

10-4 Class 練習題

練習一：電影票券 (Ticket)

情境說明

你正在設計一套電影院售票系統。每張票券記錄了電影名稱、座位號碼、票價，以及是否已被使用。票券一旦使用就不能再次入場；票價不能設為負數。

規格列表

成員變數 (皆為 `private`)

變數名稱	型態	說明
<code>movieName</code>	<code>string</code>	電影名稱
<code>seatNumber</code>	<code>int</code>	座位號碼
<code>price</code>	<code>int</code>	票價 (元)
<code>used</code>	<code>bool</code>	是否已使用

建構子

建構子	初始值
<code>Ticket()</code>	<code>movieName="未命名"</code> 、 <code>seatNumber=0</code> 、 <code>price=0</code> 、 <code>used=false</code>
<code>Ticket(string movieName, int seatNumber, int price)</code>	依參數設定， <code>used=false</code>

兩個建構子的參數名稱與成員變數相同，需使用 `this->` 或初始化列表。

Getter (皆加 `const`)

函式	回傳型態	說明
<code>getMovieName()</code>	<code>string</code>	回傳電影名稱
<code>getSeatNumber()</code>	<code>int</code>	回傳座位號碼
<code>getPrice()</code>	<code>int</code>	回傳票價
<code>isUsed()</code>	<code>bool</code>	回傳是否已使用

Setter (含驗證)

函式	驗證規則
<code>setPrice(int price)</code>	<code>price < 0</code> 時印出 "票價不能為負數!" 並放棄修改

其他成員函式

函式	回傳	說明
<code>use()</code>	<code>bool</code>	若已使用，印出 "此票券已使用過!" 並回傳 <code>false</code> ；否則將 <code>used</code> 設為 <code>true</code> ，印出入場訊息並回傳 <code>true</code>
<code>print() const</code>	<code>void</code>	印出票券所有資訊 (格式見預期輸出)

main()

```
int main() {
    Ticket t1("星際大戰", 12, 280);
    Ticket t2;
```

```

t1.print();
t2.print();

t1.use();
t1.use();          // 重複使用

t1.setPrice(-100); // 非法
t1.setPrice(250); // 合法

cout << t1.getMovieName() << " 目前票價：" << t1.getPrice() << " 元" << endl;

t1.print();
}

```

預期輸出

```

[票券] 星際大戰 | 座位 12 | 票價 280 元 | 狀態：未使用
[票券] 未命名 | 座位 0 | 票價 0 元 | 狀態：未使用
入場成功！電影：星際大戰，座位：12
此票券已使用過！
票價不能為負數！
星際大戰 目前票價：250 元
[票券] 星際大戰 | 座位 12 | 票價 250 元 | 狀態：已使用

```

練習二：水壺 (`Pitcher`)

情境說明

你設計一個登山水壺管理程式。每個水壺有名稱、最大容量和目前水量。注水時若超過容量會溢出；倒水時若水量不足則失敗；容量不能設得比目前水量還小。

規格列表

成員變數 (皆為 `private`)

變數名稱	型態	說明
<code>name</code>	<code>string</code>	水壺名稱
<code>capacity</code>	<code>int</code>	最大容量 (ml)
<code>current</code>	<code>int</code>	目前水量 (ml)

建構子

建構子	初始值
<code>Pitcher()</code>	<code>name="水壺"</code> 、 <code>capacity=1000</code> 、 <code>current=0</code>
<code>Pitcher(string name, int capacity)</code>	依參數設定， <code>current=0</code>

Getter (皆加 `const`)

函式	回傳型態	說明
<code>getName()</code>	<code>string</code>	回傳水壺名稱
<code>getCapacity()</code>	<code>int</code>	回傳最大容量
<code>getCurrent()</code>	<code>int</code>	回傳目前水量

Setter (含驗證)

函式	驗證規則
<code>setCapacity(int capacity)</code>	<code>≤ 0</code> 印出 "容量必須大於 0!" ; 新容量 <code>< current</code> 印出 "新容量小於目前水量，無法設定!"

