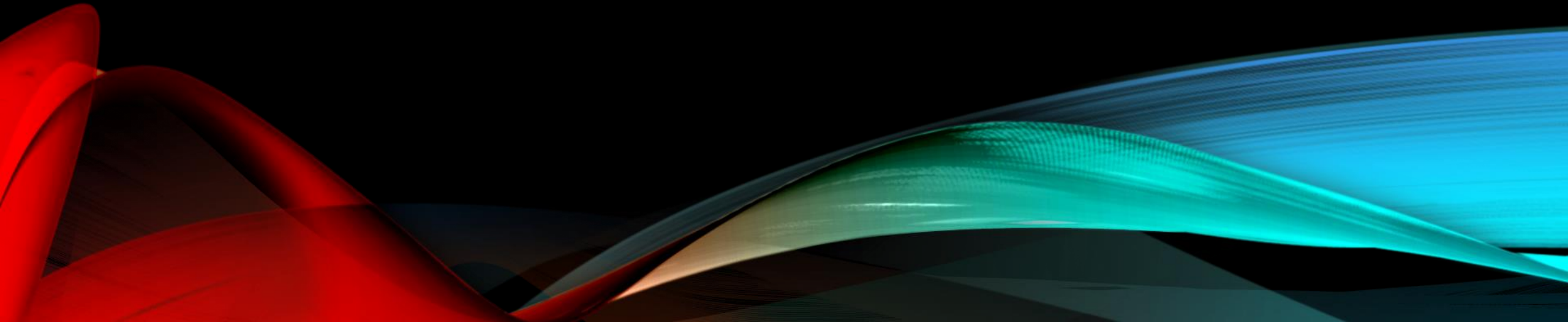


EXCEL VBA 進階班

Lecture 6

SECTION 9. USER-DEFINED TYPE

TimeSeries (3)



USER-DEFINED TYPE

- 上次利用自我迴歸模型建立了一個投資策略，但發現效果不如預期。
- 現在回過來我們使用簡單的交易策略－動量交易策略（ **Momentum Strategy** ）。
- 由於使用上很簡單我們甚至可以以投資組合方式進行。

USER-DEFINED TYPE

- Jegadeesh 與 Titman (1993) 在對資產股票組合的中期收益進行研究時發現，以三個月到一年為間隔所構造的股票組合的中期收益呈現出延續性，即中期價格具有向某一方向連續變動的動量效應，而非均數回歸之特性。
- 因此，以中期來說：
 - 表現良好的股票會繼續上漲
 - 表現不好的股票會繼續下跌

