#include <iostream>

using namespace std;

//class Base

//{

//protected:

// int value;

//public:

// Base(int value)

// {

// this->value = value;

// }

// int GetValue() const

// {

// return value;

// }

// virtual string GetName()

// {

// return "I am base";

// }

//};

//

//class Derived :public Base

//{

//public:

// Derived(int value) : Base(value)

// {

//

// }

// int GetDoubleValue()

// {

// return value \* 2;

// }

// string GetName()

// {

// return "I am derived";

// }

//};

//

//int main()

//{

// Base b(10);

//

// Base\* base = new Derived(20);

// cout << base->GetName() << endl;

//

// return 0;

//}

class Animal

{

protected:

string name;

Animal(const string& name)

{

this->name = name;

}

public:

string GetName() const

{

return name;

}

virtual void Speak()

{

cout << "????????????????" << endl;

}

};

class Cat :public Animal

{

public:

Cat(const string& name) :Animal(name)

{

}

void Speak()

{

cout << "meaouououuou" << endl;

}

void Run()

{

cout << "I can run like cat" << endl;

}

};

class Dog:public Animal

{

public:

Dog(const string& name) :Animal(name)

{

}

void Speak()

{

cout << "hahvhahavhah, I can bark like real dog" << endl;

}

void Barkbark()

{

cout << "I can just bark" << endl;

}

};

class Elephant : public Animal

{

public:

Elephant(const string& name) : Animal(name)

{

}

void Speak()

{

cout << "Bruwwwwww, men filem" << endl;

}

};

//void CallMethodDog(Dog dog)

//{

// dog.Speak();

//}

//

//void CallMethodCat(Cat cat)

//{

// cat.Speak();

//}

void CallMethod(Animal\* animal)

{

animal->Speak();

}

int main()

{

Animal\* cat = new Cat("Mestan");

Animal\* dog = new Dog("Toplan");

Animal\* elephant = new Elephant("Fred");

//cat->Speak();

//dog->Speak();

//elephant->Speak();

//CallMethod(cat);

//CallMethod(dog);

//CallMethod(elephant);

Animal\*\* animals = new Animal \* [3]{cat,dog,elephant};

for (int x = 0; x < 3; x++)

{

animals[x]->Speak();

}

for (int x = 0; x < 3; x++)

{

CallMethod(animals[x]);

}

return 0;

}