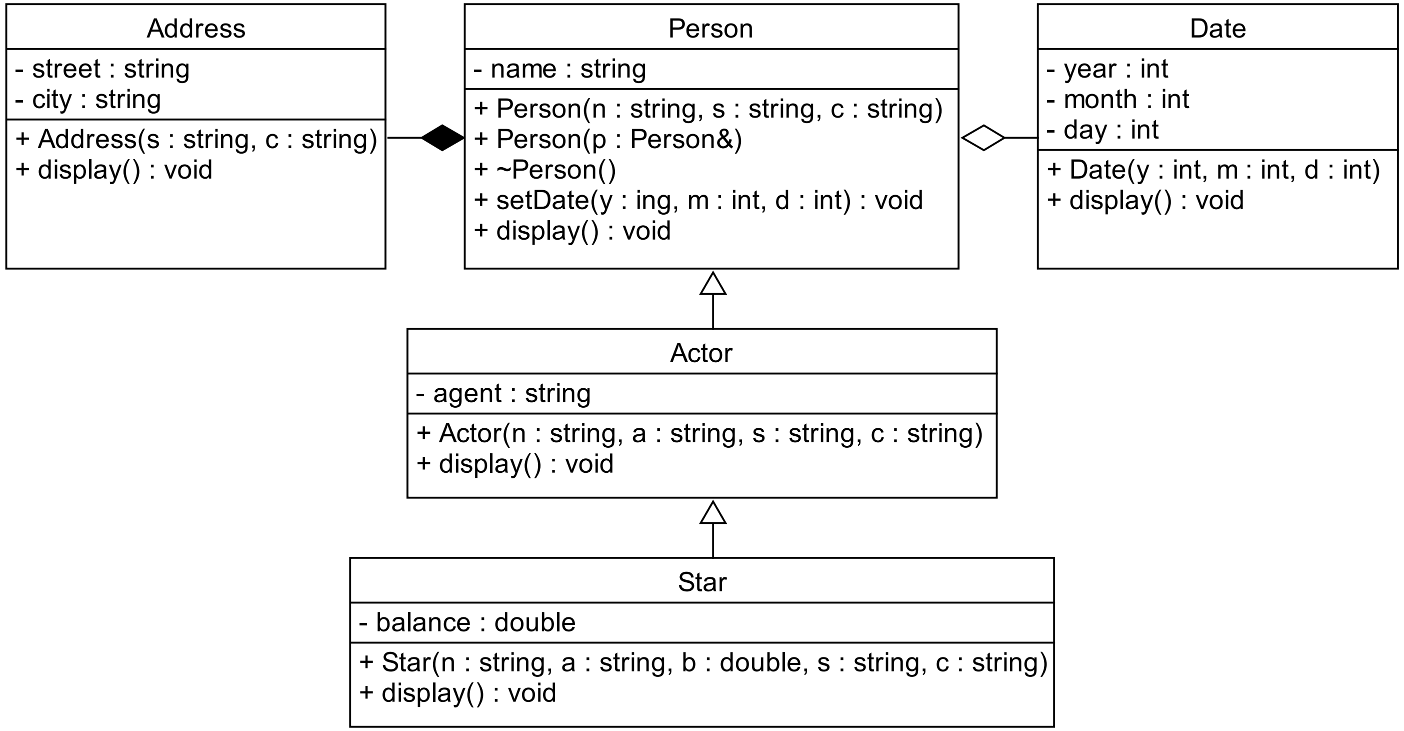
ACTOR 3

Multi-class, Multi-relational Example

Delroy A. Brinkerhoff

ACTOR 3 UML CLASS DIAGRAM



Address
- street : string
- city : string
+ Address(s : string, c : string)
+ display() : void

