```
{====================================================================
 最小化重現範例：驗證 intraBarPersist 變數在 Filled 轉換前後
 是否會被重置。

 現象說明：
 1. 進場後，出場當下對一個 intraBarPersist 宣告的日期時間變數賦值。
 2. 賦值完成後立即 Print，確認賦值當下數值正確(非0)。
 3. 但緊接著在空手(Filled=0)狀態下的下一次運算，同一個變數卻讀到
    初始值(變成0)，且此後持續維持0，不會恢復。
 4. 已排除同時掛載多個腳本實例的可能性；一般variable宣告與
    intraBarPersist宣告皆出現同樣現象。

 使用方式：掛在任一商品的自動交易上，5分鐘週期，用簡單的均線
 交叉當進出場條件(不影響本次要驗證的重點)，實際跑一段模擬單，
 觀察「【出場當下確認】」與「【下次讀取】」兩行Print的數值是否
 一致。
====================================================================}

SetTotalBar(500);
SetBackBar(300);

variable: intraBarPersist LastExitDate(0);
variable: intraBarPersist LastExitTime(0);
variable: MA5(0), MA20(0);

MA5 = Average(Close, 5);
MA20 = Average(Close, 20);

// 空手時，印出目前記錄到的LastExitDate/LastExitTime，用來觀察
// 是否在出場後的下一次運算就被重置回0
if Position = 0 then begin
    Print("【空手中-讀取現值】Time:", NumToStr(Time,0),
          " LastExitDate:", NumToStr(LastExitDate,0),
          " LastExitTime:", NumToStr(LastExitTime,0));
end;

// 簡單的多單進出場邏輯，僅用於觸發部位轉換，非重現重點
if Position = 0 then begin
    if Close > MA5 and Close > MA20 then begin
        SetPosition(1, MARKET, Label:="測試進場");
    end;
end
else if Position > 0 then begin
    if Close < MA5 then begin
        SetPosition(0, MARKET, Label:="測試出場");
        LastExitDate = Date;
        LastExitTime = Time;
        Print("【出場當下確認】Time:", NumToStr(Time,0),
```

```
49                " LastExitDate:", NumToStr(LastExitDate,0),
50                " LastExitTime:", NumToStr(LastExitTime,0));
51      end;
52  end;
53
```