

9-2 vector 的基礎操作

vector 基礎操作

1. 引入標頭檔與宣告

使用 `vector` 前，需要引入標頭檔：

```
#include <iostream>
#include <vector>

using namespace std;
```

2. 宣告與初始化

```
// 方式一：宣告空的 vector
// capacity:0, size: 0
vector<int> v1;

// 方式二：指定初始大小（元素值為 0）
// capacity:5, size: 5
vector<int> v2(5);          // [0, 0, 0, 0, 0]

// 方式三：指定大小與預設值
// capacity:5, size: 5
vector<int> v3(5, 100);    // [100, 100, 100, 100, 100]

// 方式四：用初始化列表
// capacity:5, size: 5
vector<int> v4 = {3, 1, 4, 1, 5}; // [3, 1, 4, 1, 5]

// 方式五：複製另一個 vector
// capacity:5, size: 5
vector<int> v5(v4);        // [3, 1, 4, 1, 5]

// 也可以存其他型別
vector<double> scores;
vector<string> names;
vector<bool> flags;
```

3. 查詢大小與清空

```
vector<int> v = {1, 2, 3, 4, 5};

cout << v.size();    // 5：目前元素個數
cout << v.empty();   // 0 (false)：是否為空

v.clear();           // 清空所有元素
cout << v.size();    // 0
cout << v.empty();   // 1 (true)
```

4. 新增與移除元素

在末尾新增元素

`vector` 的成員函數 `push_back(val)`，可以將一個值新增到目前 `vector` 的尾端。

```
vector<int> v;
v.push_back(10); // [10]
v.push_back(20); // [10, 20]
v.push_back(30); // [10, 20, 30]
```

移除末尾元素

vector 的成員函數 `pop_back()`，可以將 vector 尾端的那個元素移除。

```
v.pop_back(); // [10, 20]
```

⚠️ **注意**：對空的 vector 呼叫 `pop_back()` 是未定義行為 (UB)，請先確認 `!v.empty()`。

5. 存取元素

下標運算子 `[]`

```
vector<int> v = {10, 20, 30};  
cout << v[0]; // 輸出 10  
v[1] = 99; // 修改第二個元素
```

帶邊界檢查的存取 `at(i)`

```
cout << v.at(0); // 輸出 10  
cout << v.at(5); // 超出範圍 → 丟出 std::out_of_range 例外
```

方式	速度	邊界檢查	建議使用時機
<code>v[i]</code>	較快	<input type="checkbox"/> 沒有	確定索引合法時
<code>v.at(i)</code>	稍慢	<input type="checkbox"/> 有	需要安全保障時

6. 範例：動態收集成績

```
#include <iostream>  
#include <vector>  
  
using namespace std;  
  
int main() {  
    vector<int> scores;  
    int n, score;  
  
    cout << "請輸入學生人數:";  
    cin >> n;  
  
    for (int i = 0; i < n; i++)  
    {  
        cout << "輸入第 " << i + 1 << " 位學生成績:";  
        cin >> score;  
        scores.push_back(score); // 動態新增  
    }  
  
    // 計算總分  
    int total = 0;  
    for (int i = 0; i < scores.size(); i++) {  
        total += scores[i];  
    }  
  
    cout << "平均分數:" << (double)total / scores.size() << endl;  
    return 0;  
}
```

🔄Revision #1

★Created 26 April 2026 11:57:49 by huihui

🔧Updated 26 April 2026 12:22:03 by huihui