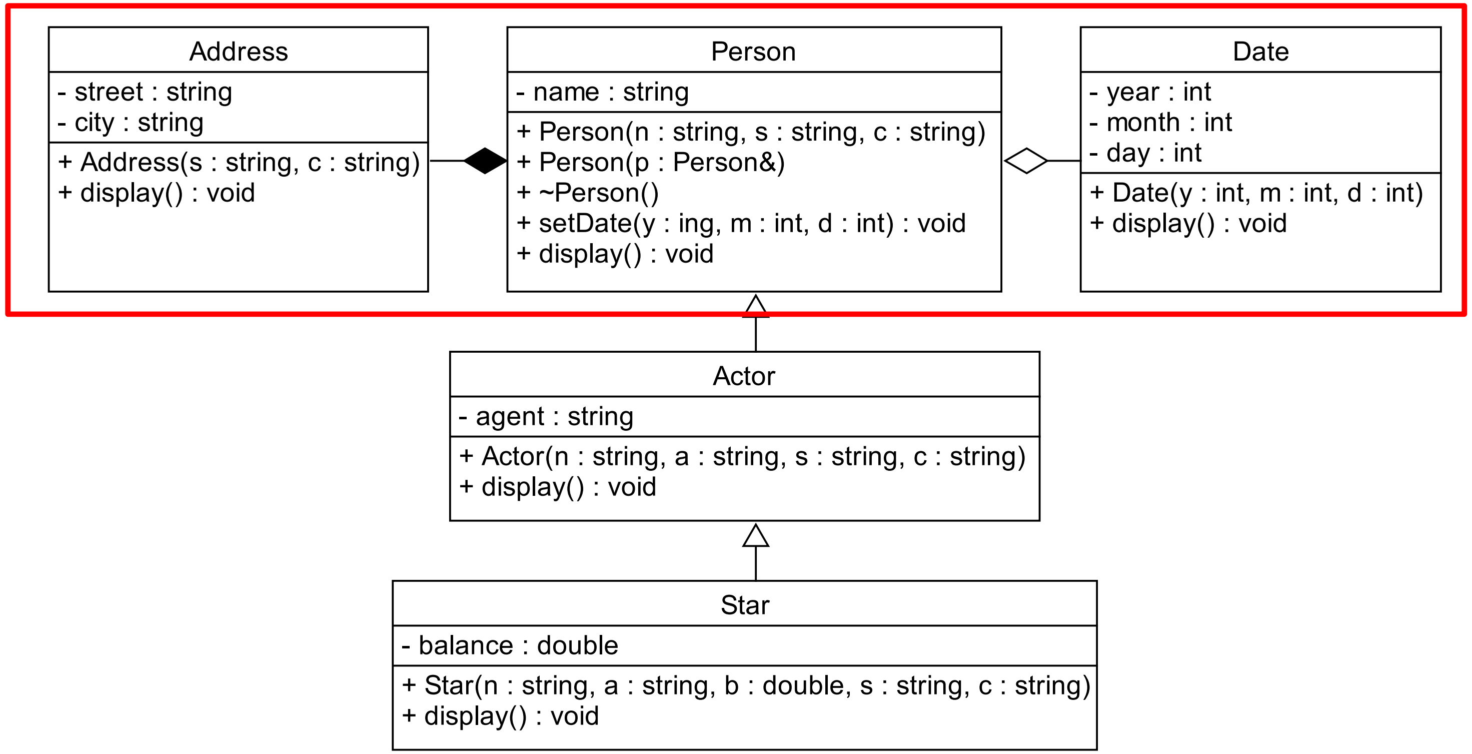
Person
- name : string
+ Person(n : string, s : string, c : string)
+ Person(p : Person&)
+ ~Person()
+ setDate(y : ing, m : int, d : int) : void
+ display() : void

Date
- year : int
- month : int
- day : int
+ Date(y : int, m : int, d : int)
+ display() : void

Actor
- agent : string
+ Actor(n : string, a : string, s : string, c : string)
+ display() : void

Star
- balance : double
+ Star(n : string, a : string, b : double, s : string, c : string)
+ display() : void

ACTOR 3 WHOLE-PART



Address
- street : string
- city : string
+ Address(s : string, c : string)
+ display() : void

Person
- name : string
+ Person(n : string, s : string, c : string)
+ Person(p : Person&)
+ ~Person()
+ setDate(y : ing, m : int, d : int) : void
+ display() : void

