# Мануал для разработчика: торговый бот на данных Binance (USDT‑M Futures)

**Диапазон:** пары вида *COIN/USDT* на Binance Futures (USDT‑маржин перпетуалы; при необходимости квартальники для cash‑and‑carry). Только публичные REST/WS-данные Binance, без внешних платных фидов. Язык реализации — Python.

| ## Содержание 1. Архитектура данных и инженерия потоков 2. Вычисляемые признаки (features) 3. Детектор режимов рынка (гейтинг стратегий) 4. Мета‑селектор (скоринг, приоритеты, неттинг) 5. Стратегии (1–14, 17, 19, 26) — ТЗ для реализации 6. Комбинирование стратегий (готовые «комбо») 7. Управление риском, комиссии и исполнение 8. Бэктест/валидация и контроль дрейфа 9. Конфигурация (пример YAML) 10. Чек‑листы разработчика |
| --- |
| ## 1) Архитектура данных и инженерия потоков ### 1.1 Источники Binance **REST:** - fapi/v1/klines — свечи (1m/5m/15m/1h/4h/1d). Берём также takerBuyBaseAssetVolume для барной CVD. - fapi/v1/aggTrades — тиковая лента (агрессор), для точной CVD и профиля. - fapi/v1/depth — снапшот книги. - fapi/v1/openInterest / futures/data/openInterestHist — текущее и историческое OI. - fapi/v1/fundingRate, fapi/v1/premiumIndex, fapi/v1/markPrice — фандинг/премия/марка. - fapi/v1/continuousKlines — линия контрактов (для квартальников). - fapi/v1/exchangeInfo — шаги цены/лот, минимумы. - fapi/v1/allForceOrders — ликвидации (также есть WS). |
| **WebSocket:** - @depth, @depth@100ms / diffDepth — инкременты книги. - @bookTicker — best bid/ask. - @aggTrade — поток принтов. - @forceOrder — ликвидации. |
| ### 1.2 Ведение локальной книги (обязательно) 1) Раз в N секунд — REST‑снапшот depth (levels ≥ 100). 2) Между снапшотами — принимаем diffDepth, **применяем строго по sequence‑id**. 3) Если пропуск/рассинхрон — **быстрая ресинхронизация** (снапшот+реплей). 4) Храним аггрегированные суммы по 10–20 уровням для метрик imbalance/refill. |
| ### 1.3 Интеграционные правила - Все таймстэмпы в **UTC**. - Консистентность: приводим REST и WS к **единой шкале времени** (nearest bar). - Аггрегирование aggTrades → барная CVD (и профиль) на окнах: 1m/5m/15m/1h/день. - Data Quality: дедуп по tradeId, пропуски WS в логи, алерт на ресинк. |

## 2) Вычисляемые признаки (features)

### 2.1 Волатильность/тренд/структура

* **ATR(14)**, **ATR% = ATR/Close**.
* **ADX(14)** на M15, H1.
* **EMA(20/50/200)** + **наклон** (deg/ATR‑норм).
* **BB bandwidth** = (BB‑Upper − BB‑Lower)/Middle; перцентили p10/p20/p30/p40 на окне ≥ 60–90 дней.
* **Donchian width**.
* **Структура** HH‑HL / LL‑LH (по свинг‑экстремумам).

### 2.2 VWAP/Anchored VWAP и профиль

* **Дневной VWAP** (UTC 00:00–24:00) и ленты ±σ.
* **Anchored VWAP** от событий: дневной open, бар пробоя, значимый свинг.
* **Профиль объёма** (из aggTrades): бининг по цене bin = max(tick, 0.05–0.10% цены) или ~ATR/8; считаем **VAH/VAL/VPOC**.

### 2.3 Потоки/книга/позиционирование

* **CVD**:
  + барная — из takerBuyBaseAssetVolume (klines),
  + тиковая — суммирование buyer/seller по aggTrades.
* **Depth‑imbalance**: (Imb = Ask\_{1..K} / Bid\_{1..K}), K=10–20; **устойчивость 10–30с**.
* **Refill (абсорбция‑прокси)**: быстрое пополнение объёма на уровне после серии агрессивных принтов.
* **ΔOI** за 30–90 мин/сутки, **funding z‑score**, **premium drift**.
* **IB** (первые 30–60 мин дня) и метрики «жирных слотов» (EU/US) — медианный range/объём.

| ## 3) Детектор режимов рынка (гейтинг) |
| --- |
| ## 4) Мета‑селектор: приоритеты, скоринг, неттинг |
| **Приоритет в тренде (одна идея — один вход):** 2 (squeeze) > 1/3 (breakout/ORB) > 6 (momentum); 5 (retest) = re‑entry/добавка. В рейндже: 7/8 базовые, 10 — только при конфлюэнсе с VWAP/Value. |
| **Скоринг сделки (пример, вход при score ≥ +2):** - +1 объём > 1.5× медианы 20 баров - +1 CVD в сторону (или дивергенция для MR) - +1 ΔOI ≥ +1…+3% за 30–90 мин - +1 устойчивый depth‑imbalance 10–30с в сторону - −1 LATE\_TREND / funding‑экстрем - −2 BTC идёт против (на H1) |
| **Неттинг и дедуп:** один символ = одна нетто‑позиция на контекст‑хэш (symbol, зона, направление, режим, TF); дубль игнорируется N баров (cooldown). Добавки — только по структуре (5) и в лимит риска. |

