

计算机二级C基础:虚拟构造函数的用处计算机二级考试 PDF
转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/608/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_608661.htm

从字面来看，谈论“虚拟构造函数”没有意义。当有一个指针或引用，但是不知道其指向对象的真实类型是什么时，可以调用虚拟函数来完成特定类型（type-specific）对象的行为。仅当还没拥有一个对象但是又确切地知道想要的对象的类型时，才会调用构造函数。那么虚拟构造函数又从何谈起呢？很简单。尽管虚拟构造函数看起来好像没有意义，其实它们有非常大的用处。例如，假设编写一个程序，用来进行新闻报道的工作，每一条新闻报道都由文字或图片组成。可以这样管理它们：

```
class NLComponent { //用于 newsletter components public:// 的抽象基类 ... //包含至少一个纯虚函数 }. class TextBlock: public NLComponent { public: ... // 不包含纯虚函数 }. class Graphic: public NLComponent { public: ... // 不包含纯虚函数 }. class Newsletter { // 一个 newsletter 对象 public:// 由NLComponent 对象 ... // 的链表组成 private: list components. }. 在Newsletter中使用的list类是一个标准模板类（STL）。list类型对象的行为特性有些象双向链表，尽管它没有以这种方法来实现。对象NewLetter不运行时就会存储在磁盘上。为了能够通过位于磁盘的替代物来建立Newsletter对象，让NewLetter的构造函数带有istream参数是一种很方便的方法。当构造函数需要一些核心的数据结构时，它就从流中读取信息：
```

```
class Newsletter { public: Newsletter(istream& str) { while (str) { 从str读取下一个component对象. 把对象加入到newsletter的 components对象
```

的链表中去。}} 或者，把这种技巧用于另一个独立出来的函数叫做readComponent，如下所示：

```
class Newsletter { public: ...
private: // 为建立下一个NLComponent对象从str读取数据, // 建立component并返回一个指针。 static NLComponent *
readComponent(istreamamp. str) { while (str) { //
把readComponent返回的指针添加到components链表的最后，
//\"push_back\" 一个链表的成员函数，用来在链表最后进行
插入操作。 components.push_back(readComponent(str)). }} 考
虑一下readComponent所做的工作。它根据所读取的数据建立
了一个新对象，或是TextBlock或是Graphic。因为它能建立新
对象，它的行为与构造函数相似，而且因为它能建立不同类
型的对象，我们称它为虚拟构造函数。虚拟构造函数是指能
够根据输入给它的数据的不同而建立不同类型的对象。虚拟
构造函数在很多场合下都有用处，从磁盘（或者通过网络连
接，或者从磁带机上）读取对象信息只是其中的一个应用。
还有一种特殊种类的虚拟构造函数——虚拟拷贝构造函数——
也有着广泛的用途。虚拟拷贝构造函数能返回一个指针，
指向调用该函数的对象的新拷贝。因为这种行为特性，虚拟
拷贝构造函数的名字一般都是copySelf，cloneSelf或者是象下
面这样就叫做clone。很少会有函数能以这么直接的方式实现
它：
```

```
class NLComponent { public: // declaration of virtual copy
constructor virtual NLComponent * clone() const = 0. ... }. class
TextBlock: public NLComponent { public: virtual TextBlock *
clone() const// virtual copy { return new TextBlock(*this). } //
constructor ... }. class Graphic: public NLComponent { public:
virtual Graphic * clone() const// virtual copy { return new
```

Graphic(*this). } // constructor ... }. 100Test 下载频道开通，各类
考试题目直接下载。详细请访问 www.100test.com