Date
- year : int
- month : int
- day : int
+ Date(y : int, m : int, d : int)
+ display() : void

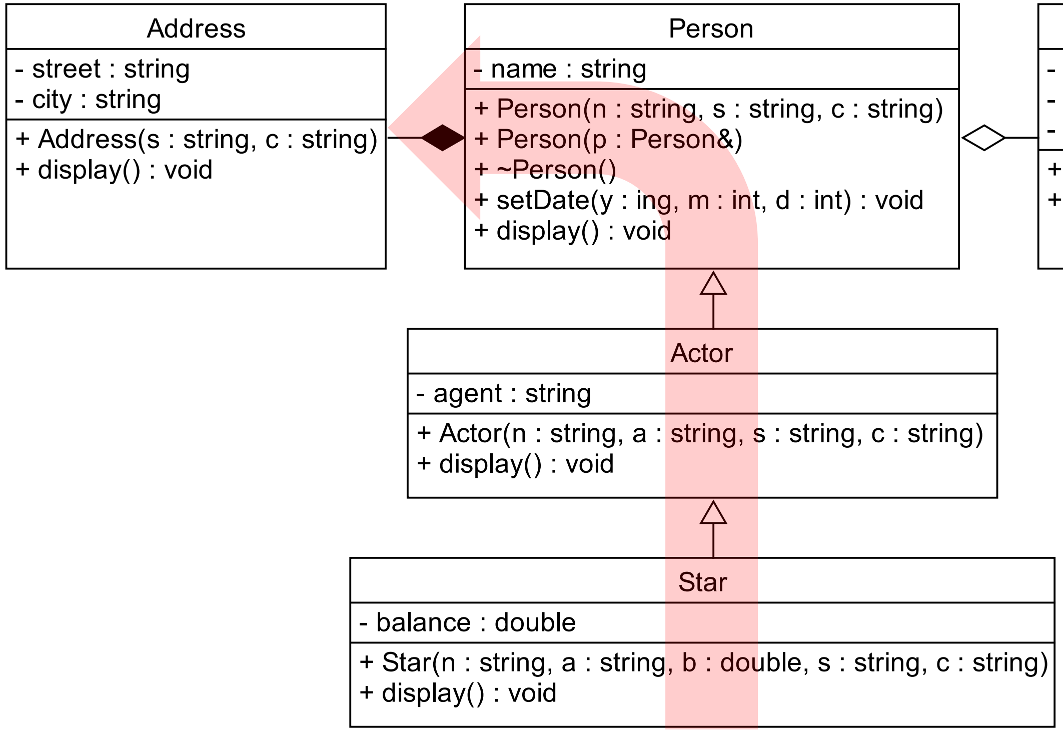
Actor
- agent : string
+ Actor(n : string, a : string, s : string, c : string)
+ display() : void

Star
- balance : double
+ Star(n : string, a : string, b : double, s : string, c : string)
+ display() : void

Person.h (1)

```
#pragma once

#include "Address.h"
#include "Date.h"
#include <iostream>
#include <string>
using namespace std;

class Person
{
    private:
        string  name;
        Address addr;
        Date*   date = nullptr;
```

ACTOR 3 CONSTRUCTORS



Address
- street : string
- city : string
+ Address(s : string, c : string)
+ display() : void

