概念题

1. 什么是对象？什么是类？二者之间的关系是怎样的？

对象属于值，是程序运⾏时的实体，是由数据及能对其施加的操作构成的封装体，是⾯向对象编程程序的基本单位。

类是静态的程序实体，⽤于描述对象的特征，通常包含数据成员和成员函数定义。

通过类来创建对象，⼀个类可以创建多个对象。⽽类是⼀组对象的抽象，描述了⼀组具有相同特性和⾏为的对象。

1. 面向对象有哪些特征？相比于面向过程，它有哪些优点？特征：抽象封装、继承、多态

优点：更容易维护、扩展、复用，结构更清晰，安全性更高，可读性更高

1. 举例说明什么情况下需要自定义析构函数？析构函数中归还的资源包括哪些？不包括哪些？

当对象创建后还额外申请了资源时需要⾃定义析构函数来归还。

析构函数归还的资源只需要包括对象额外申请的资源。

例如将指针指向新分配的内存空间需要在析构时归还：

int len;

char \*str; //系统分配

String(char\* s)

{

len = strlen(s);

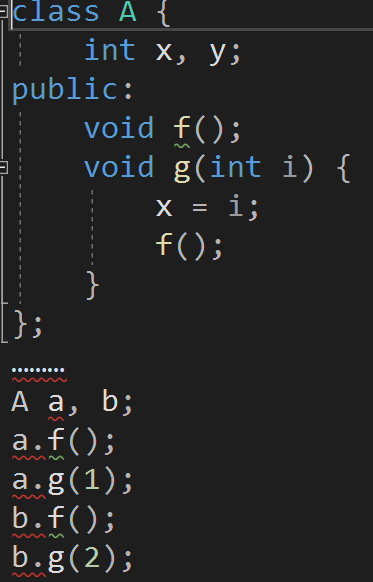
str = new char[len+1]; //申请的额外空间需要在析构的时候归还

}

1. 简述C++中this指针的作用
   1. 区别成员变量和局部变量（变量名相同）
   2. 用于return当前对象
   3. 指向自身，实现链式操作

编程题

1. 文本

   描述已自动生成
2. 
3. class Deque {

private:

struct Node {

A data;

Node\* next;

Node\* prev;

Node(const A& item) : data(item), next(nullptr), prev(nullptr) {}

};

Node\* frontNode;

Node\* rearNode;

public:

Deque() : frontNode(nullptr), rearNode(nullptr) {}

~Deque() {

while (!isEmpty()) {

popFront();

}

}

void pushFront(const A& item) {

Node\* newNode = new Node(item);

if (isEmpty()) {

frontNode = rearNode = newNode;

}

else {

newNode->next = frontNode;

frontNode->prev = newNode;

frontNode = newNode;

}

}

void pushBack(const A& item) {

Node\* newNode = new Node(item);

if (isEmpty()) {

frontNode = rearNode = newNode;

}

else {

newNode->prev = rearNode;

rearNode->next = newNode;

rearNode = newNode;

}

}

void popFront() {

if (!isEmpty()) {

Node\* temp = frontNode;

frontNode = frontNode->next;

delete temp;

if (frontNode == nullptr) {

rearNode = nullptr;

}

else {

frontNode->prev = nullptr;

}

}

}

void popBack() {

if (!isEmpty()) {

Node\* temp = rearNode;

rearNode = rearNode->prev;

delete temp;

if (rearNode == nullptr) {

frontNode = nullptr;

}

else {

rearNode->next = nullptr;

}

}

}

bool isEmpty() const {

return frontNode == nullptr;

}

};