Perp v2 important variables

## ClearingHouseConfig.sol / ClearingHouseConfigStorage.sol

**uint8 internal \_maxMarketsPerAccount;**

#### \_maxMarketsPerAccount:

Determines maximum opened markets per one account. Can be changed by the contract owner and default value is 255.

## AccountBalance.sol / AccountBalanceStorage.sol

*// trader => owedRealizedPnl*

mapping(address => int256) internal \_owedRealizedPnlMap;

*// trader => baseTokens*

mapping(address => address[]) internal \_baseTokensMap;

*// first key: trader, second key: baseToken*

mapping(address => mapping(address => AccountMarket.Info)) internal \_accountMarketMap;

#### \_baseTokenMap:

Base token registry of each trader. Base token address is pushed to the array before opening first position and removed after closing position. Maximum array length is defined in ClearingHouseConfigStorage.\_maxMarketsPerAccount

#### \_accountMarketMap:

Stores information about trader’s opened markets. Can be updated via updateTwPremiumGrowthGlobal function.

## BaseToken.sol / BaseTokenStorage.sol

IBaseToken.Status internal \_status;

address internal \_priceFeed;

uint256 constant MAX\_WAITING\_PERIOD = 5 days;

#### \_status:

Determines if traders can open/close positions. Owner can pause or stop trading on base tokens. If trading is paused more than MAX\_WAITING\_PERIOD, anyone can call the stop function. Once trading is closed, there is no way to open again.  
Values can be Open, Paused, Closed

#### \_priceFeed:

Stores address of Price feed contract. Is set during initialization and can be changed only by owner

#### MAX\_WAITING\_PERIOD:

Is constant, determines how long the trading can be paused before anyone can close.

## CollateralManager.sol / CollateralManagerStorage.sol

uint24 internal \_liquidationRatio;

uint8 internal \_maxCollateralTokensPerAccount;

#### \_liquidationRatio:

Determines maximum leverage that traders can use. When ratio is exceed, liquidator can liquidate position

#### \_maxCollateralTokensPerAccount:

Determines maximum collateral tokens for one account. Is set during initialization and can be changed anytime by owner

## MarketRegistry.sol / MarketRegistryStorage.sol

uint8 internal \_maxOrdersPerMarket;

*// key: baseToken, value: pool*

mapping(address => address) internal \_poolMap;

*// key: baseToken*

mapping(address => uint24) internal \_insuranceFundFeeRatioMap;

*// key: baseToken*

mapping(address => uint24) internal \_exchangeFeeRatioMap;

#### \_poolMap:

Stores liquidity pools addresses for base tokens

#### \_insuranceFundFeeRatioMap:

Determines what insurance fund get (exchangeFee \* insuranceFundFeeRatio). Fee ratio can be changed by contract owner

#### \_exchangeFeeRatio:

uniswap fee will be ignored and use the exchangeFeeRatio instead. Fee ratio is set during adding new pool and can be changed by contract owner

## OrderBook.sol / OrderBookStorage.sol

*// first key: trader, second key: base token*

mapping(address => mapping(address => bytes32[])) internal \_openOrderIdsMap;

*// key: openOrderId*

mapping(bytes32 => OpenOrder.Info) internal \_openOrderMap;

#### \_openOrderIdsMap:

Stores opened orders ids by trader and base token

#### \_openOderMap:

Stores all opened orders informations by order id stored in \_openOrderIdsMap

## Vault.sol / VaultStorage.sol

*// key: trader, token address*

mapping(address => mapping(address => int256)) internal \_balance;

*// trader => collateral token*

mapping(address => address[]) internal \_collateralTokensMap;

#### \_balance:

Stores traders balance for each collateral tokens

#### \_collateralTokensMap:

Stores trader’s collateral tokens. Maximum is defined in CollateralManagerStorage**.**\_maxCollateralTokensPerAccount