

Абстрактный класс

Абстрактный класс – такой класс, который не может иметь непосредственных экземпляров.

Абстрактный метод – виртуальный метод, не имеющий реализации.

Класс, имеющий хотя бы один абстрактный является абстрактным.

Пример абстрактного класса

```
class Shape{
public:
    virtual ~Shape () {}
    virtual bool draw () const = 0; //абстрактный метод
};

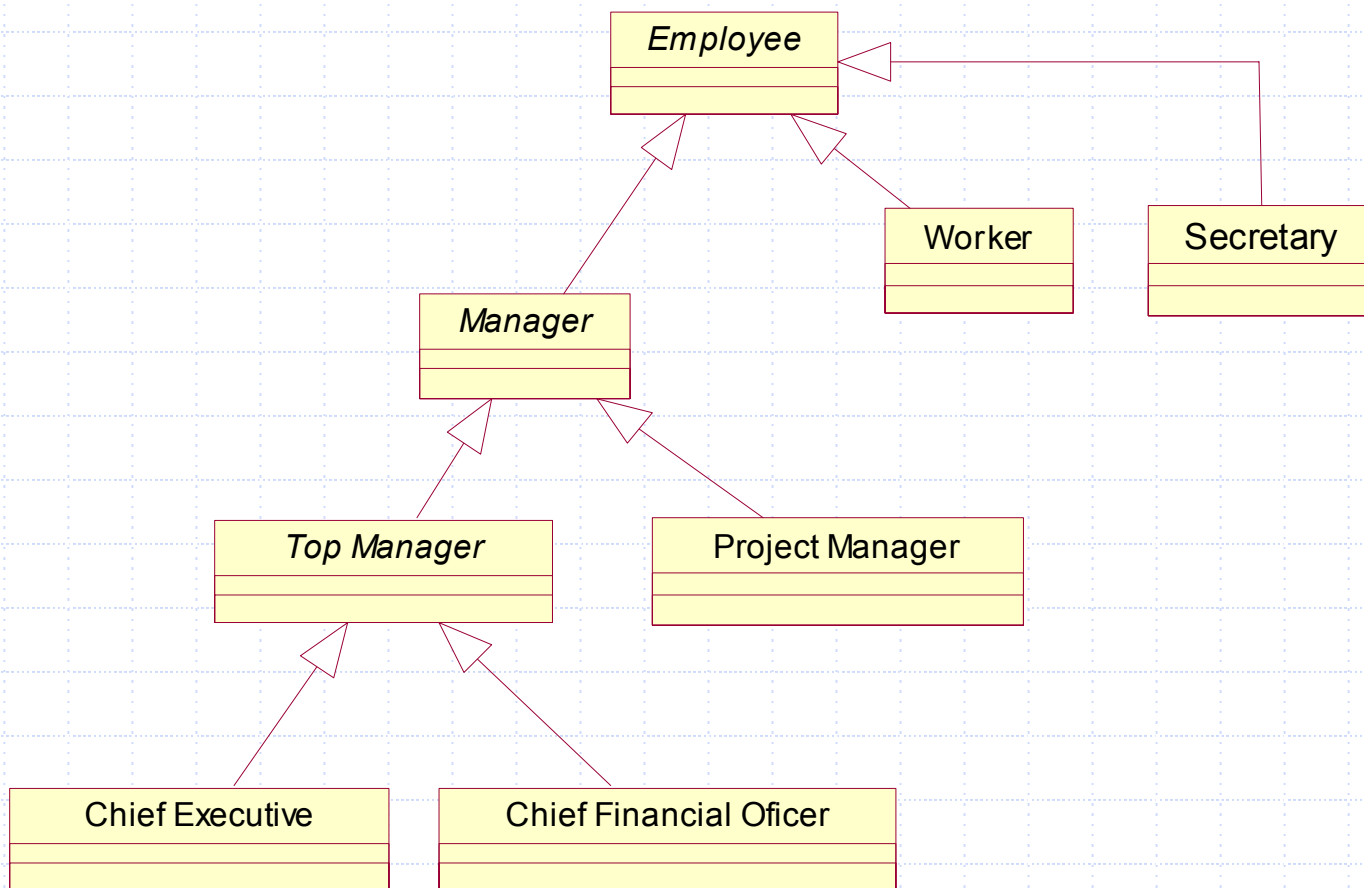
class Circle : public Shape{
public:
    virtual bool draw () const;
    virtual ~Circle ();
};

bool Circle::draw () const{
    //...
}
```

