```
//Pin-Bar Sign.mq4

#property indicator_chart_window

#property indicator_buffers 2

#property indicator_color1 Magenta
#property indicator_color2 Aqua
```

#property命令を記述

```
//インジケーターバッファーの宣言
double Arrow_Up[];
double Arrow_Down[];
double Real_Body[];
double Upper_Shadow[];
double Lower_Shadow[];
```

インジケーターバッファーを宣言

```
//変数の宣言
extern int Highest_Period    = 20;
extern int Lowest_Period     = 20;
extern int Magnification     = 3;
extern int Minimum_Length = 30;

double Adjusted_Point = 0;
```

変数を宣言

```
//関数の定義
//ピップポイントの計算
double AdjustPoint(string Currency)
{
  int Calculated_Digits = MarketInfo(Symbol(),MODE_DIGITS);

  if(Calculated_Digits == 2 || Calculated_Digits == 3)
    {
      double Calculated_Point = 0.01;
    }
  else if(Calculated_Digits == 4 || Calculated_Digits == 5)
          {
            Calculated_Point = 0.0001;
          }

  return(Calculated_Point);
}
```

関数を定義

```
int init()
{
  IndicatorBuffers(5);

  //インジケーターバッファーのインデックス
  SetIndexBuffer(0,Arrow_Up);
  SetIndexBuffer(1,Arrow_Down);
  SetIndexBuffer(2,Real_Body);
  SetIndexBuffer(3,Upper_Shadow);
  SetIndexBuffer(4,Lower_Shadow);

  //インジケーターのラベル
  SetIndexLabel(0,NULL);
  SetIndexLabel(1,NULL);

  //インジケーターのスタイル
  SetIndexStyle(0,DRAW_ARROW);
  SetIndexArrow(0,233);
  SetIndexStyle(1,DRAW_ARROW);
  SetIndexArrow(1,234);

  //PointとPipsの調整
  Adjusted_Point = AdjustPoint(Symbol());

  return(0);
}
```

基本設定を記述

```
int start()
{
  int limit = Bars - IndicatorCounted();

  if(Bars < Highest_Period || Bars < Lowest_Period)
    {
      return(0);
    }

  //実体の計算
  for(int i = limit - 1; i >= 0; i--)
      {
        Real_Body[i] = MathAbs(Open[i] - Close[i]);

        if(Real_Body[i] == 0)
          {
            Real_Body[i] = Real_Body[i] + Adjusted_Point;
          }
      }

  //上ヒゲの計算
  for(i = limit - 1; i >= 0; i--)
      {
        Upper_Shadow[i] = MathMin(High[i] - Open[i],High[i] - Close[i]);
      }

  //下ヒゲの計算
  for(i = limit - 1; i >= 0; i--)
      {
        Lower_Shadow[i] = MathMin(Open[i] - Low[i],Close[i] - Low[i]);
      }

  //矢印の設定
  for(i = limit - 1; i >= 0; i--)
      {
        //上矢印の設定
        if(Real_Body[i] * Magnification <= Lower_Shadow[i] && Minimum_Length * Adjusted_Point <= Lower_Shadow[i] &&
           Low[i] < Low[iLowest(NULL,0,MODE_LOW,Lowest_Period,i+1)])
         {
           Arrow_Up[i] = Low[i] - Adjusted_Point;
         }

        //下矢印の設定
        if(Real_Body[i] * Magnification <= Upper_Shadow[i] && Minimum_Length * Adjusted_Point <= Upper_Shadow[i] &&
           High[i] > High[iHighest(NULL,0,MODE_HIGH,Highest_Period,i+1)])
         {
           Arrow_Down[i] = High[i] + Adjusted_Point;
         }
      }

  return(0);
}
```

具体的な処理内容
を記述