第9章 列表初始化

《现代C++语言核心特性解析》 谢丙堃

直接初始化和拷贝初始化

- 使用括号初始化的方式叫做直接初始化
- 使用等号初始化的方式叫做拷贝初始化

```
struct C {
    C(int var) : member(var){}
    int member;
};

int main()
{
    int x = 5;
    int x1(8);
    C x2 = 4;
    C x3(4);
}
```

列表初始化语法

- 使用大括号{}对变量进行初始化

```
int x = {5};            // 拷贝初始化
int x1{8};              // 直接初始化
```

- 列表初始化标准容器

```
std::vector<int> x2{ 1,2,3,4,5 };
std::vector<int> x3 = { 1,2,3,4,5 };
std::list<int> x4{ 1,2,3,4,5 };
std::list<int> x5 = { 1,2,3,4,5 };
std::set<int> x6{ 1,2,3,4,5 };
std::set<int> x7 = { 1,2,3,4,5 };
std::map<std::string, int> x8{ {"bear",4}, {"cassowary",2}, {"tiger",7} };
std::map<std::string, int> x9 = { {"bear",4}, {"cassowary",2}, {"tiger",7} };
```

std::initializer_list

- 支持std::initializer_list为形参的构造函数

```
#include <string>
struct C {
    C(std::initializer_list<std::string> a)
    {
        for (const std::string* item = a.begin(); item != a.end(); ++item) {  }
    }
};
int main()
{
    C c{ "hello", "c++", "world" };
}
```

使用列表初始化的注意事项

- 隐式缩窄转换问题

```
int x = 999;
char c1 = x;  // 编译成功，传统变量初始化支持隐式缩窄转换
char c2{ x };
```

- 列表初始化的优先级问题

```
std::vector<int> x1(5, 5);
std::vector<int> x2{ 5, 5 };
```

指定初始化

- 基本语法

```
struct Point {
    int x;
    int y;
};

Point p{ .x = 4, .y = 2 };
```

指定初始化

- 语法要求
  1. 必须是一个聚合类型
  2. 数据成员必须是非静态数据成员
  3. 数据成员最多只能初始化一次
  4. 非静态数据成员的初始化必须按照声明的顺序进行
  5. 针对联合体中的数据成员只能初始化一次，不能同时指定
  6. 不能嵌套指定初始化数据成员
  7. 一旦使用指定初始化就不能混用其他方法对数据成员初始化了
  8. 禁止对数组使用指定初始化

感谢聆听
欢迎关注