Объектно-ориентированное программирование

Множественное наследование

Множественное наследование

Множественное наследование

- Множественное наследование означает, что класс имеет несколько базовых классов.
- При этом, если в базовых классах есть одноименные элементы, может произойти конфликт идентификаторов, который устраняется с помощью операции доступа к области видимости.

Пример реализации множественного наследования

```
class A
{public:
        void doSmth(){ /*...*/ }
};

class   B
{public:
      void doSmth(){ /*...*/ }
      void doSmthElse(){ /*...*/ }
};
class C: public A, public B { /*...*/ };
```

Проблемы множественного наследования

- Использование в программе метода, определенного в обоих базовых классах приведет к ошибке, поскольку компилятор не в состоянии разобраться, метод какого из базовых классов требуется вызвать.

```
C some;
some.B::doSmth();
```

- Если у базовых классов есть общий предок, это приведет к тому, что производный от этих базовых класс унаследует два экземпляра полей предка, что чаще всего является нежелательным.

Множественное наследование

```
class Animal
{ public:
  virtual void eat();  // Метод определяется для данного класса
};

class Mammal : public Animal
{ public:
  virtual Color getHairColor();
};

class WingedAnimal : public Animal
{ public:
  virtual void flap();
};

// A bat is a winged mammal
class Bat : public Mammal, public WingedAnimal {};
// обратите внимание, что метод eat() не переопределен в Bat

Bat bat;
```

Как ест летучая мышь?

- Но как летучая мышь ест? Что происходит при вызове метода `bat.eat()`?
- у каждого наследника (`WingedAnimal, Mammal`) метод `eat()` определен по-своему.
- «Ромбическое наследование» (Diamond Inheritance):
  - сущность Animal единственна по сути;
  - `Bat` — это `Mammal` и `WingedAnimal`
  - но свойство животности (`Animalness`) летучей мыши (`Bat`), оно же свойство животности млекопитающего (`Mammal`) и оно же свойство животности `WingedAnimal` — по сути это одно и то же свойство.

Представление классов в памяти

- При наследовании классы предка и наследника просто помещаются в памяти друг за другом.
- Таким образом объект класса Bat это на самом деле последовательность объектов классов (Animal,Mammal,Animal,WingedAnimal,Bat), размещенных последовательно в памяти.
- При этом Animal повторяется дважды, что и приводит к неоднозначности.

Виртуальное множественное наследование

- Чтобы избежать дублирования, требуется при наследовании общего предка определить его как виртуальный класс

```
class monster { ... };
class daemon: virtual public monster { ... };
class lady: virtual public monster { ... };
class god: public daemon, public lady { ... };
```

- Класс god содержит только один экземпляр полей класса monster.

Множественное наследование

```
class Animal
{
 public:
  virtual void eat();
};

// Two classes virtually inheriting Animal:
class Mammal : public virtual Animal {
 public:
  virtual Color getHairColor();
};

class WingedAnimal : public virtual Animal
{
 public:
  virtual void flap();
};

// A bat is still a winged mammal
class Bat : public Mammal, public WingedAnimal {};
```

Представление классов в памяти

- Теперь, часть Animal объекта класса Bat::WingedAnimal *та же самая*, что и часть Animal, которая используется в Bat::Mammal, и можно сказать, что Bat имеет в своем представлении только одну часть Animal и вызов Bat::eat() становится однозначным.
- Виртуальное наследование реализуется через добавление указателей на виртуальную таблицу vtable в классы Mammal и WingedAnimal
- Таким образом, Bat представляется, как (vtable*, Mammal, vtable*, WingedAnimal, Bat, Animal).

Особенности виртуального наследования

- Размер объектов классов, использующих множественное виртуальное наследование, больше по сравнению со случаем, когда виртуальное наследование не используется.
- Доступ к данным-членам виртуальных базовых классов также медленнее, чем к данным невиртуальных базовых классов.
- Не применяйте виртуальных базовых классов до тех пор, пока в этом не возникнет настоятельная потребность. По умолчанию используйте невиртуальное наследование.
- Если все же избежать виртуальных базовых классов не удается, старайтесь не размещать в них данных (классы-интерфейсы)