## 5) Стратегии — ТЗ для реализации

Формат по каждой: **Идея • Данные • Сканер (условия) • Триггер входа • Стоп/тейки/менеджмент • Подтверждения • WR/RR • Примечания.**

### 1) Трендовый пробой (Donchian High/Low)

* **Идея:** вход по закрытию за максимум/минимум коридора, жить за счёт RR.
* **Данные:** klines; (опц.) aggTrades/CVD, openInterestHist, WS depth.
* **Сканер:** Donchian(20) пробит закрытием на H1; H4 — EMA50↑ и ADX>20, BBw/ATR% до пробоя низкие (сжатие p30–p40); объём >1.5× мед.20.
* **Триггер:** close за уровнем ≥0.25 ATR **или** 2 close подряд; (вариант 50/50 — часть по ретесту).
* **Стоп:** за свингом/ретест‑свечой +0.2–0.3 ATR.
* **Тейки:** TP1=+1R (частичный выход), TP2=+2–3R, остаток трейли по ATR/свингам.
* **Подтв.:** CVD в сторону, ΔOI +1…+3%, imbalance вперёд, funding не экстремален.
* **WR/RR:** ~38–55% (с фильтрами 45–58%); RR 2–4R.
* **Примеч.:** не брать при LATE\_TREND и против BTC.

### 2) Squeeze→Breakout (NR7 / BB squeeze → Keltner)

* **Идея:** выход из низковолатильного коридора.
* **Данные:** klines; (опц.) aggTrades, OI, depth.
* **Сканер:** BBw < p25 (90 баров), длительность сжатия ≥12 баров; H4 тренд в сторону, ADX>20.
* **Триггер:** close за краем ≥0.25 ATR; 50/50 — часть на ретест.
* **Стоп/тейки:** как в (1).
* **Подтв.:** объём >1.5–2×, CVD dir, ΔOI +1…+3%, depth‑imbalance; funding не экстремален.
* **WR/RR:** ~42–55% → **50–60%** (с фильтрами) → **55–65%** с ретестом; RR умеренный.
* **Примеч.:** не входить, если к моменту сигнала цена >1.5 ATR от 20EMA (переразгон).

### 3) ORB/IRB (30–60 мин «открытие»)

* **Идея:** пробой границы первичного диапазона суток/окна.
* **Данные:** klines по паре и BTC.
* **Сканер:** выбранные слоты (00:00–01:00, 07:00–08:00, 13:30–14:30 UTC); IB\_width < p30 (60д) и <0.8·ATR(14); H4 тренд в сторону.
* **Триггер:** close за IB ≥0.25 ATR; часть — на ретест IB.
* **Стоп/тейки:** за противоположной границей IB +0.2–0.3 ATR; TP по R/свингам.
* **Подтв.:** объём>1.5×, BTC в ту же сторону, (опц.) CVD/OI/depth.
* **WR:** ~40–55% → 48–60% → **55–65%** (полный набор).

### 4) Откат к MA/EMA/Anchored‑VWAP

* **Идея:** покупка «дип» в тренде от динамической опоры.
* **Данные:** klines; AVWAP (из свечей/aggTrades); depth/CVD/OI опц.
* **Сканер:** H4 тренд (EMA50↑, ADX>20), 1–2‑й откат; зона = EMA20±0.3 ATR или AVWAP от пробоя; глубина 0.38–0.62 Фибо.
* **Триггер:** свеча‑отклонение + close над EMA20 **или** пробитие high свечи сигнала; 50/50 — часть лимитом на ретест.
* **Стоп/тейки:** за свинг‑лоу +0.25 ATR; TP1=+1R, далее 2–3R/трейл.
* **Подтв.:** объём>1.2–1.5×, CVD flip вверх, imbalance, OI не падает.
* **WR:** ~45–55% → **52–62%** → **56–66%**.

### 5) Break & Retest

* **Идея:** ретест уровня после подтверждённого пробоя.
* **Данные:** klines; (опц.) aggTrades, depth, OI.
* **Сканер:** пробой с close ≥0.25 ATR и объёмом >1.5–2×; зона ретеста = экстремум±0.2–0.3 ATR ∩ AVWAP(бар пробоя) ∩ (опц. POC/фибо 38–62).
* **Триггер:** 50% лимитом в зоне, 50% — по подтверждению (rebound close/перебитие high свечи реакции).
* **Стоп/тейки:** за свинг‑реакцией +0.2–0.3 ATR; TP как в (1).
* **Подтв.:** CVD flip, imbalance flip/refill, OI не падает (лучше растёт).
* **WR:** ~48–55% → **52–62%** → **56–66%**.

### 6) ATR‑модуль моментума (протяжка)

* **Идея:** дожим импульса с короткой паузы/флага.
* **Данные:** klines; (опц.) aggTrades, OI, ликвидации, depth.
* **Сканер:** импульс‑бар ≥1.4× ATR, close в верхн.20%; follow‑through; H4 тренд; до сопротивления ≥1.5 ATR; **не LATE\_TREND**.
* **Триггер:** пробой high импульса/флага ≥0.2–0.3 ATR или micro‑pullback к EMA9/20.
* **Стоп/тейки:** за low импульса +0.25 ATR; тайм‑стоп 6–8 баров без 0.5 ATR прогресса.
* **Подтв.:** объём>2×, CVD dir, ΔOI +1…+3%, ликвидации в сторону, imbalance.
* **WR:** ~42–55% → **48–60%** → **55–65%**.

