

CONSTRUCTOR AND DESTRUCTOR

构造函数与析构函数

小组成员：

邵东林 :202307885

谭棵 :202306630

苏怡力 :202305962

目录

Contents

01. 思想与好处

concept and benefit

02. 相关语法

Relevant Syntax

03. 示例分析

Example Analysis

04. 知识点总结

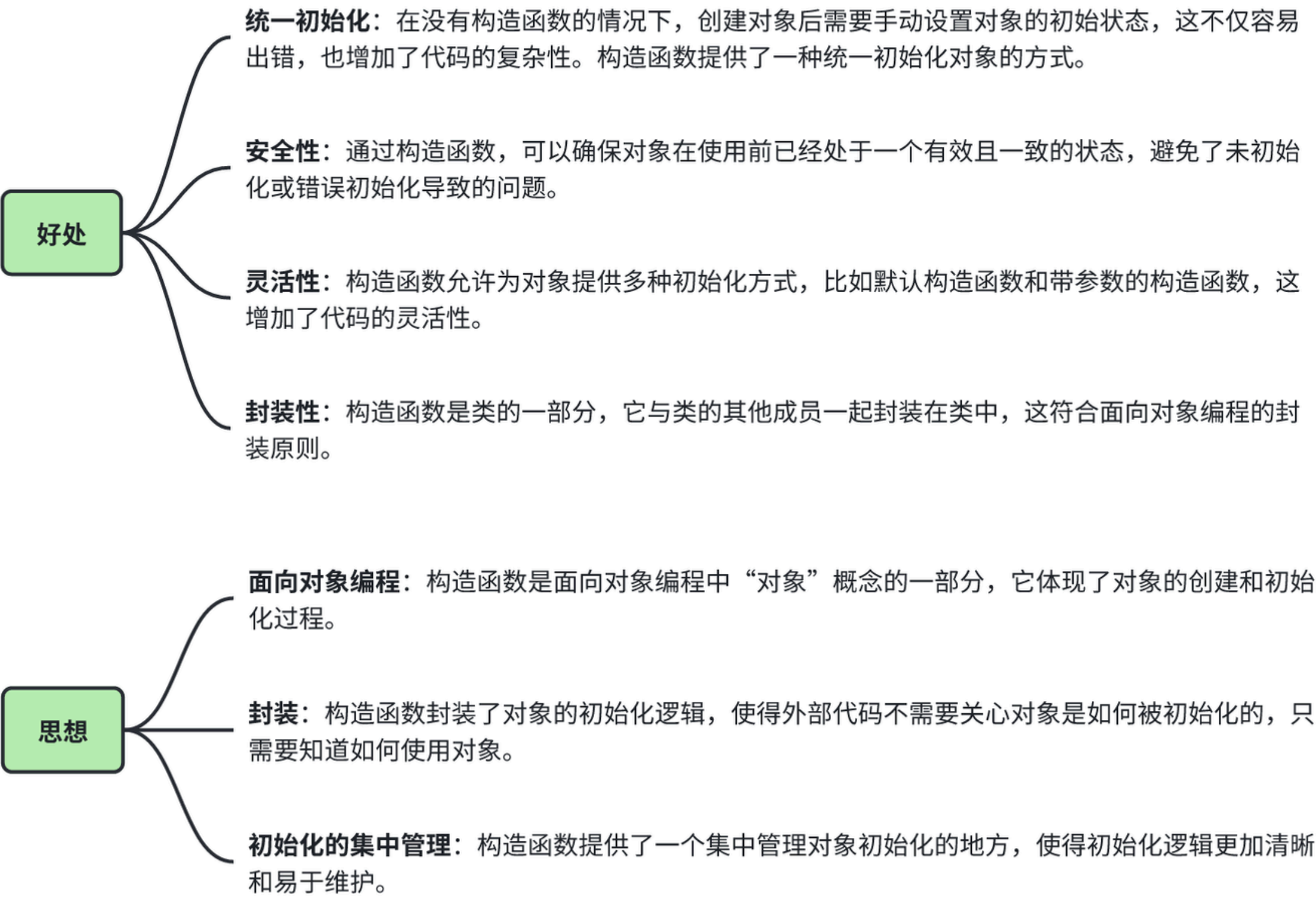
Key Points summary

思想与好处

Background significance of the topic

01.

思想与好处



构造函数

在C++中，类的构造函数是类的一种特殊的成员函数，它会在每次创建类的新对象时执行。

析构函数

与构造函数相对应，析构函数用于在对象生命周期结束时进行清理工作，比如释放资源。

相关语法

Relevant Syntax

02.

