

C++核心编程

本阶段主要针对C++面向对象编程技术做详细讲解，探讨C++中的核心和精髓。

1 内存分区模型

C++程序在运行时，将内存大方向划分为4个区域

- 代码区：存放函数体的二进制代码，由操作系统进行管理的
- 全局区：存放全局变量和静态变量以及常量
- 栈区：由编译器自动分配释放，存放函数的参数值、局部变量等
- 堆区：由程序员分配和释放，若程序员不释放，程序结束时由操作系统回收

内存四区意义：

不同区域存放的数据，赋予不同的生命周期，给我们更大的灵活编程

1.1 程序运行前

在程序编译后，生成了exe可执行程序，未执行该程序前分为两个区域

代码区：

- 存放 CPU 执行的机器指令
- 代码区是共享的，共享的目的是对于频繁被执行的程序，只需要在内存中有一份代码即可
- 代码区是只读的，使其只读的原因是防止程序意外地修改了它的指令

全局区：

- 全局变量和静态变量存放在此。
- 全局区还包含了常量区，字符串常量和静态常量也存放在此。
- 该区域的数据在程序结束后由操作系统释放。

栈区：

- 由编译器自动分配释放，存放函数的参数值、局部变量等
- 注意事项：不要返回局部变量的地址，栈区开辟的数据由编译器自动释放

堆区：

- 由程序员分配释放，若程序员不释放，程序结束时由操作系统回收
- 在C++中主要利用new在堆区开辟内存

示例：

```
1 int* func()
2 {
3     int* a = new int(10);
4     return a;
5 }
6
7 int main() {
8
9     int *p = func();
10
11     cout << "p << endl;
12     cout << "p << endl;
13
14     system("pause");
15
16     return 0;
17 }
```

const修饰的全局变量
静态常量：static 数据类型 常量名=值；（static为静态常量关键字）

全局变量：没有写在函数内的变量或其他量
局部变量：写在函数体内的变量

```
15  
16     return 0;  
17 }
```

new操作符

2023年9月18日 17:16

1.3 new操作符

C++中利用new操作符在堆区开辟数据

堆区开辟的数据，由程序员手动开辟，手动释放，释放利用操作符 delete

语法: new 数据类型

利用new创建的数据，会返回该数据对应的类型的指针



//new返回是 该数据类型的指针

示例2: 开辟数组

```
1 //堆区开辟数组
2 int main() {
3
4     int* arr = new int[10];
5
6     for (int i = 0; i < 10; i++)
7     {
8         arr[i] = i + 100;
9     }
10
11     for (int i = 0; i < 10; i++)
12     {
13         cout << arr[i] << endl;
14     }
15     //释放数组 delete 后加 []
16     delete[] arr;
17
18     system("pause");
```

示例1: 基本语法

```
1 int* func()
2 {
3     int* a = new int(10);
4     return a;
5 }
6
7 int main() {
8
9     int *p = func();
10
11     cout << *p << endl;
12     cout << *p << endl;
13
14     //利用delete释放堆区数据
15     delete p;
16
17     //cout << *p << endl; //报错，释放的空间不可访问
18
19     system("pause");
```

new返回的是该数据类型的指针，而指针就是间接记录地址的，可以理解为返回的是该数据的地址。

基本语法： new 数据类型；

例： new int (1) ；即在堆区开辟了一个记录 1 的内存，new会返回 1 的指针来保存 1 的地址。但这样无法调用 1 这个数据，所以要用个指针来接收 1 ；即 int * a =new int (1) ；
开辟动态数组，同样理解

```
Allworker** newspace = new Allworker* [newsizel] ;
```

还可以通过指针进行数组的动态内存分配。可以使用`new`关键字来动态分配数组的内存空间，并返回指向数组的指针。示例代码如下：

```
```cpp
int size = 5;
int *arr = new int[size]; // 动态分配一个大小为5的整型数组
```

指针的嵌套：高级指针记录着低一级指针的地址，理解即可