### 7) VWAP/Value mean‑reversion (рейндж‑дни)

* **Идея:** возврат к «справедливой цене» (VWAP/POC) в балансе.
* **Данные:** aggTrades (VWAP/профиль/CVD), klines, WS depth.
* **Сканер:** RANGE/CHOP: ADX<20, ATR%<p40, BBw<p30, EMA20/50 плоские, BTC нейтральный. Зоны: VAH/VAL/POC, ленты VWAP±σ, H4‑свинг.
* **Триггер:** свеча‑отклонение + reclaim внутрь value; вход 50/50 (лимит + подтверждение).
* **Стоп/тейки:** за экстремум +0.25 ATR; TP1=VWAP/POC, TP2=середина/противоположная лента; тайм‑стоп 6–8 баров.
* **Подтв.:** CVD‑дивергенция, imbalance flip, OI/funding спокойны.
* **WR:** **50–60%** → **55–65%** → **60–70%** (в истинный рейндж).

### 8) Range‑fade от границ диапазона

* **Идея:** отбой от краёв устоявшегося рейнджа.
* **Данные:** klines, aggTrades (профиль/VWAP), WS depth.
* **Сканер:** RANGE‑детектор; качественные границы (≥2–3 теста, конфлюэнс с VA/VWAP/H4 свинг), IB не чрезмерно широкий.
* **Триггер/стоп/тейки:** как в (7).
* **Подтв.:** CVD‑дивергенция, imbalance flip, абсорбция‑прокси, OI не разгоняется.
* **WR:** ~48–58% → **55–65%** → **60–70%** (в рейндже).

### 9) Профиль объёма: VAH/VPOC/VAL (рамка + сигналы)

* **Идея:** у края value различаем **rejection (fade)** vs **acceptance (продолжение)**.
* **Данные:** aggTrades (профиль), klines, WS depth, OI.
* **Rejection:** close обратно в value, POC не сдвигается; CVD‑flip; imbalance flip; OI не растёт.
* **Acceptance:** ≥2 close за VA **или** ≥0.25 ATR; объём/POC смещаются; CVD/OI по выходу.
* **Триггеры/стоп/тейки:** fade/acceptance по правилам (7)/(1).
* **WR:** fade **55–65%**, acceptance **50–60%**; комбинированный переключатель **58–68%**.

### 10) RSI/Стохастик mean‑reversion

* **Идея:** адаптивные пороги перекуп/перепрод.
* **Данные:** klines; (опц.) aggTrades, WS depth.
* **Сканер:** RANGE‑дни; RSI пороги по перцентилям (p15/p85 на 90d), стохастик крест обратно; зона = край рейнджа ∩ VWAP/VA.
* **Триггер/стоп/тейки:** как в (7).
* **Подтв.:** CVD‑дивергенция, imbalance flip.
* **WR:** ~45–55% → **52–62%** → **58–68%** (в рейндже с конфлюэнсом).

### 11) Ликвидностный свейп (stop‑hunt)

* **Идея:** укол за хай/лоу и быстрый reclaim → fade; при acceptance — продолжение.
* **Данные:** klines, aggTrades, WS depth, OI/ликвидации.
* **Сканер:** прокол ≥0.1–0.3 ATR или ≥0.1–0.2%; объём свейпа >1.5–2×; режим по детектору.
* **Триггеры:**
  + **Fade:** reclaim внутрь в том же/след. баре + CVD flip + imbalance flip.
  + **Continuation:** acceptance (2 close/0.25 ATR) + объём/POC‑сдвиг + OI↑.
* **Стоп/тейки:** за хвост свейпа +0.25 ATR; TP как в (7)/(1).
* **WR:** ~48–58% → **55–65%** (fade) / **52–62%** (cont.) → **60–70%** (переключатель).

### 12) Ордер‑флоу/имбэланс/абсорбция (прокси)

* **Идея:** торговать только когда **поток** подтверждает уровень.
* **Данные:** WS depth/diffDepth (imbalance, refill), aggTrades (серии агрессора), klines (close).
* **Сканер:** уровень (VA/VWAP/свинг), режим соответствует.
* **Триггер:** только вместе с **ценовым подтверждением** (reclaim/acceptance).
* **Стоп/тейки:** по структуре.
* **WR:** голые прокси 48–58%; с уровнем+ценой **58–68%**.

### 13) CVD / дивергенции

* **Идея:** развороты/подтверждения по балансу агрессоров.
* **Данные:** aggTrades (точнее) или барная CVD (klines).
* **Сканер:** у уровня/в режиме; для MR — дивергенция; для пробоя — подтверждение.
* **Триггер:** как в (11)/(1).
* **WR:** ~48–60% → **55–65%** → **58–68%** (с depth/OI/ценой).

### 14) Time‑of‑Day (временные паттерны)

* **Идея:** сессии EU/US дают лучшие импульсы.
* **Данные:** klines (слоты), OI/CVD по слотам.
* **Сканер:** выбираем 1–2 «жирных» окна для пробоев/моментума и 1–2 «тонких» для MR.
* **Триггер:** только в выбранных слотах при объёме>1.5× и воле выше медианы.
* **WR:** голые 48–58% → **55–65%** с фильтрами.

### 17) «Пила‑режим» (range/chop): детектор и правила

* **Идея:** торговать MR‑набор только когда рынок балансит.
* **Детектор:** см. §3 RANGE/CHOP (3 из 5 условий).
* **Правила:** зоны VA/VWAP; сигнал = свейп/отклонение + CVD‑дивергенция + imbalance flip; вход 50/50; стоп за хвост +0.25 ATR; тайм‑стоп 6–8 баров.
* **WR:** **60–70%** в истинный рейндж; <40% при ошибке режима.

