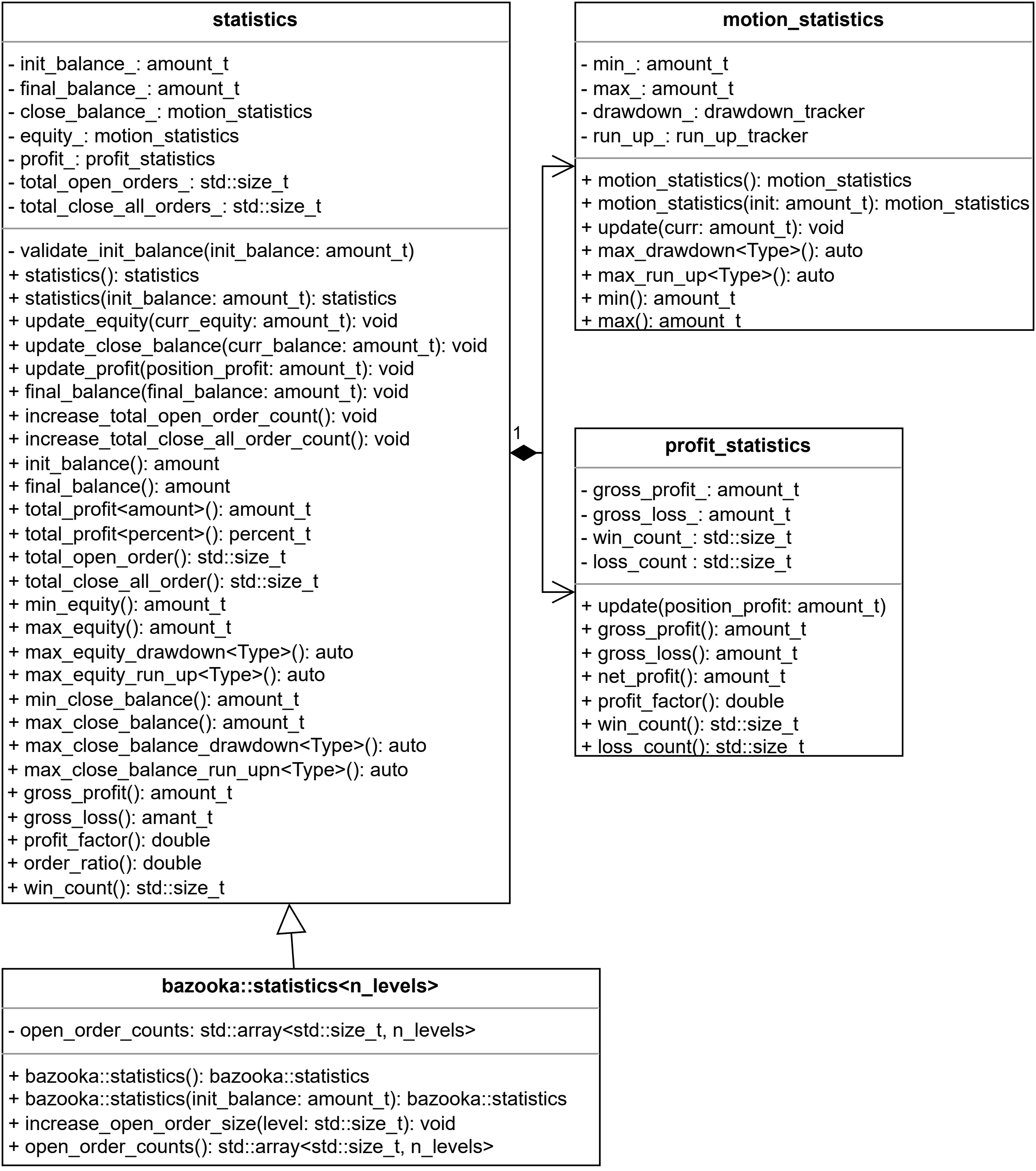
statistics
- init_balance_: amount_t
- final_balance_: amount_t
- close_balance_: motion_statistics
- equity_: motion_statistics
- profit_: profit_statistics
- total_open_orders_: std::size_t
- total_close_all_orders_: std::size_t
- validate_init_balance(init_balance: amount_t)
+ statistics(): statistics
+ statistics(init_balance: amount_t): statistics
+ update_equity(curr_equity: amount_t): void
+ update_close_balance(curr_balance: amount_t): void
+ update_profit(position_profit: amount_t): void
+ final_balance(final_balance: amount_t): void
+ increase_total_open_order_count(): void
+ increase_total_close_all_order_count(): void
+ init_balance(): amount
+ final_balance(): amount
+ total_profit<amount>(): amount_t
+ total_profit<percent>(): percent_t
+ total_open_order(): std::size_t
+ total_close_all_order(): std::size_t
+ min_equity(): amount_t
+ max_equity(): amount_t
+ max_equity_drawdown<Type>(): auto
+ max_equity_run_up<Type>(): auto
+ min_close_balance(): amount_t
+ max_close_balance(): amount_t
+ max_close_balance_drawdown<Type>(): auto
+ max_close_balance_run_upn<Type>(): auto
+ gross_profit(): amount_t
+ gross_loss(): amant_t
+ profit_factor(): double
+ order_ratio(): double
+ win_count(): std::size_t
motion_statistics
- min_: amount_t
- max_: amount_t
- drawdown_: drawdown_tracker
- run_up_: run_up_tracker
+ motion_statistics(): motion_statistics
+ motion_statistics(init: amount_t): motion_statistics
+ update(curr: amount_t): void
+ max_drawdown<Type>(): auto
+ max_run_up<Type>(): auto
+ min(): amount_t
+ max(): amount_t
1
profit_statistics
- gross_profit_: amount_t
- gross_loss_: amount_t
- win_count_: std::size_t
- loss_count : std::size_t
+ update(position_profit: amount_t)
+ gross_profit(): amount_t
+ gross_loss(): amount_t
+ net_profit(): amount_t
+ profit_factor(): double
+ win_count(): std::size_t
+ loss_count(): std::size_t
bazooka::statistics<n_levels>
- open_order_counts: std::array<std::size_t, n_levels>
+ bazooka::statistics(): bazooka::statistics
+ bazooka::statistics(init_balance: amount_t): bazooka::statistics
+ increase_open_order_size(level: std::size_t): void
+ open_order_counts(): std::array<std::size_t, n_levels>