

```

property copyright "Copyright 2020, Zyuna32246"
property link      "http://zyuna32246.livedoor.blog/"
property version   "1.00"
#property strict

//---パラメーター設定
sinput int MagicNumber=2020;           //マジックナンバー
input int Slippage=10;                  //スリッページ
input double Lots=0.01;                 //ロット数
input int TakeProfit=100;                //利確幅(pips)
input int StopLoss=50;                  //損切幅(pips)
input int TrailingPips=20;              //トレーリングストップ(pips)
//---インジケーター設定
input int Period1=5;                   //期間1
input int Period2=25;                  //期間2
input int Period3=75;                  //期間3
//---プログラム全体に影響する変数
double Pips;
int SLP;
//-----
int OnInit() {                         //EA導入時に一度だけ動くプログラム
    Pips=AdjustPoint(_Symbol);
    SLP=(int)AdjustSlippage(_Symbol,Slippage);

    return(INIT_SUCCEEDED);
}
//-----
void OnTick() {                        //ティック受信する度に動くプログラム
    //トレーリングストップ
    TrailingStop();

    //ポジション数を確認
    int BuyPosition=Position(1);
    int SellPosition=Position(-1);

    //エントリー判定
    int Entry=PerfectOrder();
    if(BuyPosition==0 && Entry==1)OpenOrder(1);
    if(SellPosition==0 && Entry==-1)OpenOrder(-1);

    //クローズ判定
    if(BuyPosition!=0 && PerfectClose(1)==1)CloseOrder(1);
    if(SellPosition!=0 && PerfectClose(-1)==-1)CloseOrder(-1);

} //OnTick
//-----
void OnDeinit(const int reason) {       //EA削除時に最後に動くプログラム
}
//-----
//オーバーライド関数
//ポジション調整
double AdjustPoint(string Currency) {
    int SymbolDigits=(int)MarketInfo(Currency,MODE_DIGITS);
    double CalculatedPoint=0.0;
    if(SymbolDigits==2 || SymbolDigits==3) {
        CalculatedPoint=0.01;
    } else
    if(SymbolDigits==4 || SymbolDigits==5) {
        CalculatedPoint=0.0001;
    }
    return(CalculatedPoint);
}

//スリッページ調整
double AdjustSlippage(string Currency, int SlippagePips) {
    double CalculatedSlippage=0.0;
    int SymbolDigits=(int)MarketInfo(Currency,MODE_DIGITS);
    if(SymbolDigits==2 || SymbolDigits==3) {
        CalculatedSlippage=SlippagePips;
    } else
    if(SymbolDigits==4 || SymbolDigits==5) {
        CalculatedSlippage=SlippagePips*10;
    }
    return(CalculatedSlippage);
}

//ポジション数を取得
int Position(int PositionDirection) {
    int res=0;
    for(int i=0;i<OrdersTotal();i++) {
        if(OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==true) {
            if(OrderSymbol()==_Symbol && OrderMagicNumber()==MagicNumber) {
                if(PositionDirection==1) {
                    if(OrderType()==OP_BUY)res++;
                } else
                    if(PositionDirection==-1) {
                        if(OrderType()==OP_SELL)res++;
                    }
                }
            }
        }
    }
    return(res);
}

//ポジションエントリー関数
void OpenOrder(int EntryPosition) {
    int res=0;
    double TP=TakeProfit*Pips;

```

```

double SL=StopLoss*Pips;
if(EntryPosition==1){
    res=OrderSend(_Symbol,OP_BUY,Lots,Ask,SLP,Ask-SL,Ask+TP,NULL,MagicNumber,0,clrBlue);
}else
if(EntryPosition== -1){
    res=OrderSend(_Symbol,OP_SELL,Lots,Bid,SLP,Bid+SL,Bid-TP,NULL,MagicNumber,0,clrRed);
}
}

//ポジションクローズ関数
void CloseOrder(int ClosePosition){
    int res=0;
    for(int i=OrdersTotal()-1;i>=0;i--) {
        if(OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==true) {
            if(OrderSymbol()==_Symbol && OrderMagicNumber()==MagicNumber) {
                if(OrderType()==OP_BUY && ClosePosition==1) {
                    res=OrderClose(OrderTicket(),OrderLots(),OrderClosePrice(),SLP,clrMagenta);
                }else
                if(OrderType()==OP_SELL && ClosePosition== -1) {
                    res=OrderClose(OrderTicket(),OrderLots(),OrderClosePrice(),SLP,clrMagenta);
                }
            }
        }
    }
}

//トレーリングストップ
void TrailingStop(){
    double TSP;
    int res=0;
    for(int i=OrdersTotal()-1;i>=0;i--) {
        if(OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==true) {
            if(OrderSymbol()==_Symbol && OrderMagicNumber()==MagicNumber) {
                if(OrderType()==OP_BUY) {
                    TSP=Ask-TrailingPips*Pips;
                    if(OrderOpenPrice()<TSP && OrderStopLoss()<TSP) {
                        res=OrderModify(OrderTicket(),OrderOpenPrice(),TSP,0,0,clrYellow);
                    }
                }else
                if(OrderType()==OP_SELL) {
                    TSP=Bid+TrailingPips*Pips;
                    if(OrderOpenPrice()>TSP && OrderStopLoss()>TSP) {
                        res=OrderModify(OrderTicket(),OrderOpenPrice(),TSP,0,0,clrYellow);
                    }
                }
            }
        }
    }
}

//パーフェクトオーダーエントリー
int PerfectOrder(){
    int entry=0;
    double MA1=iMA(_Symbol,0,Period1,0,MODE_SMA,PRICE_CLOSE,1);
    double MA2=iMA(_Symbol,0,Period2,0,MODE_SMA,PRICE_CLOSE,1);
    double MA3=iMA(_Symbol,0,Period3,0,MODE_SMA,PRICE_CLOSE,1);

    if(MA3<MA2 && MA2<MA1)entry=1;
    if(MA3>MA2 && MA2>MA1)entry=-1;

    return(entry);
}

//パーフェクトオーダークローズ
int PerfectClose(int Position){
    int close=0;
    double MA1=iMA(_Symbol,0,Period1,0,MODE_SMA,PRICE_CLOSE,1);
    double MA2=iMA(_Symbol,0,Period2,0,MODE_SMA,PRICE_CLOSE,1);
    double MA3=iMA(_Symbol,0,Period3,0,MODE_SMA,PRICE_CLOSE,1);

    if(Position==1) {
        if(MA3>MA2 || MA2>MA1)close=1;
    }else
    if(Position== -1) {
        if(MA3<MA2 || MA2<MA1)close=-1;
    }

    return(close);
}

```