

CS 485

Advanced Object Oriented Design

Visitor Pattern

Spring 2019

<http://www.netobjectives.com/PatternRepository/index.php?title=PatternsByAlphabet>

<http://www.netobjectives.com/files/books/dpe/design-patterns-matrix.pdf>

Visitor Pattern

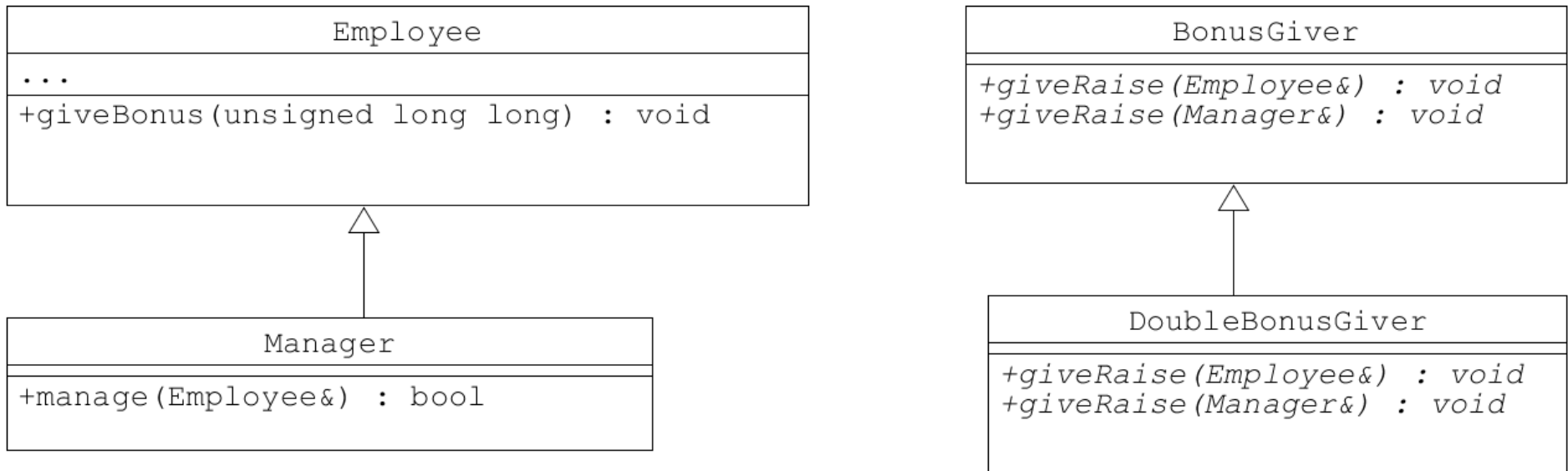
- Separate an operation from the structure that contains objects
 - customize operation for each type of object in the structure
- Easy to add new operations on the same structure

Represent an operation to be performed on elements of an object structure. **Visitor lets you define a new operation without changing the classes of the elements on which it operates.**

- Design Patterns, Gang of Four

Double Dispatch

- Decide which function to call, at runtime, based on the type of two different objects



Code - which function gets invoked?

```
BonusGiver cBonusGiver;  
DoubleBonusGiver cDb1BonusGiver;  
Employee cEmp;  
Manager cMang;
```

```
Employee &rcEmp = cMang;  
BonusGiver& rcBonusGiver = cDb1BonusGiver;
```

```
cBonusGiver.giveBonus(cEmp);  
cBonusGiver.giveBonus(cMang);
```

```
cDb1BonusGiver.giveBonus(cEmp);  
cDb1BonusGiver.giveBonus(cMang);
```

```
cBonusGiver.giveBonus(rcEmp);  
cDb1BonusGiver.giveBonus(rcEmp);
```

```
rcBonusGiver.giveBonus(rcEmp);
```