其他成員函式

函式	回傳	說明
<code>fill(int amount)</code>	<code>void</code>	注水；若超出容量則補滿並印出溢出量，否則正常注水
<code>pour(int amount)</code>	<code>bool</code>	倒水；水量不足回傳 <code>false</code> 並印出提示，否則正常倒出回傳 <code>true</code>
<code>status() const</code>	<code>void</code>	印出水壺狀態，含百分比（格式見預期輸出）

main()

```
int main() {
    Pitcher p1("運動水壺", 750);
    Pitcher p2;

    p1.status();
    p2.status();

    p1.fill(500);
    p1.fill(400);        // 會溢出

    p1.pour(200);
    p1.pour(600);        // 水量不足

    p1.setCapacity(-1); // 非法
    p1.setCapacity(100); // 非法 (小於目前水量 550)

    cout << p1.getName() << " 容量：" << p1.getCapacity() << " ml" << endl;

    p1.status();
}
```

預期輸出

```
[運動水壺] 0 / 750 ml (0%)
[水壺] 0 / 1000 ml (0%)
運動水壺 注入 500 ml。
運動水壺 已裝滿，多餘 150 ml 溢出。
運動水壺 倒出 200 ml。
運動水壺 水量不足，無法倒出 600 ml！
容量必須大於 0！
新容量小於目前水量，無法設定！
運動水壺 容量：750 ml
[運動水壺] 550 / 750 ml (73%)
```

練習三：遊戲計分板 (ScoreBoard)

情境說明

你正在設計一款小遊戲的計分系統。計分板記錄玩家名稱、歷史最高分、當前分數和剩餘生命數。每次得分若超過歷史最高分則更新紀錄；失去所有生命時宣告遊戲結束；重新開始時分數歸零但最高分保留。

規格列表

成員變數 (皆為 `private`)

變數名稱	型態	說明
<code>playerName</code>	<code>string</code>	玩家名稱
<code>highScore</code>	<code>int</code>	歷史最高分
<code>currentScore</code>	<code>int</code>	當前分數
<code>lives</code>	<code>int</code>	剩餘生命數

建構子

建構子	初始值
ScoreBoard()	playerName="玩家"、highScore=0、currentScore=0、lives=3
ScoreBoard(string playerName, int lives)	依參數設定, highScore=0、currentScore=0

Getter (皆加 const)

函式	回傳型態	說明
getPlayerName()	string	回傳玩家名稱
getHighScore()	int	回傳歷史最高分
getCurrentScore()	int	回傳當前分數
getLives()	int	回傳剩餘生命數

Setter (含驗證)

函式	驗證規則
setPlayerName(string playerName)	名稱為空字串時印出 "玩家名稱不能為空!" 並放棄修改

其他成員函式

函式	回傳	說明
addScore(int points)	void	加分 (≤0 印出提示拒絕); 若 currentScore > highScore 則更新並印出新紀錄訊息
loseLife()	bool	扣一條命; lives=0 時印出遊戲結束訊息並回傳 false, 否則回傳 true
reset()	void	currentScore 歸零、lives 回復 3, highScore 保留, 印出提示
print() const	void	印出所有資訊 (格式見預期輸出)

main()

```
int main() {
    ScoreBoard sb("小明", 3);
    ScoreBoard sb2;

    sb.print();
    sb2.print();

    sb.addScore(100);
    sb.addScore(250);
    sb.addScore(50);

    sb.loseLife();
    sb.loseLife();
    sb.loseLife();    // 第三條命，遊戲結束

    sb.setPlayerName(""); // 非法
    sb.setPlayerName("小明Pro");

    sb.reset();
    sb.print();

    sb.addScore(500); // 超越舊最高分
    sb.print();
}
```

預期輸出

```
== 小明 == 當前: 0 | 最高: 0 | 命: 3
== 玩家 == 當前: 0 | 最高: 0 | 命: 3
新紀錄! 最高分更新為 100
小明得 100分, 目前: 100
新紀錄! 最高分更新為 350
小明得 250分, 目前: 350
新紀錄! 最高分更新為 400
```

```
小明 得 50 分，目前：400
小明 失去一條命，剩餘：2
小明 失去一條命，剩餘：1
小明 失去一條命，剩餘：0
遊戲結束！小明 最終得分：400
玩家名稱不能為空！
小明Pro 重新開始，最高分保留：400
== 小明Pro == 當前：0 | 最高：400 | 命：3
新紀錄！最高分更新為 500
小明Pro 得 500 分，目前：500
== 小明Pro == 當前：500 | 最高：500 | 命：3
```

🕒Revision #1

★Created 31 May 2026 00:59:16 by huihui

✎Updated 31 May 2026 01:06:27 by huihui