### 19) Базис / Cash‑and‑Carry (дельта‑нейтрально)

* **Идея:** лонг спот + шорт перп (при +фандинге) или наоборот (при −фандинге); либо спот↔квартальник (contango/backwardation).
* **Данные:** premiumIndex, fundingRate, mark/index, спот‑цены; (для шорта спота — ставки маржи из Margin API).
* **Пороги:** 8h‑фандинг ≥ 3–5 б.п. **после** комиссий/заёма; по квартальнику APR ≥ 6–10% после издержек.
* **Исполнение:** maker, номинал 1:1 к индексу, кросс‑маржа с буфером 3–5×, авто‑пауза при ухудшении.
* **WR:** **95–100% дней в плюс** при дисциплине; доходность умеренная.

### 26) Маркет‑мейкинг / DOM‑скальпинг (исполнение‑зависимо)

* **Идея:** квотирование вокруг mid с анти‑токсик фильтрами.
* **Данные:** WS depth/diffDepth (≥10Hz), bookTicker, aggTrades, (опц.) allForceOrders, klines(1s–1m).
* **Не квотим:** спред < 2×fees; σ(1–5с) выше порога; BTC импульс; всплеск ликвидаций.
* **Микро‑альфа:** OFI, устойчивый depth‑imbalance, time‑at‑best, elasticity.
* **Очередь:** post‑only, мелкие лоты на 2–3 ценах, quote‑lifetime, оценка «позиции в очереди».
* **Риск:** инвентарь |Q|≤q\_max, hard‑стоп −2…−3 тика, time‑стоп 1–2s, tail‑стопы.
* **WR:** наивные 48–58%; с продвинутыми фильтрами **60–70%**, но хвосты/латентность критичны.

| ## 6) Комбинирование стратегий (готовые «комбо») - **2 + объём + CVD + ΔOI + depth** → пробой **55–65% WR**. - **5 + абсорбция + CVD flip + OI flat/↑** → ретест **56–66% WR**. - **7 + 17 + 9 + CVD див + imbalance flip** → MR‑рейндж **60–70% WR**. - **11 переключатель (fade vs acceptance) + объём + ΔOI** → **60–70% WR**. - **3 + 14 + BTC‑фильтр + объём** → ORB‑окна **55–65% WR**. Правила: не дублировать входы; «одна идея — один трейд»; скоринг ≥ +2; неттинг по символу. |
| --- |
| ## 7) Управление риском, комиссии и исполнение - **Единица риска R**; риск/сделку ≤ 0.5–1.0R; дневной/недельный стоп; макс. одновременных позиций. - **Размер позиции** от дистанции до стопа (ATR‑норм), учёт exchangeInfo (tick/step/minNotional). - **Комиссии/ребейты:** разделять maker/taker; добиваться maker‑tier; BNB‑скидки. - **Проскальзывание:** моделировать консервативно, особенно на альтах. - **Исполнение:** OCO/стоп‑маркеты, reduce‑only, post‑only. - **Kill‑switch:** рассинхрон книги, σ(1–5с)↑, всплеск ликвидаций, latency/API‑ошибки, аномальный premium/funding. |

## 8) Бэктест/валидация и контроль дрейфа

* **Walk‑forward**, purged k‑fold с «эмбарго» по времени; out‑of‑sample ≥ 3–6 мес.
* **Монте‑Карло**: перестановки сделок, стресс по комиссиям/скольжению.
* **Метрики:** expectancy (R/сделка), hit‑rate, avg R, max DD, Ulcer, серийность убыточных, время в рынке.
* **Дрейф:** перецентровка порогов в перцентилях еженедельно/ежемесячно; CUSUM/ADWIN на WR/EV; авто‑переключение в «flat‑режим».

| ## 9) Конфигурация (пример YAML) yaml symbols: ["BTCUSDT", "ETHUSDT"] timeframes: ["1m", "5m", "15m", "1h", "4h", "1d"] regime: adx\_low: 20 atrp\_p40\_lookback\_days: 60 bbw\_p30\_lookback\_bars: 90 vwap\_band\_pct: 0.60 # 60% баров внутри ±1σ late\_trend\_atr\_to\_ema20: 1.8 scoring: volume\_mult: 1.5 doi\_min\_pct: 1.0 depth\_imbalance\_ratio: long\_max: 0.90 short\_min: 1.10 btc\_filter\_tf: "1h" enter\_threshold: 2 risk: risk\_per\_trade\_R: 0.75 daily\_stop\_R: 3.0 weekly\_stop\_R: 7.0 execution: use\_post\_only: true oco\_stops: true time\_stop\_bars: 8 atr\_buffer\_stop: 0.25 partial\_tp\_R: [1.0, 2.5] cooldown\_bars\_same\_context: 10 carry: min\_funding\_bp\_8h\_net: 3 min\_calendar\_apr\_net: 0.10 profile: price\_bin\_pct: 0.001 # 0.1% composite\_days: 3 ws: depth\_levels: 20 imbalance\_window\_sec: 20 |
| --- |
| ## 10) Чек‑листы разработчика **Данные:** снапшот+diffDepth, ресинк при пропуске; дедуп aggTrades; UTC‑нормализация. **Фичи:** перцентильные пороги по паре/ТФ; пере‑калибровка weekly. **Режим:** решение по K‑из‑M признакам, гистерезис 2 бара; BTC‑фильтр. **Селектор:** приоритеты, скоринг ≥ +2, неттинг и cooldown. **Стратегии:** реализовать ровно по ТЗ выше; добавки только по (5). **Риск:** R‑блоки, дневной/недельный стоп, учёт комиссий/скольжения. **Операционка:** алерты (WS‑гэпы, latency, forceOrders всплески); kill‑switch. **Логи:** кто/когда/почему вошёл/вышел; режим/скор/флаги подтверждений. **Тесты:** walk‑forward, Монте‑Карло; отчёт по метрикам и серийности; контроль дрейфа. |

