uint256 size; 仓位价值(USD)
uint256 collateral; 抵押物品价值(USD)
uint256 averagePrice; 均价(USD)
uint256 entryFundingRate; 入场资金费率
uint256 reserveAmount; 储备金额(token number) int256 realisedPnl; 已实现盈亏(USD)
uint256 lastIncreasedTime; 最后一次加仓时间 (sposition getMinPrice markPrice > = _price getMaxPrice tx.gasprice <=maxGasPrice msg.sender == router 更新累加资金费率 仓位价值(USD) 抵押代币价值(USD) uint256 size; uint256 collateral; uint256 averagePrice; 均价(USD) uint256 entryFundingRate; 入场资金费率 uint256 reserveAmount; 储备金额 (代币数量) int256 realisedPnl; 记实现盈亏(USD) uint256 lastIncreasedTime; 最后一次加仓时间 markPrice <= maxGasPrice pos.size > sizeDelta pos.collateral > = _collateralDelta pos.reserveAmount = (sizeDelta / size) * pos.reserveAmount 并更新全局reserve 减仓费用: 减少仓位 * 杠杆费用(0.1%)、资金费用(之前仓位USD * 当前资金费率) 获取 (bool _hasProfit, uint256 delta) fetDelta isLong == true getMinPrice getMaxPrice price priceDelta = _averagePrice > price ? _averagePrice.sub(price) : price.sub(_averagePrice); uint256 delta = _size.mul(priceDelta).div(_averagePrice); 整体盈亏 isLong == true → 杏 hasProfit = price < _averagePrice hasProfit minBps = (减仓时间 > 最后一次加仓时间 + minProfitTime) ? 0 : minProfitBasisPoints[indexToken] hasProfit && delta.mul(BASIS_POINTS_DIVISOR) <= _size.mul(minBps)) adjustedDelta = _sizeDelta.mul(delta).div(position.size); (减少仓位) 部分的盈亏 空头就需要减少或增加对应代币池的数量 usdOut = adjustedDelta; _decreasePoolAmount(_collateralToken, tokenAmount); hasProfit && adjustedDelta > 0 pos.realisedPnl +=adjustedDelta pos.collateral -= adjustedDelta; ←亏损——— !hasProfit && adjustedDelta > 0 _increasePoolAmount(_collateralToken, tokenAmount); isLong == true position.realisedPnl -= int256(adjustedDelta); pos.collateral = pos.collateral - _collateralDelta; usdOut = usdOut.add(position.collateral); position.size == _sizeDelta position.collateral = 0; 亏损价值 = 抵押品价值 亏损价值 = 仓位价值 * 比例 收益 - 手续费(杠杆费+资金费), 如果亏损, 则用抵押 品价值扣除 比例 = priceDelta / averagePrice priceDelata = price - averagePrice - price price = priceDelata + averagePrice