

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #pragma once
3 #include<iostream>
4
5
6 class Animal
7 {
8 private:
9     char* name;
10    int age;
11
12 public:
13    Animal()
14    {
15        name = NULL;
16        age = 0;
17    }
18    Animal(const char * name, const int age = 0)
19    {
20        int len = strlen(name);
21        this->name = new char[ len + 1];
22        strcpy( this->name, name);
23
24        this->age = age;
25    }
26
27    virtual ~Animal()
28    {
29        delete[] name;
30    }
31
32    char* GetName();
33    int GetAge();
34    virtual void Sound (){}
35    virtual void Eat() {}
36    virtual char* GetType()
37    {
38        return 0;
39    }
40    void SetName(char* _name);
41    virtual void Sound3() {}
42    virtual void Sound2() {}
43 };
44
45
46
47 class Dog : public Animal
48 {
49     char* type;
50 public:
51     Dog(const char* name, const int age) : Animal(name, age)
52     {
53         int len = strlen("Dog");
54         type = new char[ len + 1];
55         strcpy(type, "Dog");
56     }
```

```
57     Dog() : Animal()
58     {
59         int len = strlen("Dog");
60         type = new char[len + 1];
61         strcpy(type, "Dog");
62     }
63     ~Dog() {
64         delete[] type;
65     }
66
67     char* GetType();
68     void Sound();
69     void Eat();
70     void Sound3();
71     void Sound2();
72 };
73
74
75 class Cat : public Animal
76 {
77     char* type;
78 public:
79     Cat(const char* name, const int age) : Animal(name, age)
80     {
81         int len = strlen("Cat");
82         type = new char[len + 1];
83         strcpy(type, "Cat");
84     }
85
86     Cat() : Animal()
87     {
88         int len = strlen("Cat");
89         type = new char[len + 1];
90         strcpy(type, "Cat");
91     }
92
93     ~Cat() {
94         delete[] type;
95     }
96     void Sound3();
97     void Sound2();
98     char* GetType();
99     void Sound();
100    void Eat();
101
102 };
```