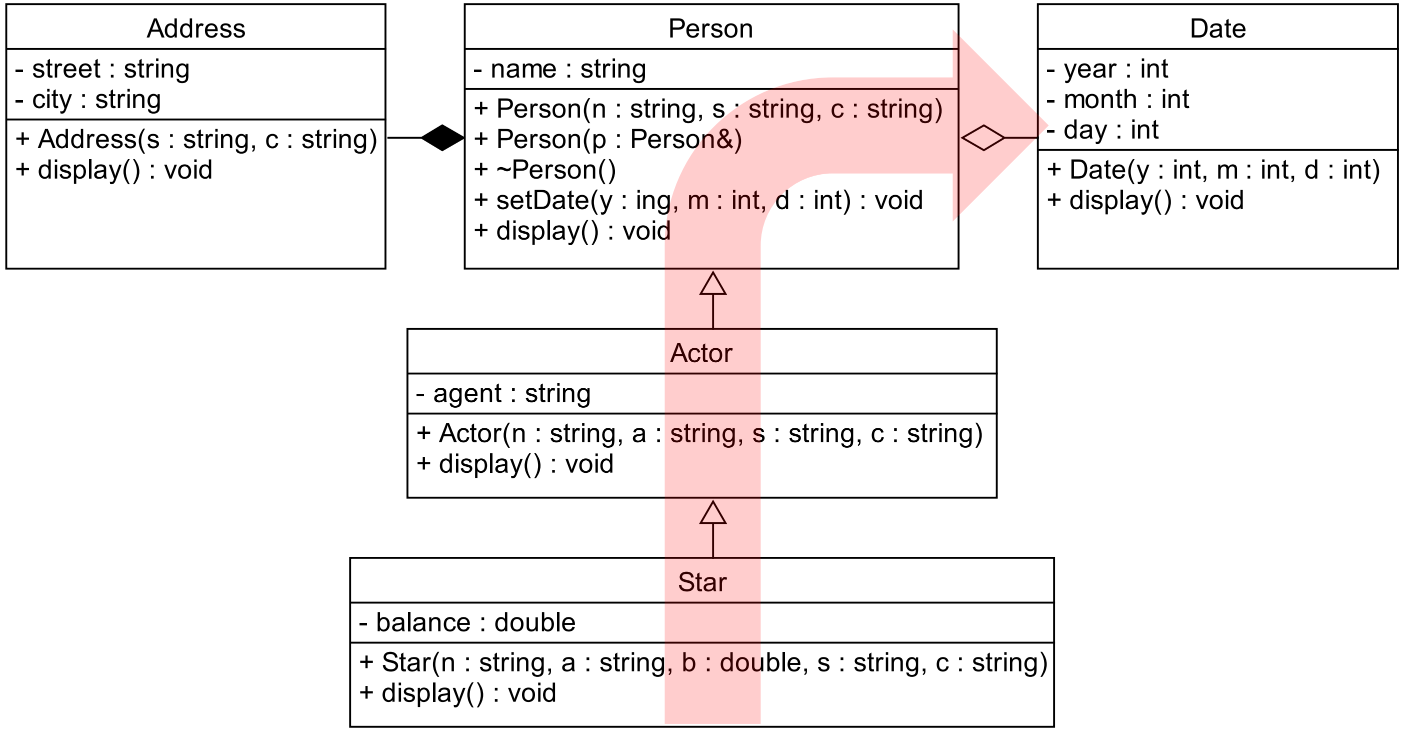
Person
- name : string
+ Person(n : string, s : string, c : string)
+ Person(p : Person&)
+ ~Person()
+ setDate(y : ing, m : int, d : int) : void
+ display() : void

Date
- year : int
- month : int
- day : int
+ Date(y : int, m : int, d : int)
+ display() : void

Actor
- agent : string
+ Actor(n : string, a : string, s : string, c : string)
+ display() : void

Star
- balance : double
+ Star(n : string, a : string, b : double, s : string, c : string)
+ display() : void

AGGREGATION ALTERNATIVE



Address

- street : string
- city : string

+ Address(s : string, c : string)
+ display() : void

Person

- name : string

+ Person(n : string, s : string, c : string)
+ Person(p : Person&)
+ ~Person()
+ setDate(y : ing, m : int, d : int) : void
+ display() : void

Date

- year : int
- month : int
- day : int

+ Date(y : int, m : int, d : int)
+ display() : void

Actor

- agent : string

+ Actor(n : string, a : string, s : string, c : string)
+ display() : void

Star

- balance : double

+ Star(n : string, a : string, b : double, s : string, c : string)
+ display() : void

Person.h (2)

