# 构造函数

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 修订内容 | 修改人 |
|  | V0.1 | 初稿 | 朱灿基 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 一、概述

构造函数是特殊的成员函数，只要创建类类型的新对象都要执行构造函数。构造函数的工作是保证每个对象的数据成员具有合适的初始值。

当定义一个新类型的时候，需要显式或隐式地指定复制、赋值和撤销该类型的对象时会发生什么——这是通过定义特殊成员：复制构造函数、赋值构造函数和析构函数来达到的。

复制构造函数是一种特殊的构造函数，具有单个形参，该形参（常用const修饰）是对该类类型的应用。当定义一个新对象并用一个同类型的对象对它进行初始化时，将显式使用复制构造函数。当将该类型的对象传递给函数或从函数返回该类型的对象时，将隐式使用复制构造函数。

# 二、构造函数

# 三、复制（拷贝）构造函数

特点：

1. 单个形参
2. 形参是对本类类型对象的引用（常用const修饰）

可用于：

1. 根据另一个同类型的对象显式或隐式初始化一个对象。
2. 将该类型的对象传递给函数时，将隐式使用复制构造函数。（复制一个对象，将它作为实参传给一个函数）
3. 从函数返回该类型的对象时，将隐式使用复制构造函数（从函数返回时复制一个对象）。
4. 初始化顺序容器中的元素。
5. 根据元素初始化列表初始化数组元素。

初始化形式：直接初始化和复制初始化。

直接初始化：直接调用与实参匹配的构造函数（将初始化式放在圆括号里）。

复制初始化：总是调用复制构造函数（使用=符号）。

当形参为非引用类型的时候，将复制实参的值。以非引用类型作返回值时，将返回return语句中的值的副本。

当形参或返回值为类类型时，由复制构造函数进行复制。

|  |
| --- |
| #include <iostream>  using namespace std;  class HasPtrmen  {  public:  HasPtrmen():d(new int(0)){  cout << "Construct: "<<++n\_cstr<< endl;  }  HasPtrmen(const HasPtrmen & h)  :d(new int(\*h.d)){  cout <<"Copy construct: "<<++n\_cptr<< endl;  }  ~HasPtrmen(){  //delete d;  cout << "Destruct: "<<++n\_dstr<<endl;  }  int \* d;  static int n\_cstr;  static int n\_dstr;  static int n\_cptr;  };  int HasPtrmen::n\_cstr = 0;  int HasPtrmen::n\_dstr = 0;  int HasPtrmen::n\_cptr = 0;  HasPtrmen GetTemp()  {  HasPtrmen a;  return a;  }  void GetTempPara(HasPtrmen has)  {  cout << \*has.d << endl;  }  int main()  {  HasPtrmen a = GetTemp();  //GetTempPara(a);  //HasPtrmen b(a);  } |

观察以上代码，再main函数中，分别测试复制构造函数应用场景。

1、根据一个对象初始化另一个对象。

2、函数返回对象。

3、对象作为函数参数。

其中，函数返回对象的场景并没有显式输出预期的值。不确定是什么原因？

# 参考文献