1.构造函数

```
class ClassName
{
public:
    // 默认构造函数
    ClassName()
    {
        // 初始化代码
    }

    // 带参数的构造函数
    ClassName(Type1 param1, Type2 param2)    identifier "Type1" is undefined
    {
        // 初始化代码, 使用参数
    }

    // 成员初始化列表
    ClassName(Type1 param1, Type2 param2) : member1(param1), member2(param2)
    {
        // 其他初始化代码
    }
};
```

特点

- 与类同名，没有返回类型。
- 可以重载(Overload) ，以支持不同的初始化方式。
- 默认情况下，编译器会生成默认构造函数，如果用户没有定义任何构造函数。

2.析构函数

```
class ClassName
{
public:
    // 析构函数
    ~ClassName()
    {
        // delete[] dynamicMemory;
        // closeFile();
        // 清理内存等数据的代码
    }
};
```

特点

- 与类同名，前面加波浪号（~），没有参数，没有返回类型。
- 不能重载。
- 默认情况下，编译器会生成默认析构函数，如果用户没有定义任何析构函数。

3.成员初始化列表

```
Line::Line(double len) : length(len)    this declaration has no stor
{
    cout << "Object is being created, length = " << len << endl;
}

Line::Line(double len)
{
    length = len;
    cout << "Object is being created, length = " << len << endl;
}

C::C(double a, double b, double c) : X(a), Y(b), Z(c)
{
    ....
}
```

特点

- 更高效的初始化方式，特别是在初始化常量成员或引用成员时。
- 直接在构造函数调用前初始化成员变量。

示例分析

Example Analysis

03.

示例分析

1. 定义一个类 MyComputer:

- 用于模拟计算机的属性，包括CPU、GPU和内存。
- 包含构造函数和析构函数，分别在对象创建和销毁时执行。

2. 构造函数:

- 初始化计算机的属性(CPU、GPU和内存) 。
- 输出创建计算机时的属性信息。

3. 析构函数:

- 在对象销毁时输出提示信息，表示计算机被销毁。

4. 成员函数:

- 提供 get_cpu()、get_gpu() 和 get_memory() 函数，用于获取计算机的属性值。

5. 主函数 main():

- 创建一个 MyComputer 对象，传入CPU、GPU和内存参数。
- 调用成员函数获取并输出计算机的属性。
- 程序结束时，对象自动销毁，触发析构函数。

```
(t1) d:\algorithm\cpp\mooc\example\Vending Machine\output>.\"1.exe"
Computer is being created
CPU: Intel 13600kf
GPU: NVIDIA RTX 4060Ti
Memory: 16GB
Computer is being destroyed
```

```
class MyComputer
{
public:
    ~MyComputer();           // 这是析构函数声明
    MyComputer(string cpu, string gpu, int memory); // 这是构造函数声明
    string get_cpu();        // 获取CPU信息
    string get_gpu();        // 获取GPU信息
    int get_memory();        // 获取内存信息

private:
    string cpu;
    string gpu;
    int memory;
};

// 成员函数定义，包括构造函数
MyComputer::MyComputer(string cpu, string gpu, int memory) : cpu(cpu), gpu(gpu), memory(memory)
{
    cout << "Computer is being created" << endl;
    cout << "CPU: " << cpu << endl;
    cout << "GPU: " << gpu << endl;
    cout << "Memory: " << memory << "GB" << endl;
}

MyComputer::~MyComputer()
{
    cout << "Computer is being destroyed" << endl;
}

string MyComputer::get_cpu()
{
    return cpu;
}

string MyComputer::get_gpu()
{
    return gpu;
}

int MyComputer::get_memory()
{
    return memory;
}

int main()
{
    MyComputer computer("Intel 13600kf", "NVIDIA RTX 4060Ti", 16);

    return 0;
}
```

知识点总结

Key Points summary

04.

知识点总结

- **函数重载：**C++允许构造函数重载，即一个类可以有多个构造函数，每个构造函数的参数列表不同。这允许对象以不同的方式被初始化。
- **成员初始化列表：**在构造函数中，可以使用成员初始化列表来初始化类的成员变量。这种方式比在构造函数体内赋值更高效，因为它直接在对象创建时初始化成员变量。
- **默认构造函数：**如果用户没有定义任何构造函数，编译器会自动生成一个默认构造函数，它不执行任何操作。如果需要，用户也可以显式定义一个默认构造函数。
- **构造函数的调用：**当创建对象时，构造函数会被自动调用。如果对象是通过直接调用 `new` 操作符创建的，那么在分配内存后，构造函数会被调用以初始化对象。

构造函数就是对象在创建时执行的函数,析构函数就是在对象删除时执行的函数

THANK YOU FOR WATCHING

恳请大家批评指正

小组成员：邵东林 谭棵 苏怡力