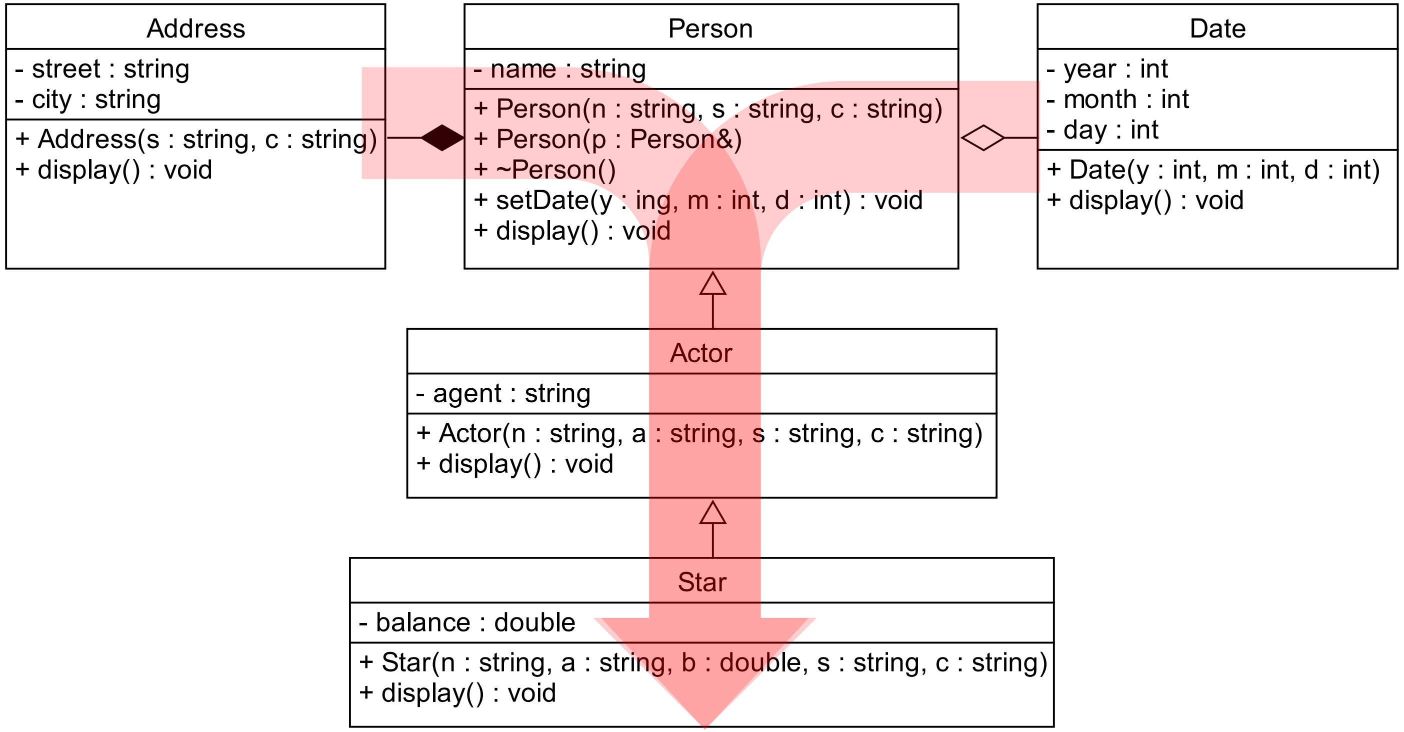
```
Person(string n, string s, string c)
        : addr(s, c), name(n) {}

Person(Person& p)
    : name(p.name),
      addr(p.addr),
      date(new Date(*p.date)) {}

    ~Person() { delete date; }
```

AGGREGATION SETTER FUNCTIONS

```
void Person::setDate(int y, int m, int d)
{
    delete date;
    date = new Date(y, m, d);
}
```

```
void Person::setDate(Date* d)
{
    delete date;
    date = d;
}
```

ACTOR 3 DISPLAY



Address

- street : string
- city : string

+ Address(s : string, c : string)
+ display() : void

Person

- name : string

+ Person(n : string, s : string, c : string)
+ Person(p : Person&)
+ ~Person()
+ setDate(y : ing, m : int, d : int) : void
+ display() : void