Вызов абстрактных методов

- В созданном объекте абстрактный метод получает реализацию и может быть вызван
- Во время исполнения конструкторов или деструкторов абстрактный метод не имеет реализации и не может быть вызван

Иерархии классов

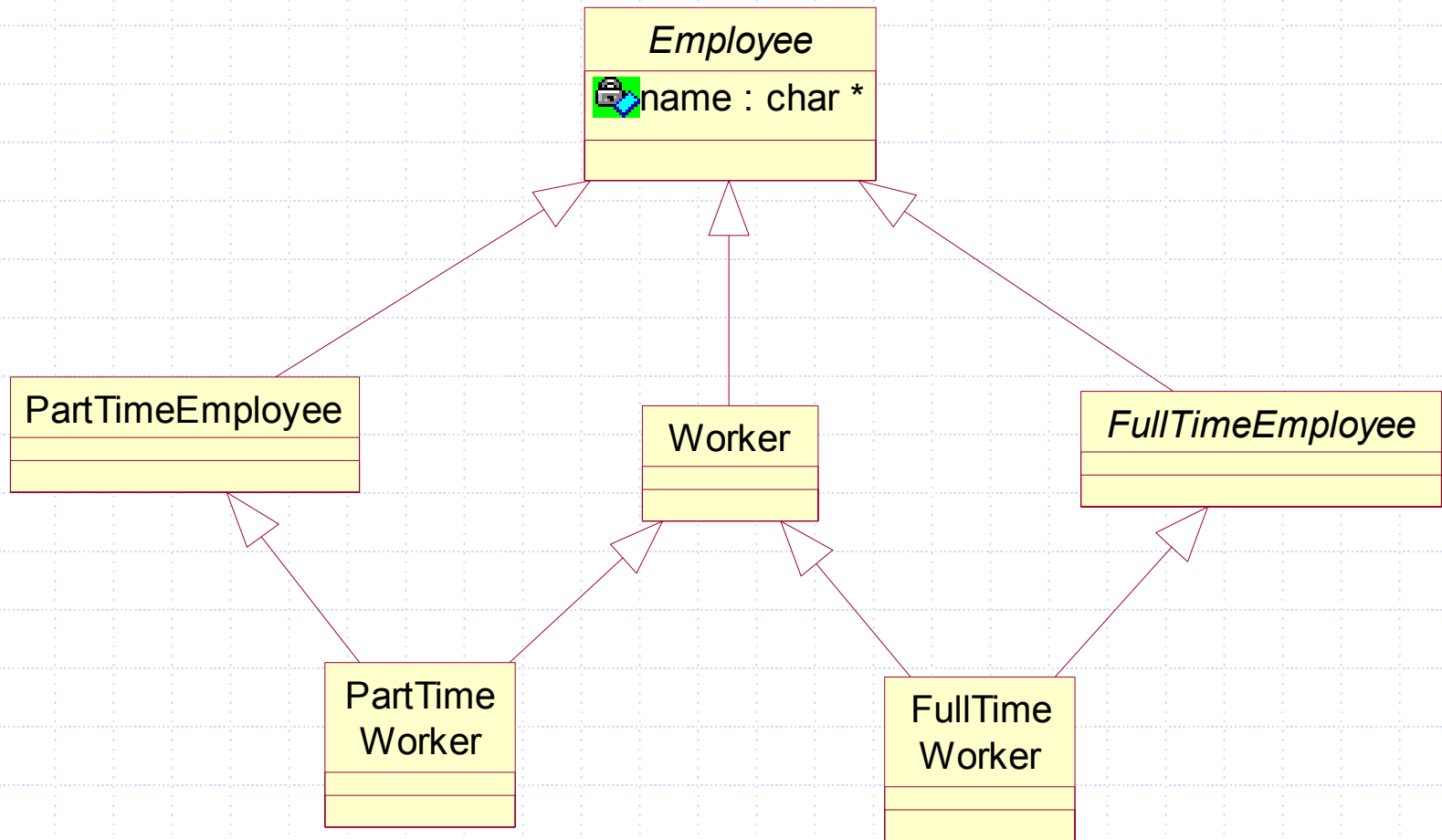


Интерфейс

Интерфейс – сущность, подобная классу, описывающая, что должен делать класс, который реализует ее, но не определяющая это поведение.

В C++ интерфейс – это абстрактный класс, в котором все методы абстрактны, а атрибуты отсутствуют.

Множественное наследование



Реализация множественного наследования

```
class Employee{
public:
    char * name;
//...};

class PartTimeEmployee: public Employee{
//...};

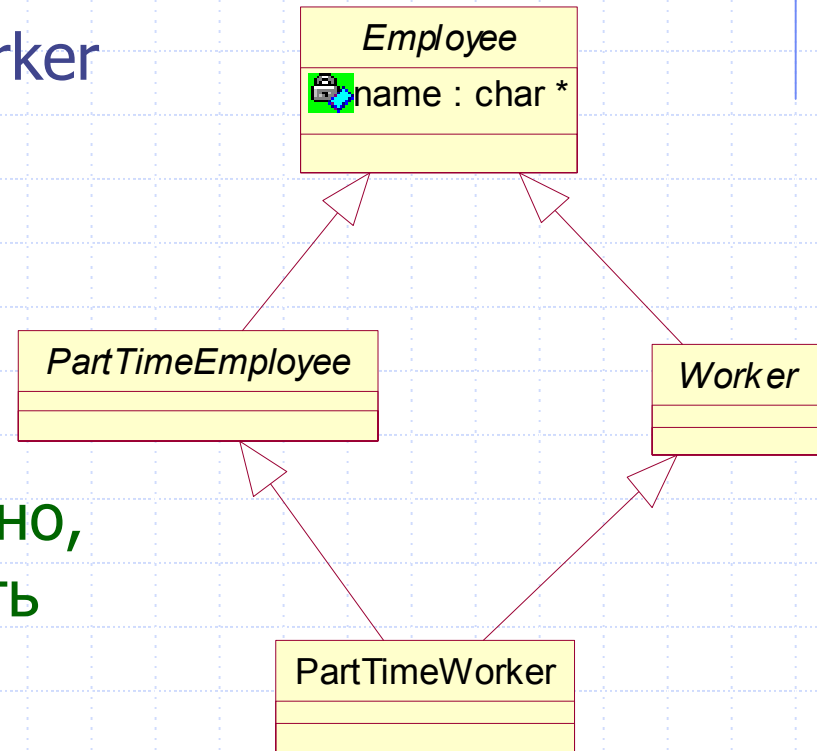
class Worker: public Employee{
//...};

class PartTimeWorker: public Worker, public PartTimeWorker{
//...};
```

Ромбовидная иерархия

Проблемы:

1. Объект класса `PartTimeWorker` содержит в себе 2 объекта класса `Employee`
2. При приведении объекта класса `PartTimeWorker` к классу `Employee` непонятно, какой из объектов выбирать



Виртуальные базовые классы

```
class Employee{  
public:  
    char * name;  
//...};  
  
class PartTimeEmployee: virtual public Employee{  
//...};  
  
class Worker: virtual public Employee{  
//...};  
  
class PartTimeWorker: public Worker, public PartTimeWorker{  
//...};
```

Множественное наследование: размышления на тему

При использовании множественного наследования существует вероятность образования ромбовидной иерархии