## Примечание по «прокси»

Все прокси (CVD, imbalance, refill, профиль VAH/VAL/VPOC) строятся **на стороне бота** из публичных REST/WS Binance и бесплатны. MBO/L3 недоступен на Binance; при необходимости L3 см. Coinbase full или платные провайдеры (вне рамок этого ТЗ).

| ### Итог Этот документ — **готовое ТЗ**: какой поток собрать, какие признаки вычислить, как распознать режимы, как и когда включать каждую стратегию, по каким правилам вход/стоп/тейк/менеджмент, как комбинировать, как тестировать и эксплуатировать. Реализация — Python (asyncio) + Binance REST/WS, с упором на качество данных и дисциплину исполнения. |
| --- |
| ## 11) Тестовые чек-листы и юнит‑тесты для фич |
| Ниже — практический пакет тестов/чек‑листов для разработчика. Покрывает загрузку данных, расчёт ключевых признаков (ATR/ADX/VWAP/CVD/профиль/imbalance), детектор режимов, сигналы стратегий, скоринг/гейтинг и «предохранители». Примеры на **pytest** (Python 3.11+). Для property‑based — **hypothesis**. |
| ### 11.1 Чек‑лист Data & WS‑инженерии - [ ] Все таймстэмпы приводятся к **UTC**; различие между REST и WS ≤ 1 бар рабочего ТФ. - [ ] depth снапшот → корректная инициализация локальной книги (levels ≥ 100). - [ ] diffDepth применяется **строго по sequence id**; пропуски детектятся и приводят к мгновенной ресинхронизации. - [ ] Дедупликация aggTrades по a/f/l id, отсутствие дыр в последовательности. - [ ] Повторный запуск клиента не приводит к «двойному счёту» (идемпотентность загрузки). - [ ] Защитные алерты: WS gap, рост latency, пустые окна данных, HTTP429/418. |
| ### 11.2 Юнит‑тесты расчётов (pytest) ```python import numpy as np import pandas as pd import pytest from hypothesis import given, strategies as st |
| # === Фикстуры синтетики === @pytest.fixture def klines\_df(): # синтетика 1m: тренд + шум n = 2000 rng = np.random.default\_rng(42) prices = 100 + np.cumsum(rng.normal(0, 0.1, n)) high = prices + rng.uniform(0.0, 0.15, n) low = prices - rng.uniform(0.0, 0.15, n) open\_ = np.r\_[prices[0], prices[:-1]] close = prices vol = rng.uniform(50, 150, n) taker\_buy = vol \* rng.uniform(0.3, 0.7, n) ts = pd.date\_range(“2024-01-01”, periods=n, freq=“1min”, tz=“UTC”) return pd.DataFrame({“open”:open\_,“high”:high,“low”:low,“close”:close, “volume”:vol,“takerBuyBaseAssetVolume”:taker\_buy}, index=ts) |
| # === Примитивы расчёта (примерные; заменить на продовые) === def atr(df, n=14): tr = np.maximum(df.high-df.low, np.maximum(abs(df.high-df.close.shift()), abs(df.low-df.close.shift()))) return tr.ewm(span=n, adjust=False).mean() |
| def adx(df, n=14): up = df.high.diff(); dn = -df.low.diff() plus\_dm = np.where((up>dn)&(up>0), up, 0.0) minus\_dm = np.where((dn>up)&(dn>0), dn, 0.0) tr = np.maximum(df.high-df.low, np.maximum(abs(df.high-df.close.shift()), abs(df.low-df.close.shift()))) atr\_ = pd.Series(tr, index=df.index).ewm(span=n, adjust=False).mean() pdi = pd.Series(plus\_dm, index=df.index).ewm(span=n, adjust=False).mean()/atr\_ \* 100 mdi = pd.Series(minus\_dm, index=df.index).ewm(span=n, adjust=False).mean()/atr\_ \* 100 dx = (abs(pdi-mdi)/(pdi+mdi))\*100 return dx.ewm(span=n, adjust=False).mean() |
| def vwap\_from\_trades(trades: pd.DataFrame): # trades: columns [price, qty, ts] num = (trades.price\*trades.qty).sum() den = trades.qty.sum() return num/den if den>0 else np.nan |
| # === Тесты === |
| def test\_atr\_positive\_monotonicity(klines\_df): a = atr(klines\_df, 14) assert (a.dropna()>0).all() a7 = atr(klines\_df, 7); a28 = atr(klines\_df, 28) # более длинное окно сглаживает: дисперсия ниже assert a28.dropna().std() < a7.dropna().std() |
| def test\_adx\_bounds(klines\_df): a = adx(klines\_df, 14).dropna() assert (a>=0).all() and (a<=100).all() |
| def test\_vwap\_equivalence\_to\_weighted\_mean(): prices = pd.Series([100, 101, 99, 100.5]) qty = pd.Series([1, 2, 3, 4]) df = pd.DataFrame({“price”:prices, “qty”:qty}) v = vwap\_from\_trades(df) w = np.average(prices, weights=qty) assert abs(v-w) < 1e-9 |
