cout格式控制

在C++中，`cout` 通过流操作符和格式控制函数支持多种输出格式。以下是常见的格式控制方法及示例：

1. 整数进制控制

- **十六进制**：`hex`
- **八进制**：`oct`
- **十进制**：`dec`
- **显示进制前缀**：`showbase`
- **字母大写**：`uppercase`

```cpp
#include <iostream>
using namespace std;

int main()
{
    int num = 255;
    cout << showbase << hex << num << endl;       // 输出 0xff
    cout << uppercase << hex << num << endl;      // 输出 0XFF
    cout << oct << num << endl;                   // 输出 0377
    cout << dec << num << endl;                   // 输出 255
    return 0;
}
```

2. 浮点数格式

- **固定小数位数**：`fixed`
- **科学计数法**：`scientific`
- **设置精度**：`setprecision(n)`（需包含 `<iomanip>`）

```cpp
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    double pi = 3.1415926535;
    cout << fixed << setprecision(2) << pi << endl;     // 输出 3.14
    cout << scientific << setprecision(3) << pi << endl; // 输出 3.142e+00
    return 0;
}
```

3. 字段宽度与填充

- **设置宽度**：`setw(n)`
- **左对齐/右对齐**：`left` / `right`

- 填充字符：`setfill(c)`

```cpp
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    cout << left << setw(10) << setfill('*') << 42 << endl;  // 输出 42********
    cout << right << setw(5) << setfill('0') << 7 << endl;   // 输出 00007
    return 0;
}
```

4. 布尔值格式

- 文本形式：`boolalpha`
- 数字形式：`noboolalpha`

```cpp
#include <iostream>
using namespace std;

int main()
{
    cout << boolalpha << true << endl;   // 输出 true
    cout << noboolalpha << true << endl; // 输出 1
    return 0;
}
```

5. 其他格式控制

- 显示正号：`showpos`
- 恢复默认浮点格式：`defaultfloat`

```cpp
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    cout << showpos << 10 << endl;        // 输出 +10
    cout << defaultfloat << 123.456 << endl; // 输出根据值自动选择格式
    return 0;
}
```

6. 综合示例

```cpp
#include <iostream>
#include <iomanip>
```

```cpp
using namespace std;

int main()
{
    int num = 30;
    double value = 1234.5678;

    // 十六进制、宽度8、填充0、大写
    cout << uppercase << hex << showbase << setw(8) << setfill('0') << num << endl;
// 输出 0X00001E

    // 科学计数法、精度3位、显示符号
    cout << scientific << showpos << setprecision(3) << value << endl; // 输出
+1.235e+03

    // 表格格式：左对齐，宽度10，填充空格
    cout << left << setfill(' ')
         << setw(10) << "Name" << setw(5) << "Age" << endl
         << setw(10) << "Alice" << setw(5) << 25 << endl;
    // 输出：
    // Name      Age
    // Alice     25
    return 0;
}
```

注意事项

- **持久性设置**：`hex`、`fixed` 等设置会持续生效，直到被修改。
- **瞬时设置**：`setw` 仅影响下一个输出项。
- **头文件依赖**：`setw`、`setprecision` 等需包含 `<iomanip>`。

通过组合这些操作符，可以灵活控制输出格式以满足不同需求。