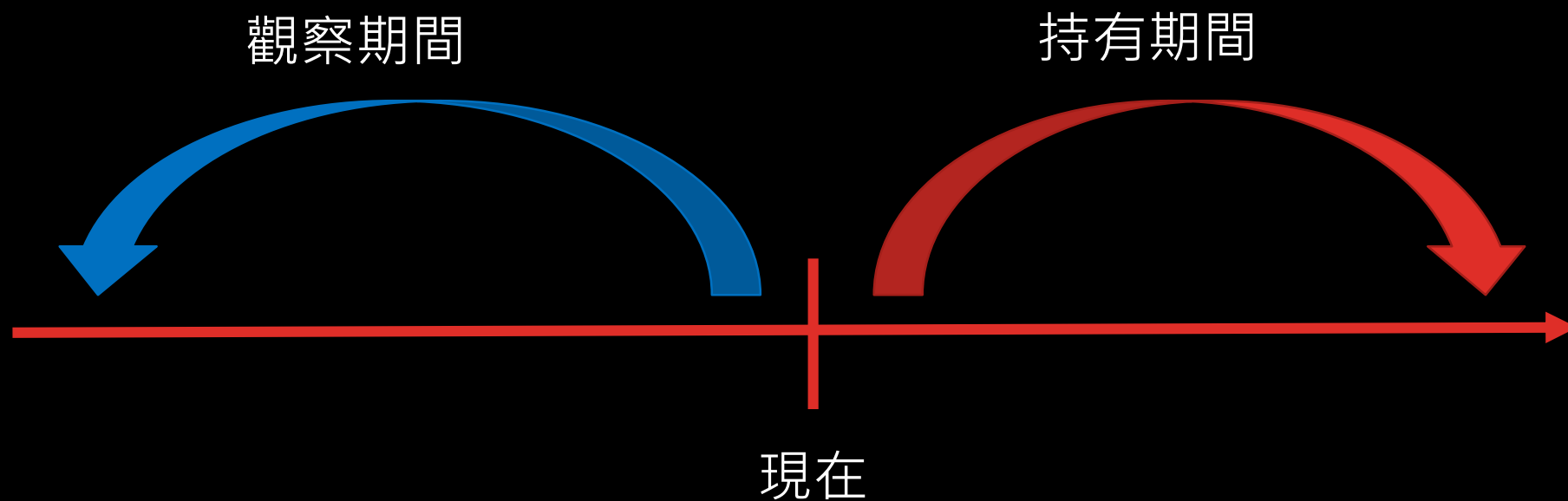
USER-DEFINED TYPE

- 由此我們可以利用該特性建立一個投資策略：
 - 步驟 1. 首先定義好：
 - 觀察個股票市場，假設有 N 個不同的股票。
 - 決定好一段過去觀察期間。
 - 決定好一段持有期間。

USER-DEFINED TYPE

- 步驟 2. 間，假設我們人在 t 時間點，利用過去觀察期的歷史資料與選定的方法來衡量各股票表現之績效。
- 步驟 3. 將表現最佳的幾個股票定為贏家股票，而表現最差的幾個股票定為輸家股票，接著：
 - 3-1. 做多贏家股票，到持有期結束平倉。
 - 3-2. 放空輸家股票，到持有期結束平倉。

USER-DEFINED TYPE



USER-DEFINED TYPE

- 而要如何決定衡量股票績效的方法？
- 方法 1. 第一個簡單的方法是以目前價格除上過去 52 週最高價的距離：

$$\text{股票表現} = \frac{\text{目前價格}}{\text{過去 52 週最高價}}$$

- 方法 2. 另一個簡單方法的是計算過去一段時間的平均超額報酬率：
 - 超額報酬率之計算為：超額報酬率 = 股票報酬率 - 市場報酬率
 - 假設我們的持有期為三個月，我們可以計算各個股票過去三個月超額報酬率平均並排序，將最高者作為贏家股票，最差者作為輸家股票，並進行交易策略。

USER-DEFINED TYPE

- 而如何決定輸家與贏家組合數量呢？
- 根據 George 與 Hwang (2004) 是以所有觀察股票的表現前 10% 作為贏家股票，最後 10% 作為輸家股票。

USER-DEFINED TYPE

- 可以先用範例資料與 Excel 試驗結果。
- 以美國股票為例，將 S&P 500 報酬率當作市場報酬率進行測試。
- 以 2014 年資料尋找贏家與輸家股票，2015 年進行交易。
- 如果要連續多年要如何做？

補充：TIME SERIES MOMENTUM

- 前述方法有一個共通缺點，所有贏家與輸家股票給予相同的買賣單位數。
- 可以將其改進為：
 - 越穩定獲利或損失的做多或放空越多單位數。
 - 反之波動越大越不穩定的，要較保守，因此買賣較少的單位數。

補充：TIME SERIES MOMENTUM

- 對於該部份我們可以參考另外一個衡量績效的方法，Moskowitz、Ooi 與 Pedersen (2012) 之 Time Series Momentum。
- 該方法是以一時間序列模型來衡量股票績效，模型如下：

$$\frac{r_t}{\sigma_{t-1}} = \alpha + \beta_h \frac{r_{t-h}}{\sigma_{t-h-1}} + \varepsilon_t$$

補充：TIME SERIES MOMENTUM

- 每一期的時間是一個月，個參數代表意義如下：
- r_t ：第 t 期之報酬率。
- σ_t ：到第 t 期為止之股票年化波動度。

補充：TIME SERIES MOMENTUM

- 年化波動度的計算方式如下：

$$\sigma_t^2 = 261 \sum_{i=0}^{\infty} (1 - \delta) \delta^i (r_{t-1-i} - \bar{r}_t)$$

