```
#property copyright "Copyright 2022, MetaQuotes Ltd."
#property link      "https://www.mql5.com"
#property version   "1.00"
//+------------------------------------------------------------------+
//|Include                                                           |
//+------------------------------------------------------------------+
#include <Trade\Trade.mqh>
//+------------------------------------------------------------------+
//| Input Variables                                                  |
//+------------------------------------------------------------------+
input int InpFastPeriod = 5;
input int InpSlowPeriod = 10;
input int InpStopLoss = 100;
input int InpTakeProfit = 200;
//+------------------------------------------------------------------+
//| Global Variables                                                 |
//+------------------------------------------------------------------+
int fastHandle;
int slowHandle;
double fastBuffer[];
double slowBuffer[];
datetime openTimeBuy = 0;
datetime openTimeSell = 0;
CTrade trade;
//+------------------------------------------------------------------+
//| Expert initialization function                                   |
//+------------------------------------------------------------------+
int OnInit(){
if(InpFastPeriod <= 0)
```

```
{
  Alert("Fast Period <= 0");
  return INIT_PARAMETERS_INCORRECT;
}
if(InpSlowPeriod <= 0)
{
  Alert("Slow Period <= 0");
  return INIT_PARAMETERS_INCORRECT;
}
if(InpFastPeriod >= InpSlowPeriod)
{
  Alert("InpFastPeriod >= InpSlowPeriod");
  return INIT_PARAMETERS_INCORRECT;
}
if(InpStopLoss <= 0)
{
  Alert("Stop Loss <= 0");
  return INIT_PARAMETERS_INCORRECT;
}
if(InpTakeProfit <= 0)
{
  Alert("Take Profit <= 0");
  return INIT_PARAMETERS_INCORRECT;
}
//Create handles
  fastHandle = iMA(_Symbol, PERIOD_CURRENT, InpFastPeriod,0 , MODE_SMA, PRICE_CLOSE);
  if(fastHandle == INVALID_HANDLE){
  Alert("Failed to create fast handle");
  return INIT_FAILED;
```

```
  }

 slowHandle = iMA(_Symbol, PERIOD_CURRENT, InpSlowPeriod,0 , MODE_SMA, PRICE_CLOSE);
  if(slowHandle == INVALID_HANDLE){
  Alert("Failed to create slow handle");
  return INIT_FAILED;
  }

  ArraySetAsSeries(fastBuffer, true);
  ArraySetAsSeries(slowBuffer, true);

  return(INIT_SUCCEEDED);
 }



//+------------------------------------------------------------------+
//| Expert deinitialization function                                 |
//+------------------------------------------------------------------+
void OnDeinit(const int reason){
if(fastHandle != INVALID_HANDLE){
IndicatorRelease(fastHandle);
 }
 if(slowHandle != INVALID_HANDLE){
IndicatorRelease(slowHandle);
 }
}



//+------------------------------------------------------------------+
```

```
//| Expert tick function                             |
//+------------------------------------------------------------------+
void OnTick(){
int values = CopyBuffer(fastHandle, 0, 0, 2, fastBuffer);
if(values != 2)
{
Print("Not enough data for fast moving average");
return;
}
  values = CopyBuffer(slowHandle, 0, 0, 2, slowBuffer);
  if(values != 2)
  {
  Print("Not enough data for slow moving average");
  return;
  }
  Comment("fast[0]:", fastBuffer[0], "\n",
          "fast[1]:", fastBuffer[1], "\n",
          "slow[0]:", slowBuffer[0], "\n",
          "slow[1]:", slowBuffer[1], "\n");


          //Check for cross buy
          if(fastBuffer[1] <= slowBuffer[1] && fastBuffer[0] > slowBuffer[0] && openTimeBuy !=
iTime(_Symbol, PERIOD_CURRENT, 0)){
          openTimeBuy = iTime(_Symbol, PERIOD_CURRENT, 0);
          double ask = SymbolInfoDouble(_Symbol, SYMBOL_ASK);
          double sl = ask - InpStopLoss * SymbolInfoDouble(_Symbol, SYMBOL_POINT);
          double tp = ask + InpTakeProfit * SymbolInfoDouble(_Symbol, SYMBOL_POINT);


          trade.PositionOpen(_Symbol, ORDER_TYPE_BUY, 1.0, ask, sl, tp, "Cross EA");
```

```
        }
         //Check for cross sell
        if(fastBuffer[1] >= slowBuffer[1] && fastBuffer[0] < slowBuffer[0] && openTimeSell != iTime(_Symbol, PERIOD_CURRENT, 0)){
        openTimeSell = iTime(_Symbol, PERIOD_CURRENT, 0);
        double bid = SymbolInfoDouble(_Symbol, SYMBOL_BID);
         double sl = bid + InpStopLoss * SymbolInfoDouble(_Symbol, SYMBOL_POINT);
        double tp = bid - InpTakeProfit * SymbolInfoDouble(_Symbol, SYMBOL_POINT);
        trade.PositionOpen(_Symbol, ORDER_TYPE_SELL, 1.0, bid, sl, tp, "Cross EA");
        }
}
```