Date

- year : int
- month : int
- day : int

+ Date(y : int, m : int, d : int)
+ display() : void

Actor

- agent : string

+ Actor(n : string, a : string, s : string, c : string)
+ display() : void

Star

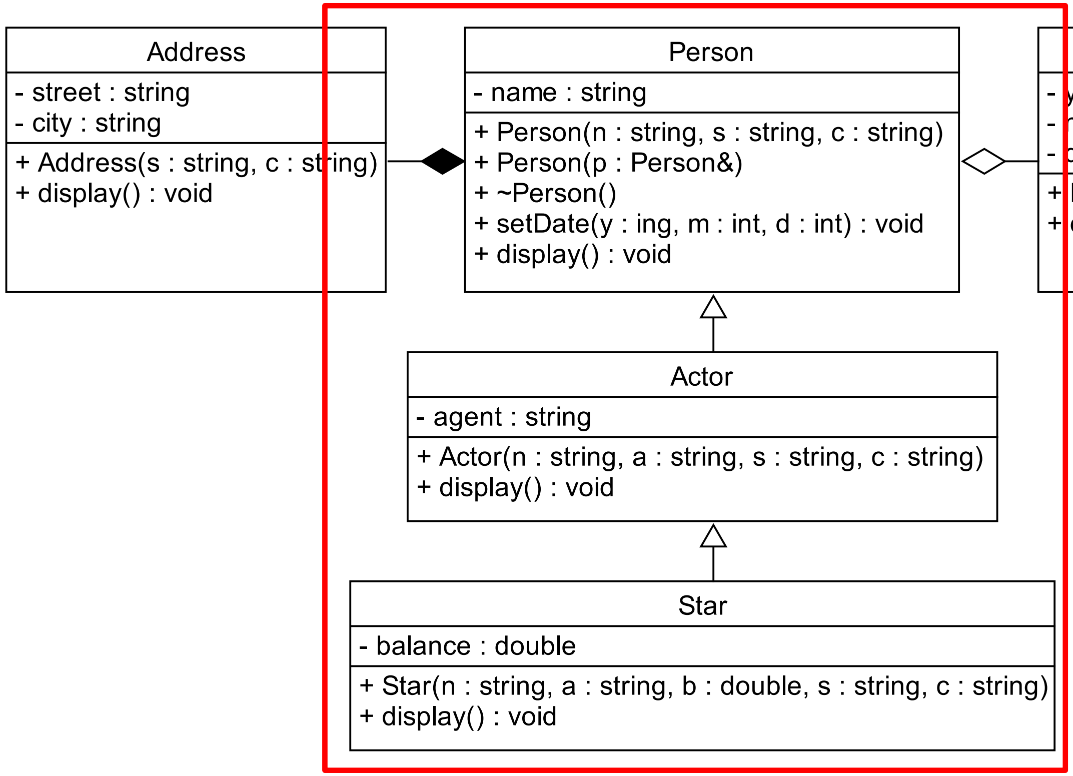
- balance : double

+ Star(n : string, a : string, b : double, s : string, c : string)
+ display() : void

person::display

```
void Person::display()
{
    cout << name << endl;
    addr.display();
    if (date != nullptr)
        date->display();
}
```

ACTOR 3 CORE



Address
- street : string
- city : string
+ Address(s : string, c : string)
+ display() : void

Person
- name : string
+ Person(n : string, s : string, c : string)
+ Person(p : Person&)
+ ~Person()
+ setDate(y : ing, m : int, d : int) : void
+ display() : void

Date
- year : int
- month : int
- day : int
+ Date(y : int, m : int, d : int)
+ display() : void

Actor
- agent : string
+ Actor(n : string, a : string, s : string, c : string)
+ display() : void

Star
- balance : double
+ Star(n : string, a : string, b : double, s : string, c : string)
+ display() : void

Actor AND Star CONSTRUCTORS

```
Actor(string n, string a, string s, string c) : Person(n, s, c), agent(a) {}


Star(string n, string a, double b, string s, string c) : Actor(n, a, s, c), balance(b) {}
```