- 261 代表的是一年的工作天，將日波動度呈上一年的工作天變為年化波動度。

補充：TIME SERIES MOMENTUM

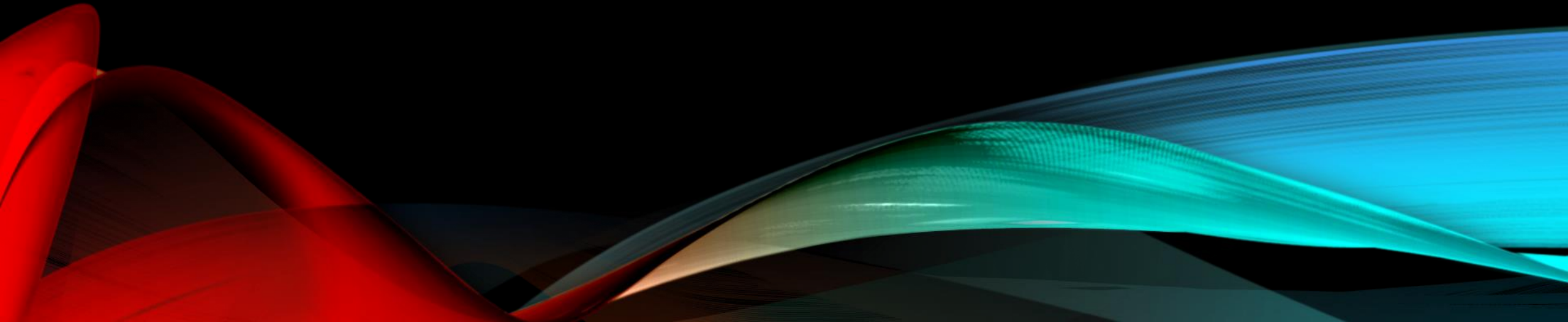
- 注意：
 - 此處的每一期為一日，並非前面的一年。
 - 加總到無限大代表的意義為使用所有過去的歷史資料，但為了避免久遠歷史影響過大，使用權種 δ ，決定方法如下頁所示。
 - \bar{r}_t 為所有歷史資料的加權平均，權重計算方法與波動度相同。
- 而權重為給定一個 δ 使得：

$$\sum_{i=0}^{\infty} (1 - \delta) \delta^i i = \frac{\delta}{1 - \delta} = 60$$

補充：TIME SERIES MOMENTUM

- 投資的單位數可以依照波動度進行調整，為 $\frac{1}{\sigma_{t-1}}$ 。
- 由於波動度介於 0 ~ 1 之間，因此：
 - 波動度越大的股票，單位數會越少。
 - 波動度越小的股票，單位數會越大。

