защита от:

* Ложных повторных входов
* Попытки продать больше, чем есть
* Попытки купить меньше minNotional

## 🚀 **Спринт 4: “Бот навчився думати”**

**Цель**:  
Научить бота **анализировать рынок с помощью RSI и MACD** и принимать решения: покупать, продавать или ждать.

### 📋 Задачи спринта:

#### 1. 📚 Добавить расчёт индикаторов RSI и MACD:

* Использовать ta-библиотеку
* Вычислить RSI (например, с периодом 14)
* Вычислить MACD (12-26-9)
* Добавить эти значения как колонки в DataFrame

#### 2. 🧠 Создать простую стратегию:

* Пример:

python

КопироватьРедактировать

if rsi < 30 and macd > macd\_signal:

signal = "buy"

elif rsi > 70 and macd < macd\_signal:

signal = "sell"

else:

signal = "hold"

* Реализовать функцию, которая принимает df и возвращает список сигналов (по строкам)

#### 3. 🧪 Протестировать стратегию на исторических свечках:

* Пройтись по DataFrame (после dropna)
* Выполнить "бумажную торговлю" — считать:
  + сколько раз купили
  + когда продали
  + какая получилась прибыль

#### 4. 📦 Логировать действия:

* Сохранять сигналы в CSV или лог
* Выводить в консоль, когда "покупка", "продажа", "ожидание"

### 🧾 Формат отчёта по спринту:

* strategies/indicators.py — расчёт RSI и MACD
* strategies/rsi\_macd.py — логика стратегии
* core/engine.py — подключение стратегии и логика торговли
* logs/bot.log — события
* data/ — CSV с OHLCV и результатами

### ⏱ Время на спринт: 3–4 дня (по 2–3 часа в день)

Но ты можешь двигаться в своём ритме 💚  
Если хочешь — разбью задачи ещё мельче на “мини-задания” на каждый день.