| def test\_cvd\_from\_klines\_matches\_sign\_of\_takers(klines\_df): # барная CVD: возрастание при доминировании takerBuy cvd = (klines\_df[“takerBuyBaseAssetVolume”] - (klines\_df[“volume”]-klines\_df[“takerBuyBaseAssetVolume”])) cum = cvd.cumsum() assert cum.iloc[-1] == pytest.approx(cvd.sum()) |
| def test\_profile\_binning\_conservation(): # проверяем, что сумма объёма по бинам = сумме исходного prices = np.array([100,100,100.1,100.2,100.21,100.19]) qty = np.array([1,2,3,2,1,4]) df = pd.DataFrame({“price”:prices, “qty”:qty}) bin\_size = 0.1 bins = np.floor((df.price - df.price.min())/bin\_size).astype(int) vol\_by\_bin = df.groupby(bins)[“qty”].sum().sum() assert vol\_by\_bin == df.qty.sum() |
| @given(st.lists(st.floats(min\_value=80, max\_value=120), min\_size=5, max\_size=30), st.lists(st.floats(min\_value=0.01, max\_value=5.0), min\_size=5, max\_size=30)) def test\_vwap\_invariant\_to\_trade\_order(prices, qty): assume = pytest.importorskip(“hypothesis”).assume assume(len(prices)==len(qty)) df = pd.DataFrame({“price”:prices, “qty”:qty}) v1 = vwap\_from\_trades(df) df\_shuffled = df.sample(frac=1.0, random\_state=7) v2 = vwap\_from\_trades(df\_shuffled) assert np.isfinite(v1) and np.isfinite(v2) assert abs(v1-v2) < 1e-8 ``` |
| ### 11.3 Depth/imbalance/рефилл — симуляционные тесты ```python from collections import deque |
| class LocalBook: def **init**(self): self.bids = {} # price -> size self.asks = {} def snapshot(self, bids, asks): self.bids = {float(p):float(s) for p,s in bids} self.asks = {float(p):float(s) for p,s in asks} def apply\_diff(self, bids\_upd, asks\_upd): for p,s in bids\_upd: p=float(p); s=float(s) if s==0: self.bids.pop(p, None) else: self.bids[p]=s for p,s in asks\_upd: p=float(p); s=float(s) if s==0: self.asks.pop(p, None) else: self.asks[p]=s def imbalance(self, k=10): b = sum(s for *,s in sorted(self.bids.items(), reverse=True)[:k]) a = sum(s for* ,s in sorted(self.asks.items())[:k]) return a/max(b,1e-9) |
| def test\_depth\_imbalance\_bounds(): lb = LocalBook() lb.snapshot(bids=[(99.9, 10),(99.8,5)], asks=[(100.1, 8),(100.2,6)]) r = lb.imbalance(2) assert r>0 |
| def test\_refill\_proxy\_after\_aggressive\_hits(): lb = LocalBook() lb.snapshot(bids=[(99.9, 10)], asks=[(100.1, 10)]) # имитируем серию ударов по bid (sell-taker), size уходит, затем refill lb.apply\_diff(bids\_upd=[(99.9, 2)], asks\_upd=[]) depleted = lb.imbalance(1) lb.apply\_diff(bids\_upd=[(99.9, 12)], asks\_upd=[]) refilled = lb.imbalance(1) assert refilled < depleted # больше bid → отношение ask/bid падает ``` |
| ### 11.4 Детектор режимов — тест‑наборы ```python from enum import Enum |
| class Regime(Enum): RANGE=1; TREND\_UP=2; TREND\_DOWN=3; SQUEEZE=4; EXPANSION=5 |
| # Заглушки функций: заменить на продовые |
| def detect\_regime(features) -> Regime: # features: dict с ADX, ATRp, BBw, ema\_slope, pct\_in\_vwap\_band, dOI, cvd\_dir, breakout # Простейшая логика для тестов if features[‘breakout’] and features[‘dOI’]>0 and features[‘cvd\_dir’]>0: return Regime.EXPANSION if features[‘ADX’]<20 and features[‘ATRp’]<features[‘ATRp\_p40’] and features[‘BBw’]<features[‘BBw\_p30’] and features[‘pct\_in\_vwap\_band’]>=0.6: return Regime.RANGE if features[‘ADX’]>20 and features[‘ema\_slope’]>0: return Regime.TREND\_UP if features[‘ADX’]>20 and features[‘ema\_slope’]<0: return Regime.TREND\_DOWN if features[‘BBw’]<features[‘BBw\_p20’] and features[‘ATRp’]<features[‘ATRp\_p30’]: return Regime.SQUEEZE return Regime.RANGE |
| def test\_regime\_range\_case(): f = dict(ADX=15, ATRp=0.008, ATRp\_p40=0.01, BBw=0.02, BBw\_p30=0.03, pct\_in\_vwap\_band=0.7, ema\_slope=0, dOI=0, cvd\_dir=0, breakout=False, BBw\_p20=0.02, ATRp\_p30=0.008) assert detect\_regime(f) == Regime.RANGE |
| def test\_regime\_trend\_up\_case(): f = dict(ADX=25, ATRp=0.012, ATRp\_p40=0.01, BBw=0.04, BBw\_p30=0.03, pct\_in\_vwap\_band=0.3, ema\_slope=+1, dOI=0.02, cvd\_dir=+1, breakout=False, BBw\_p20=0.02, ATRp\_p30=0.008) assert detect\_regime(f) == Regime.TREND\_UP |
