



链滴

RAII

作者: [xtestw](#)

原文链接: <https://ld246.com/article/1458633922570>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>C++中的RAII全称是“Resource acquisition is initialization”, 直译为“资源获取就是初始化”。其实是一种gc类, 利用类的构造函数和析构数来保证资源获取后的正确释放, 防止内存泄露。</p>

<p>网上找了个例子很好的说明这个用法: </p>

```
<pre class="brush: cpp">void Func()
{
    FILE *fp;
    char* filename = "test.txt";
    if((fp=fopen(filename,"r"))==NULL)
    {
        printf("not open");
        exit(0);
    }
    ... // 如果 在使用fp指针时产生异常 并退出
        // 那么 fp文件就没有正常关闭
```

fclose(fp);

</pre>

<p>而使用RAII,则是如下的方式: </p>

```
<pre class="brush: cpp">class FileRAII{
public:
    FileRAII(FILE* aFile):file_(aFile){
        ~FileRAII() { fclose(file_); } //在析构函数中进行文件关闭
    FILE* get() {return file_;}
private:
    FILE* file_;
};
```

void Func()

{

FILE fp;

char filename = "test.txt";

if((fp=fopen(filename,"r"))==NULL)

{

printf("not open");

exit(0);

}

FileRAII fileRAII(fp);

... // 如果 在使用fp指针时产生异常 并退出

// 那么 fileRAII在栈展开过程中会被自动释放, 析构函数也就会自动地将fp关闭

// 即使所有代码是都正确执行了, 也无需手动释放fp, fileRAII它的生命期在此结束时, 它的析构函数会自动执行!

</pre>

<p> </p>

<p> </p>