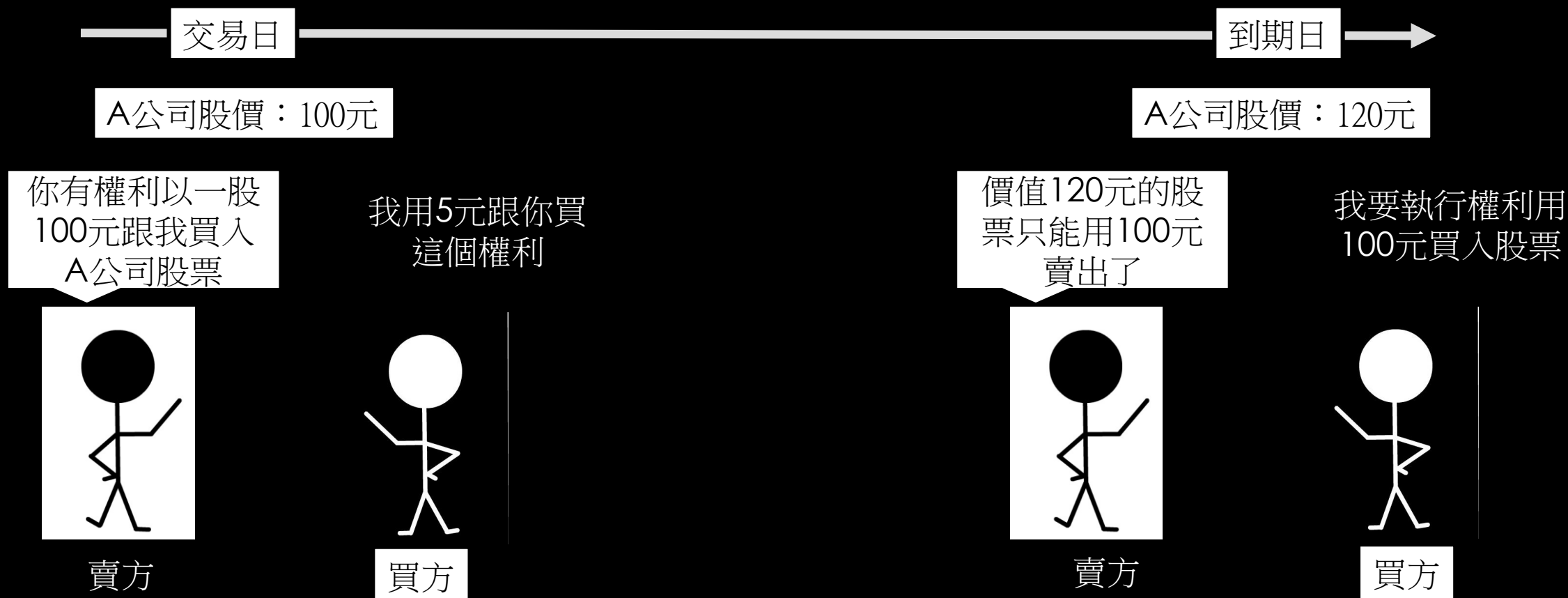
SECTION 10. 物件導向簡介



範例：歐式選擇權

- 我們利用衍生性金融商品到期損益計算作為範例，初步介紹何謂物件導向。
- 首先來回憶歐式買權賣權損益計算：

物件導向簡介



物件導向簡介

- 買權是買方以約定價格向賣方買入該資產的權利。
- 買方用100元買入價值120元的股票，賺了20元。
- 但如到期日股票價格為90元時，買方會執行此權利嗎？

物件導向簡介

- 首先，我們可以先想像一個歐式買權擁有以下的資訊：

- 履約價：已約定價格買（或賣）某一商品價格。
- 到期日：該選擇權決定是否履約的日期。

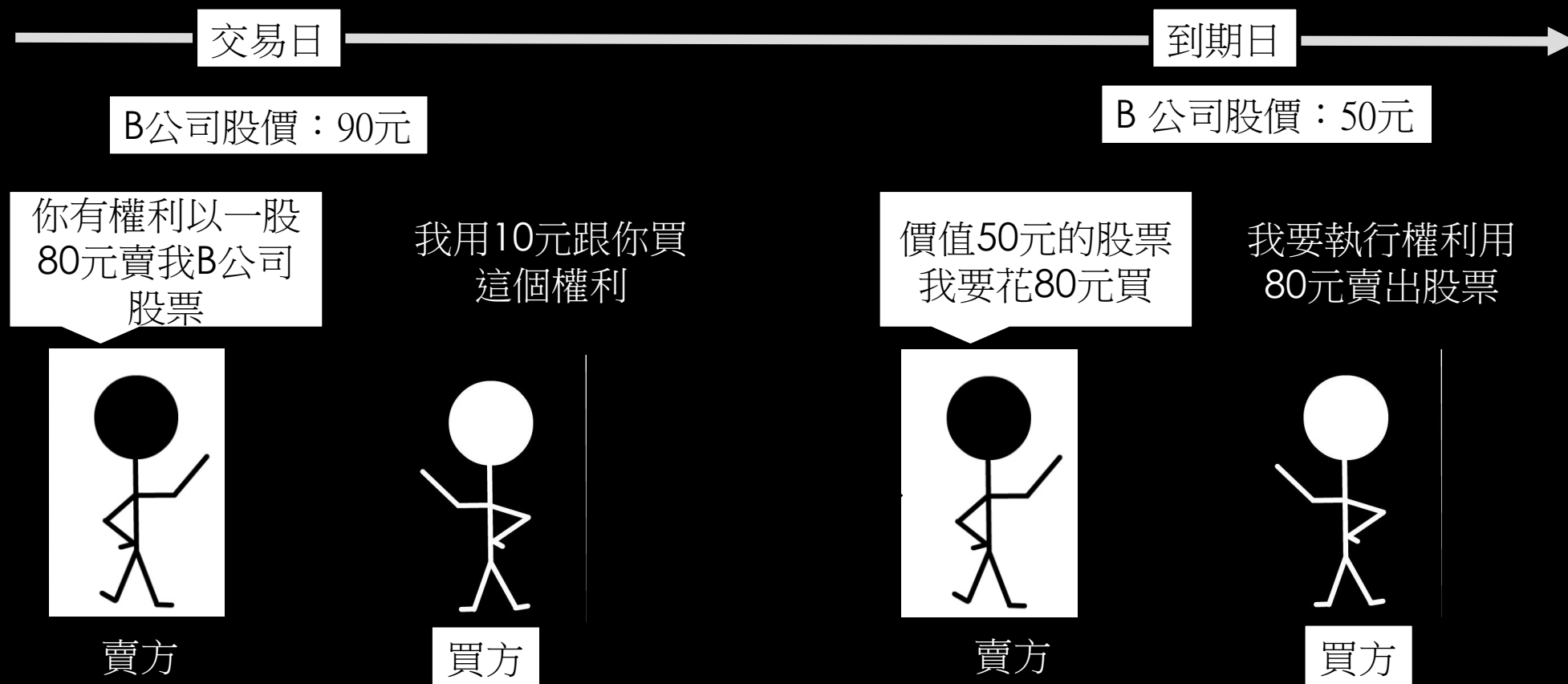
屬性

- 而擁有一個歐式買權可以做以下動作：

- 在到期日執行選擇權獲得報酬。

方法

物件導向簡介



物件導向簡介

- 賣權是買方以約定價格向賣方賣出該資產的權利。
- 買方用80元賣出價值50元的股票，賺了30元。
- 但如到期日前股票價格都高於100元，買方會執行此權利嗎？

物件導向簡介

- 由此可知，歐式買權到期損益為：

$$payoff_{call} = \max(spot - strike)$$

- 歐式賣權到期損益為：

$$payoff_{put} = \max(strike - spot)$$

物件導向簡介

- 首先一個投資人手上持有一個歐式買權的投資組合 (假設所有選擇權標的物皆為同一個資產)，每個選擇權紀錄了以下資料：

| | A | B | C |
|---|------|------------|-----|
| 1 | 買權編號 | 到期日 | 履約價 |
| 2 | 1 | 2016/11/9 | 100 |
| 3 | 2 | 2016/11/9 | 102 |
| 4 | 3 | 2016/12/20 | 101 |
| 5 | 4 | 2017/1/4 | 103 |
| 6 | 5 | 2017/2/2 | 102 |

物件導向簡介

- 一開始，我們希望有一個程式能計算各個買權在某一天的損益，我們可以新增以下程式：

```
Sub calculateTodayNetting()
```

```
' 記錄當天資訊
```

```
Dim spot As Double: spot = Range("spot").Value
```

```
Dim valueDate As Date: valueDate = Range("valueDate").Value
```

```
'紀錄投資組合資訊
```

```
Dim nCall As Integer: nCall = Cells(1, 1).End(xlDown).Row - 1
```

```
Dim strike As Variant: strike = Range("C2", Cells(nCall + 1, "C")).Value
```

```
Dim expiryDate As Variant: expiryDate = Range("B2", Cells(nCall + 1, "B")).Value
```

```
Dim i As Integer
```

```
Dim output As String
```

```
For i = 1 To nCall
```

```
    output = output & "買權編號 " & i & " 之到期損益為: "
```

```
    If expiryDate(i, 1) = valueDate Then
```

```
        output = output & Application.Max(spot - strike(i, 1))
```

```
    Else
```

```
        output = output & 0
```

```
    End If
```

```
    output = output & Chr(10)
```