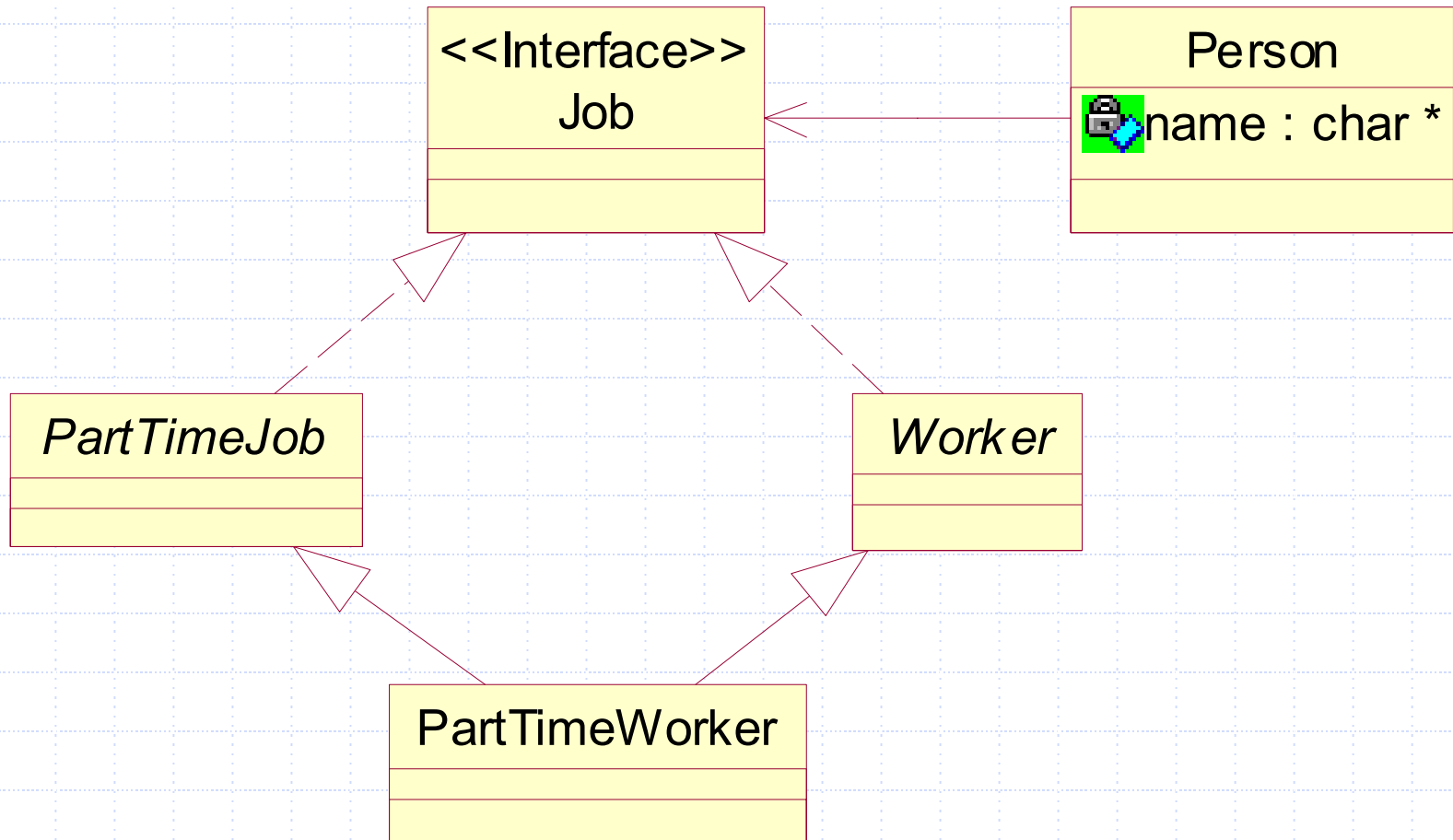
При расширении иерархии классов с множественным наследованием количество классов растет нелинейно

В большинстве случаев множественное наследование является следствием неправильной декомпозиции системы и/или ошибками проектирования

Решения проблем множественного наследования

1. Замена классов интерфейсами – переход от наследования в реализации интерфейса
2. Разделение иерархии на несколько независимых

Замена класса интерфейсом



Разделение иерархий

