

作业五

概念题

1. 请阐述 C++ 中动态绑定和静态绑定的概念，并说明在什么情况下会发生动态绑定。

- 1 静态绑定是在编译时确定函数调用的实际地址，而动态绑定是在运行时根据对象的实际类型确定函数调用的实际地址。
- 2 动态绑定通常发生在使用基类指针或引用调用虚函数的情况下，此时程序会根据对象的实际类型来确定调用哪个函数。

2. 为什么析构函数一般要声明为虚函数？

- 1 析构函数一般要声明为虚函数，是为了确保在通过基类指针或引用删除派生类对象时，能够调用到派生类的析构函数。如果基类的析构函数不是虚函数，那么当通过基类指针或引用删除派生类对象时，只会调用基类的析构函数，而不会调用派生类的析构函数，这可能导致资源泄漏或未定义的行为。

3. 为什么基类的构造函数和析构函数中对虚函数的调用不进行动态绑定？

- 1 在基类的构造函数和析构函数中对虚函数的调用不进行动态绑定的原因是，在对象的构造和析构过程中，对象的类型是已知的。在对象构造期间，对象的类型是正在构造的类，而不是派生类。同样，在对象析构期间，对象的类型也是已知的，即正在析构的类。因此，在构造函数和析构函数中调用虚函数时，会根据对象的类型直接调用对应的函数实现，而不需要进行动态绑定。

编程题

1. 假设你正在开发一个简单的动物园管理系统。在这个系统中，有不同类型的动物，每种动物都有一些共同的属性和行为，例如名称、年龄、颜色，以及发出声音。你需要使用 C++ 中的类来描述这些动物。

要求：

a. 创建一个名为 `Animal` 的基类，其中包含以下属性和虚函数：

- 名称 (`name`)
- 年龄 (`age`)

- 颜色 (color)
 - 一个虚函数 void makeSound() , 用于输出该动物发出的声音。
- b. 创建至少两个派生类, 例如 Cat 和 Dog , 它们继承自 Animal 类, 并重写 makeSound() 函数以输出特定于该动物的声音。

```
1  # include <iostream>
2
3  class Animal {
4  protected:
5      int name;
6      int age;
7      int color;
8  public:
9      void makeSound() {
10         std::cout << "Ready" << std::endl;
11     }
12     Animal(){}
13     Animal(int name, int age, int color) {
14         this->name = name;
15         this->age = age;
16         this->color = color;
17     }
18 };
19
20 class Dog : public Animal {
21 public:
22     void makeSound() {
23         std::cout << "woof, woof" << std::endl;
24     }
25     Dog(int name, int age, int color) : Animal(name, age, color)
26     { }
27     int get_age() {
28         return this->age;
29     }
30 };
31
32 class Cat : protected Animal {
33 public:
34     void makeSound() {
35         std::cout << "meow, meow" << std::endl;
```

```
36     Cat(int name, int age, int color) : Animal(name, age, color)
37     {}
38 };
39 int main(int arg_number, char **arg_value) {
40     Animal a(1, 2, 3);
41     Dog d(1, 2, 3);
42     Cat c(1, 2, 3);
43     d.makeSound();
44     c.makeSound();
45     std::cout << d.get_age() << std::endl;
46     return 0;
47 }
```

提交注意事项

截止时间：2024-4-8 23:59

文件格式：姓名-学号.pdf

提交方式：[南大计科在线实验教学平台](#)

请同学们于截止时间前在南大计科在线实验教学平台上提交，每次作业最终只需要提交一个pdf文件即可，以“姓名-学号.pdf”的方式命名。

注意：

请按要求命名文件，并且只提交一个PDF文件，编程题代码请附在PDF中。任何错误的命名和文件格式将影响你的作业得分。