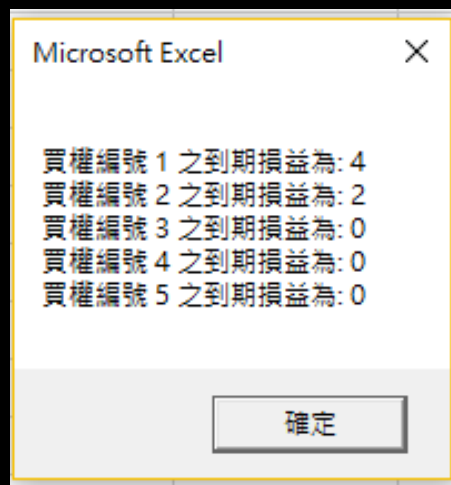
```
Next i
```

```
MsgBox output
```

```
End Sub
```

物件導向簡介

- 執行後可以得到以下結果：



物件導向簡介

- 但這之中其實存在一些問題，第一，我們沒有辦法檢查輸入的資料是不是有誤：
 - strike 是否為 double？
 - expiryDate 是否為日期？
- 假如發生以下狀況，程式執行就會遭遇錯誤：

| | A | B | C |
|---|------|------------|-----|
| 1 | 買權編號 | 到期日 | 履約價 |
| 2 | 1 | 2016/11/9 | 100 |
| 3 | 2 | 2016/11/9 | ABC |
| 4 | 3 | 2016/12/20 | 101 |
| 5 | 4 | 2017/1/4 | 103 |
| 6 | 5 | 2017/2/2 | 102 |

物件導向簡介

- 這類錯誤還是小問題，因為可以發現錯誤地方並修正，如果是出現不合理的數字呢？
- 例如選擇權並不會出現負的履約價，但假如某個選擇權不小心因輸入錯誤造成履約價為 0？

物件導向簡介

- 第二，現在我們要在新增一個子程序來計算當天總收益：

```
Sub calculateTotalGainLoss()  
' 記錄當天資訊  
Dim spot As Double: spot = Range("spot").Value  
Dim valueDate As Date: valueDate = Range("valueDate").Value  
  
'紀錄投資組合資訊  
Dim nCall As Integer: nCall = Cells(1, 1).End(xlDown).Row - 1  
Dim strike As Variant: strike = Range("C2", Cells(nCall + 1, "C")).Value  
Dim expiryDate As Variant: expiryDate = Range("B2", Cells(nCall + 1, "B")).Value  
Dim i As Integer  
Dim totalNetting As Double  
  
For i = 1 To nCall  
  
    If expiryDate(i, 1) = valueDate Then  
        totalNetting = totalNetting + Application.Max(spot - strike(i, 1))  
    End If  
Next i  
  
MsgBox DateValue(valueDate) & " 總損益為: " & totalNetting  
  
End Sub
```


物件導向簡介

- 但之後假如要加入：
 - 可以選擇買權賣權
 - 可以選擇買賣選擇權
 - 加入 EKI 與 EKO
- 那要一次改兩個地方，還要擔心有沒有漏改，怎麼處理這些問題？

物件導向簡介

- 「資料」和「程式碼」是程式的兩大核心
 - 在本章之前的程式寫作方式屬「程序導向(Process- Oriented)」,類似 C 語言。
 - 「物件導向(Object-Oriented)」是以資料(或稱物件) 為程式的核心,主要差異在於資料控制程式碼的運作。

物件導向簡介

- 「物件」統稱系統中的資料,而「物件」是來自於「類別(Class)」
- 兩者之間的關係如同「依照設計藍圖(Class) 建構房屋(Object)」類別中會定義物件的「成員(member)」
 - 成員包含了物件的屬性(物件資料, 名詞)和方法(物件行為, 動詞)
 - 利用「類別」定義的規範產生「物件」的動作稱為「實體化(instantiate)」

物件導向簡介

- 物件導向的三大特性：
 - 封裝 (encapsulation) ← VBA 支援
 - 繼承 (inheritance) ← VBA 不支援
 - 多型 (polymorphism) ← VBA 支援
- 所有的物件導向程式都需要滿足這三個 特性,才稱的上是物件導向程式 。
- 因此 **VBA 並不是一個完整的物件導向程式語言**，但仍能利用物件導向部分優點讓程式碼更容易維護。

