//虚析构和纯虚析构

#include<iostream>

#include<string>

using namespace std;

class animal {

public:

animal() {

cout << "animal构造函数" << endl;

}

//利用虚析构可以解决父类指针释放子类对象不干净问题

/\*virtual ~animal() {

cout << "animal析构函数" << endl;

}\*/

virtual ~ animal() = 0;//纯虚析构

//有了纯虚析构，这个类就属于抽象类

//纯虚函数

virtual void speak() = 0;

};

animal:: ~animal() {

cout << "animal纯析构函数" << endl;

}

class cat : public animal {

public:

cat(string name) {

cout << "cat构造函数" << endl;

mname= new string(name);

}

virtual void speak() {

cout << "cat is speaking" << endl;

}

~cat() {

if (mname != NULL) {

cout << "cat析构函数" << endl;

delete mname;

//父类指针在析构时不会调用子类析构函数，导致子类如果有堆区属性，会出现内存泄露

mname = NULL;

}

}

string \*mname;

};

void test01() {

animal\* animal1 = new cat("tom");

animal1->speak();

delete animal1;

}

int main() {

test01();

system("pause");

return 0;}