| def test\_regime\_expansion\_case(): f = dict(ADX=22, ATRp=0.015, ATRp\_p40=0.01, BBw=0.05, BBw\_p30=0.03, pct\_in\_vwap\_band=0.2, ema\_slope=+1, dOI=0.03, cvd\_dir=+1, breakout=True, BBw\_p20=0.02, ATRp\_p30=0.008) assert detect\_regime(f) == Regime.EXPANSION ``` |
| ### 11.5 Стратегические триггеры — тесты логики ```python # Примеры логики триггеров (упрощённые) |
| def is\_breakout(close\_change\_atr, closes\_outside\_n, vol\_mult): return (close\_change\_atr>=0.25 or closes\_outside\_n>=2) and vol\_mult>=1.5 |
| def is\_reclaim(reclaim\_bar\_close\_inside, cvd\_flip, imbalance\_flip): return reclaim\_bar\_close\_inside and cvd\_flip and imbalance\_flip |
| def test\_breakout\_trigger(): assert is\_breakout(0.3, 0, 1.6) assert is\_breakout(0.1, 2, 2.0) assert not is\_breakout(0.1, 1, 2.0) |
| def test\_reclaim\_trigger(): assert is\_reclaim(True, True, True) assert not is\_reclaim(True, False, True) ``` |
| ### 11.6 Скоринг/гейтинг и неттинг ```python class Selector: def **init**(self, threshold=2): self.threshold = threshold self.active\_contexts = set() def score(self, volume\_mult, cvd\_sig, doi\_pct, imb\_ok, late\_trend, btc\_agree): s = 0 if volume\_mult>=1.5: s += 1 if cvd\_sig: s += 1 if doi\_pct>=1.0: s += 1 if imb\_ok: s += 1 if late\_trend: s -= 1 if not btc\_agree: s -= 2 return s def allow(self, ctx\_hash, s): if s < self.threshold: return False if ctx\_hash in self.active\_contexts: return False self.active\_contexts.add(ctx\_hash) return True |
| def test\_selector\_threshold\_and\_dedup(): sel = Selector(threshold=2) s1 = sel.score(1.6, True, 1.5, True, False, True) assert s1 >= 2 assert sel.allow((“BTCUSDT”,“zone1”,“long”), s1) # дубликат контекста запрещён assert not sel.allow((“BTCUSDT”,“zone1”,“long”), s1) ``` |
| ### 11.7 Риск/исполнение и «предохранители» ```python class KillSwitch: def **init**(self, max\_sigma, max\_latency\_ms): self.max\_sigma = max\_sigma self.max\_latency\_ms = max\_latency\_ms def check(self, sigma\_now, latency\_ms, ws\_gap, premium\_z): return (sigma\_now>self.max\_sigma) or (latency\_ms>self.max\_latency\_ms) or ws\_gap or (abs(premium\_z)>3.0) |
| def test\_kill\_switch\_conditions(): ks = KillSwitch(max\_sigma=3.0, max\_latency\_ms=500) assert ks.check(sigma\_now=3.5, latency\_ms=100, ws\_gap=False, premium\_z=0.5) assert ks.check(sigma\_now=1.0, latency\_ms=700, ws\_gap=False, premium\_z=0.5) assert ks.check(sigma\_now=1.0, latency\_ms=100, ws\_gap=True, premium\_z=0.5) assert ks.check(sigma\_now=1.0, latency\_ms=100, ws\_gap=False, premium\_z=3.5) ``` |
| ### 11.8 Покрытие стратегий (таблица соответствия тестов) - (1) Donchian: тесты is\_breakout, ADX/ATR-percents, ΔOI/CVD подтверждение. - (2) Squeeze→Breakout: BBw перцентили, тест is\_breakout, late‑trend фильтр. - (3) ORB: IB расчёт, окно времени, is\_breakout. - (4) Откат к MA/AVWAP: расстояние до EMA20/AVWAP, свеча‑отклонение, is\_reclaim. - (5) Break&Retest: acceptance‑пробой + ретест‑зона + is\_reclaim. - (6) ATR‑моментум: импульс‑бар≥1.4×ATR, follow‑through, тайм‑стоп 6–8 баров. - (7,8,10) MR‑набор: RANGE‑детектор, VWAP/Value зоны, CVD‑див, is\_reclaim. - (9) Профиль: VAH/VAL/VPOC корректность, rejection vs acceptance. - (11) Свейп: прокол по ATR, is\_reclaim/is\_breakout свитчер, ликвидации. - (12,13) Потоки: depth‑imbalance устойчивость, CVD‑див/confirm. - (14) Время: слоты, объём/вола пороги. - (17) «Пила»: 3 из 5 критериев, BTC‑фильтр. - (19) Carry: расчёт net‑EV на окне funding/базиса, пороги входа/ролла. - (26) MM/DOM: спред/fees, OFI, queue‑lifetime, tail‑стопы. |
| ### 11.9 Как запускать тесты - Зависимости: pytest, hypothesis, pandas, numpy. - Команда: pytest -q - Для property‑based тестов: pytest -q -k hypothesis - Интеграционные (с реальными REST/WS) держать отдельно и маркировать @pytest.mark.integration. |
| > Примечание: приведённый код — шаблоны. В проде расчёты (ATR/ADX/VWAP/CVD/профиль/imbalance) должны использовать ваш рабочий модуль; тесты адаптируйте под его API. |

