C++ 笔记

shiyu-hong

2025-04-05

目录

欢迎

欢迎阅读这份 C++ 学习笔记!

第一章　未定义行为

在 C++ 中，**未定义行为（Undefined Behavior，UB）**特指违反语言规范的代码操作，其具体表现未在 C++ 标准中明确定义。此类行为虽然能够通过编译，但可能引发程序崩溃、产生错误输出，甚至因编译器实现差异或硬件特性导致完全不可预知的运行结果。

1.1　变量未定义行为

在 C++ 中，**未初始化变量**指声明变量后未显示赋初值，即直接访问其内存残留数据的操作。根据 C++ 标准，由于该操作的语义未被明确定义，故被归类为**未定义行为（Undefined Behavior，UB）**。

1.1.1　局部变量未初始化（最常见风险之一）

```
#include <iostream>

int main(int argc, char **argv) {
  int x;        // 未初始化局部变量
  int y{10};    //
  int z{x + y}; // 未定义行为（变量 x 未初始化，可能包含随机垃圾值）

  std::cout << z << std::endl; // 输出随机垃圾值

  return 0;
}
```

- **输出可能：**
  - -1847796918 (MSVC Debug)
  - 10 (MSVC Release)

1.1.2　条件分支变量未初始化

```
#include <iostream>

void check_condition() {
```

```
  bool flag;  // 未初始化条件变量
  if (flag) { // 未定义行为（条件判断可能随机成立）
    std::cout << "Flag is true!" << std::endl;
  } else {
    std::cout << "Flag is false!" << std::endl;
  }
}

int main(int argc, char **argv) {
  check_condition();

  return 0;
}
```

- **输出可能：**
  - Flag is true! (MSVC Debug)
  - Flag is false! (MSVC Release)

1.1.3 数组未初始化

```
#include <iostream>

void process_array() {
  int buffers[3];    // 未初始化数组

  for (auto i = 0; i < 3; ++i) {
    buffers[i] += 1; // 未定义行为（操作未初始化的值）
  }

  for (auto i = 0; i < 3; ++i) {
    std::cout << buffers[i] << " "; // 输出随机垃圾值
  }
}

int main(int argc, char **argv) {
  process_array();

  return 0;
}
```

- **输出可能：**
  - -1166904743 32760 -1166904743 (MSVC Debug)
  - 8 1 1 (MSVC Release)

1.1.4 类成员变量未初始化

```cpp
#include <iostream>

class Point {
public:
  void print() { std::cout << "(" << x_ << ", " << y_ << ")" << std::endl; }

private:
  int x_; // 未在构造函数中初始化
  int y_; // 未在构造函数中初始化
};

void log_point() {
  Point point;   // 未显示初始化成员
  point.print(); // 输出随机垃圾值
}

int main(int argc, char **argv) {
  log_point();

  return 0;
}
```

- **输出可能：**
  - (2130211416, 32758) (MSVC Debug)
  - (0, 0) (MSVC Release)

1.1.5 指针未初始化（最常见风险之一）

```cpp
#include <iostream>

int main(int argc, char **argv) {
  int *ptr; // 未初始化指针
  *ptr = 3; // 未定义行为（可能覆盖随机内存，触发段错误）

  return 0;
}
```

- **输出可能：**
  - 段错误 (MSVC Debug)
  - 段错误 (MSVC Release)

1.2 内存未定义行为

在 C++ 中，**内存未定义行为**指程序通过非法方式操作内存资源，导致 C++ 标准无法为其执行结果提供任何保证的行为。此类行为直接违反内存安全规则，可能引发程序崩溃、数据损坏或安全漏洞，且其具体表现高度依赖编译器实现、操作系统及硬件环境。

1.2.1 空指针解引用

```cpp
#include <iostream>

int main(int argc, char **argv) {
  int *ptr = nullptr; // 显示初始化为空指针

  // 未定义行为：解引用空指针
  *ptr = 3;                       // 危险：将数据写入空指针地址
  std::cout << *ptr << std::endl; // 危险：读取空指针内容

  return 0;
}
```

- **输出可能：**
  - 段错误 (MSVC Debug)
  - 段错误 (MSVC Release)