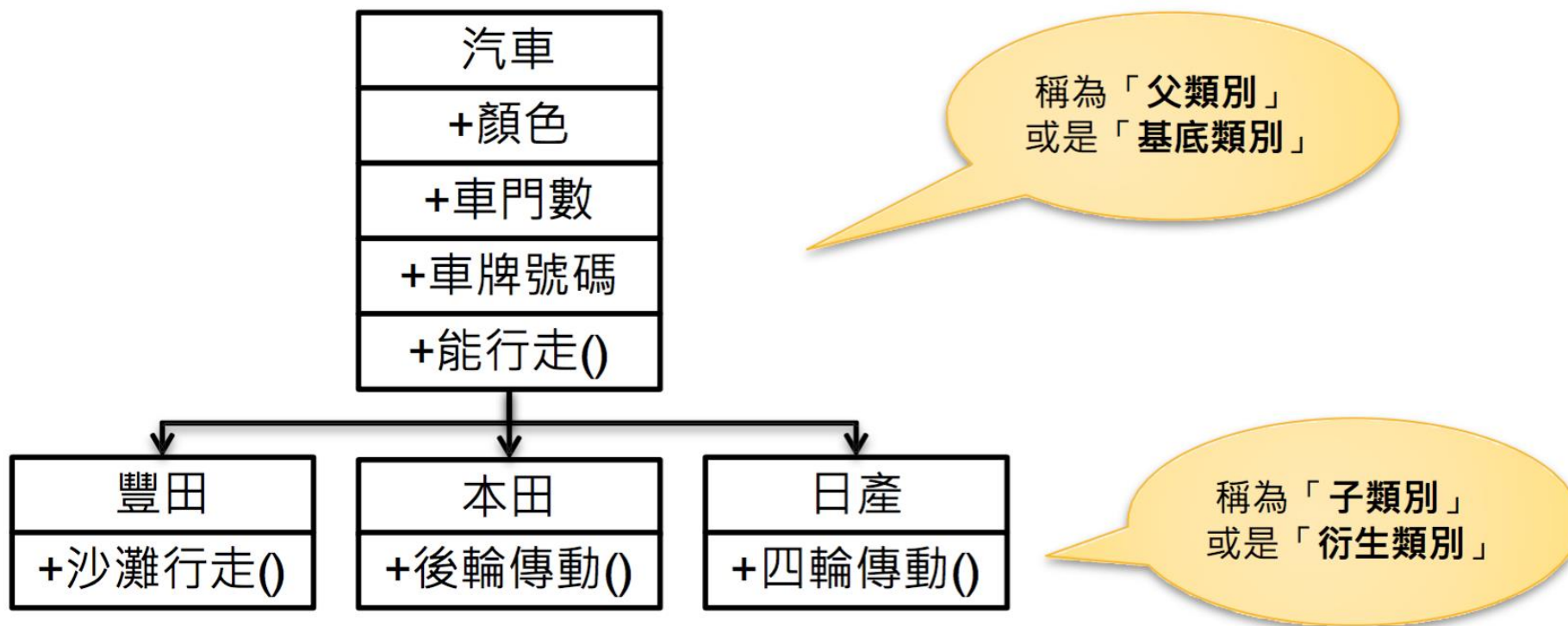
物件導向簡介：封裝

- 將類別中的程式碼或是資料保護起來，避免 受到外界不當的干擾或是使用需要存取類別中的資料時，必需透過定義明 確的「介面(**interface**)」才可以存取資料。
- 像使用電視時，我們只需要按下按鈕(介面的使用) 就可以觀賞節目，而不需要看到電晶體的運作。

物件導向簡介：繼承

- 目的在定義新類別時,不需要重複的定義相同的成員變數或是成員方法。
- VBA 並無繼承的語法，但是可以透過某些方法達到類似繼承的效果。

物件導向簡介：繼承



物件導向簡介：多型

- 多型的精神：一個介面，多種使用方法。
- 就定義面而言多型大多定義如下：將相同的訊息傳遞給不同的物件，進而引發出不同的行為反應。