## 12) KPI сигналов (для инженеров)

**Определения:**  
- **Latency (E2E)** — время от прихода последнего нужного тик/бара/книжного апдейта до публикации сигнала ботом.  
- **WR\_min** — минимально допустимый win‑rate по стратегии/семье на *валидации* и *онлайне (скользящее окно)*, ниже — авто‑деактивация.  
- **RR\_target** — целевой средний риск‑профит (R) на сделку; для MM — ожидаемая доходность в тиках/спредах после комиссий.  
- **FPR\_max** — доля «ложных срабатываний» (сигналов, не подтвердившихся по гейтингу/триггеру и снятых в коротком окне), считая по скользящему окну.

### 12.1 Таблица KPI по семьям

| Семья / стратегии | Сигнальное событие | Latency (E2E) целевая | WR\_min | RR\_target | FPR\_max |
| --- | --- | --- | --- | --- | --- |
| **Breakout & Momentum** (1,2,3,6) | close≥0.25ATR **или** 2×close за границей; ORB‑пробой; флаг‑пробой | **250–800 ms** от бар‑закрытия; для интра‑бара **150–300 ms** после условия | **≥52%** (с гейтингом) | **≥2.0R** (сред. 2–3R) | **≤15%** |
| **Pullback / Retest** (4,5) | свеча‑отклонение + reclaim/retest‑паттерн | **0.5–3 s** (ждём закрытие сигнальной свечи или подтверждение) | **≥56%** | **1.5–2.5R** | **≤12%** |
| **Range MR** (7,8,10, 11‑fade) | reclaim внутрь value/VWAP; MR‑триггер | **0.5–2 s** | **≥58%** (только в RANGE) | **1.2–2.0R** | **≤15%** |
| **Acceptance/Continuation** (9‑accept, 11‑cont) | acceptance: 2×close вне **или** ≥0.25ATR + объём/OI | **200–500 ms** от подтверждения | **≥52%** | **2.0–3.0R** | **≤15%** |
| **Flow‑подтверждения** (12,13) | CVD flip/confirm; устойчивый depth‑imbalance | **100–200 ms** обновления скоринга | n/a (фильтр) | n/a | **≤10%** «мелькающих» флип‑сигналов |
| **Time‑of‑Day гейтинг** (14) | вход/выход в слот | планово; SLA **< 1 s** от начала слота | n/a | n/a | **<1%** публикуемых вне слотов |
| **Cash‑and‑Carry** (19) | пересечение порога net‑EV | **≤60 s** (не критична мс‑латентность) | **≥95%** «дней в плюс» | n/a (таргет APR/день) | **≤1%** |
| **MM/DOM** (26) | обновление котировки/сигнала скальпа | **≤200 ms** от последнего depth/aggTrade; идеал **≤100 ms** | **≥58%** на «разрешённых» режимах | **+0.1…0.3 тика/roundtrip** после комиссий | **≤5%** токс. фьюз‑сигналов |

Примечания: • Для ORB/временных окон — latency меряется внутри окна относительно момента пробоя границы IB.  
• Для MR — сигнал публикуется **после** факта reclaim (не «в нож»).  
• Для Flow (12/13) — это **не самостоятельные входы**, а апдейтер скоринга; FPR — доля коротких «фальш‑флипов» < N секунд.

### 12.2 SLO по цепочке обработки (реактивные сигналы)

* **WS приём** (depth/aggTrade): p95 **< 50 ms** от реального времени.
* **Feature compute** (imbalance/CVD/VWAP обновление): p95 **< 50 ms**.
* **Rule eval + скоринг**: p95 **< 20 ms**.
* **Публикация сигнала → исполнение**: p95 **< 50 ms** (отправка в шину/модуль ордеров).
* **Итого бюджет**: p95 **≤ 150–200 ms** для реактивных (DOM/acceptance), **≤ 250–800 ms** для close‑зависимых.

### 12.3 Мониторинг качества (онлайн‑гейты)

* **WR\_guard**: rolling 200 сделок по семье/символу **< WR\_min − 3 п.п.** → авто‑деактивация семьи на символе, алерт.
* **FPR\_guard**: rolling 500 сигналов **> FPR\_max + 5 п.п.** → ужесточить пороги (score↑ на +1) или отключить саб‑паттерн.
* **Latency\_guard**: p95 по E2E **> целевого ×1.5** в течение 15 мин → включить деградационный режим (меньше стратегий, только close‑бэйзд), алерт DevOps.
* **Drift\_guard**: падение expectancy/SQN на бэктестовых окнах — авто‑перекалибровка перцентилей, перевод в RANGE‑режим по умолчанию.

### 12.4 Определения «ложного срабатывания» (для метрик)

* **Breakout‑false:** опубликован сигнал пробоя, но в окне **N=3 бара** не было ни подтверждения (follow‑through), ни ретеста‑входа — помечаем как FP.
* **MR‑false:** опубликован сигнал MR, но **не было reclaim** или цена прошла **≥0.25 ATR против** без триггера входа.
* **Flow‑false:** flip CVD/imbalance исчез **< 10–20 с**, не дав подтверждения цены.
* **MM‑false:** котировка эшелона привела к fill при токсичном микродрейфе (mid ушёл ≥2 тика против за ≤500 мс).

### 12.5 Как логировать KPI (минимум)

* t\_signal\_in (последний нужный апдейт данных), t\_signal\_out (публикация), t\_order\_in (принято исполнителем).
* Поля контекста: стратегия, семья, режим, score, признаки (объём/CVD/ΔOI/imbalance), дист. до EMA20 (late‑trend), BTC‑байас.
* Для FPR: статус «confirmed | retracted | timeout».
* Для MM: queue\_pos\_proxy, time\_at\_best, slippage\_ticks.