Code - double dispatch

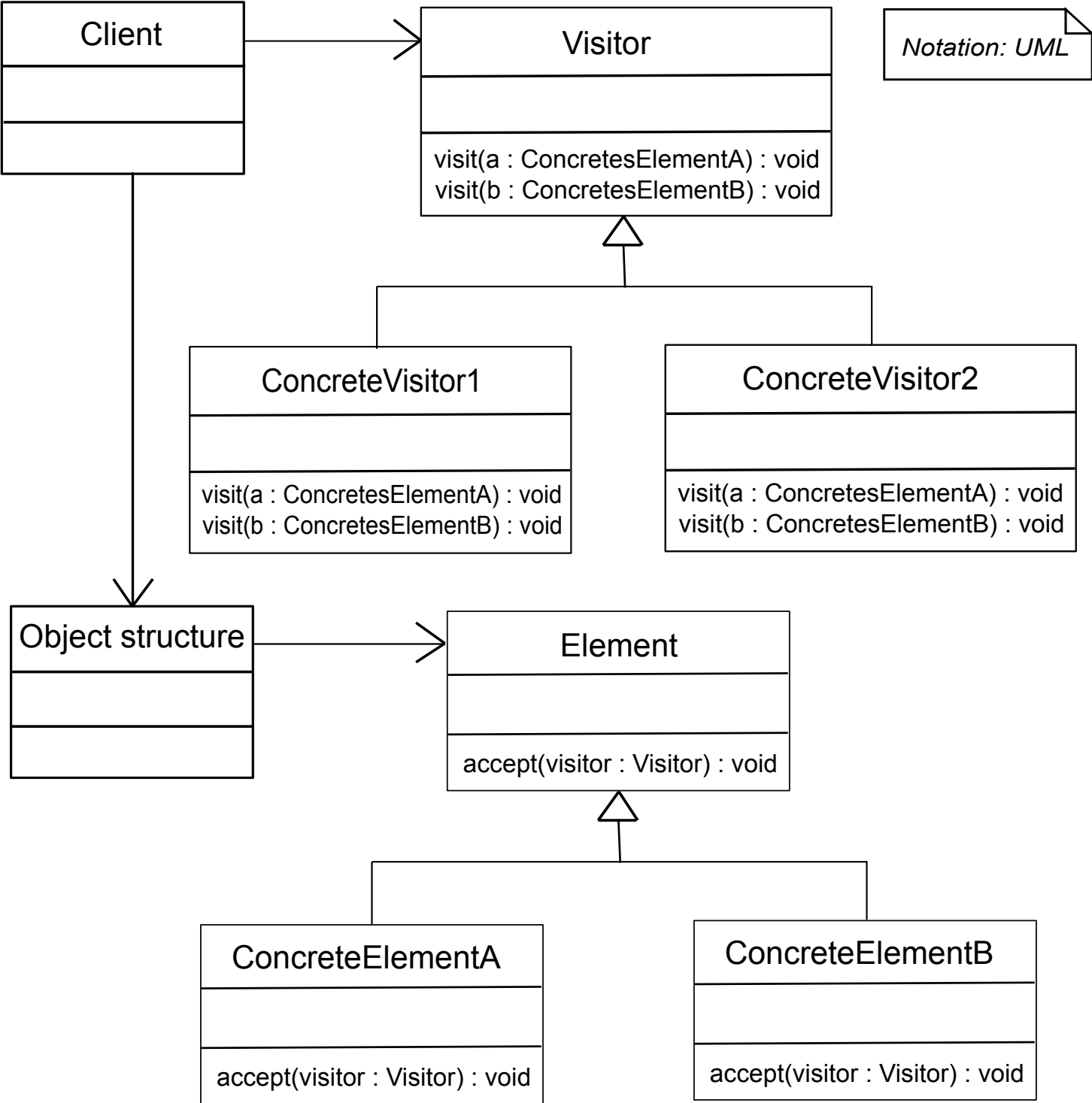
```
/*virtual*/ void Employee::getBonus(BonusGiver  
&rcBG)  
{  
    rcBG.giveBonus(*this);  
}
```

```
/*virtual*/ void Manager::getBonus(BonusGiver &rcBG)  
{  
    rcBG.giveBonus(*this);  
}
```

```
DoubleBonusGiver cDb1BonusGiver;  
BonusGiver& rcBonusGiver = cDb1BonusGiver;
```

```
Manager cMang;  
Employee &rcEmp = cMang;
```

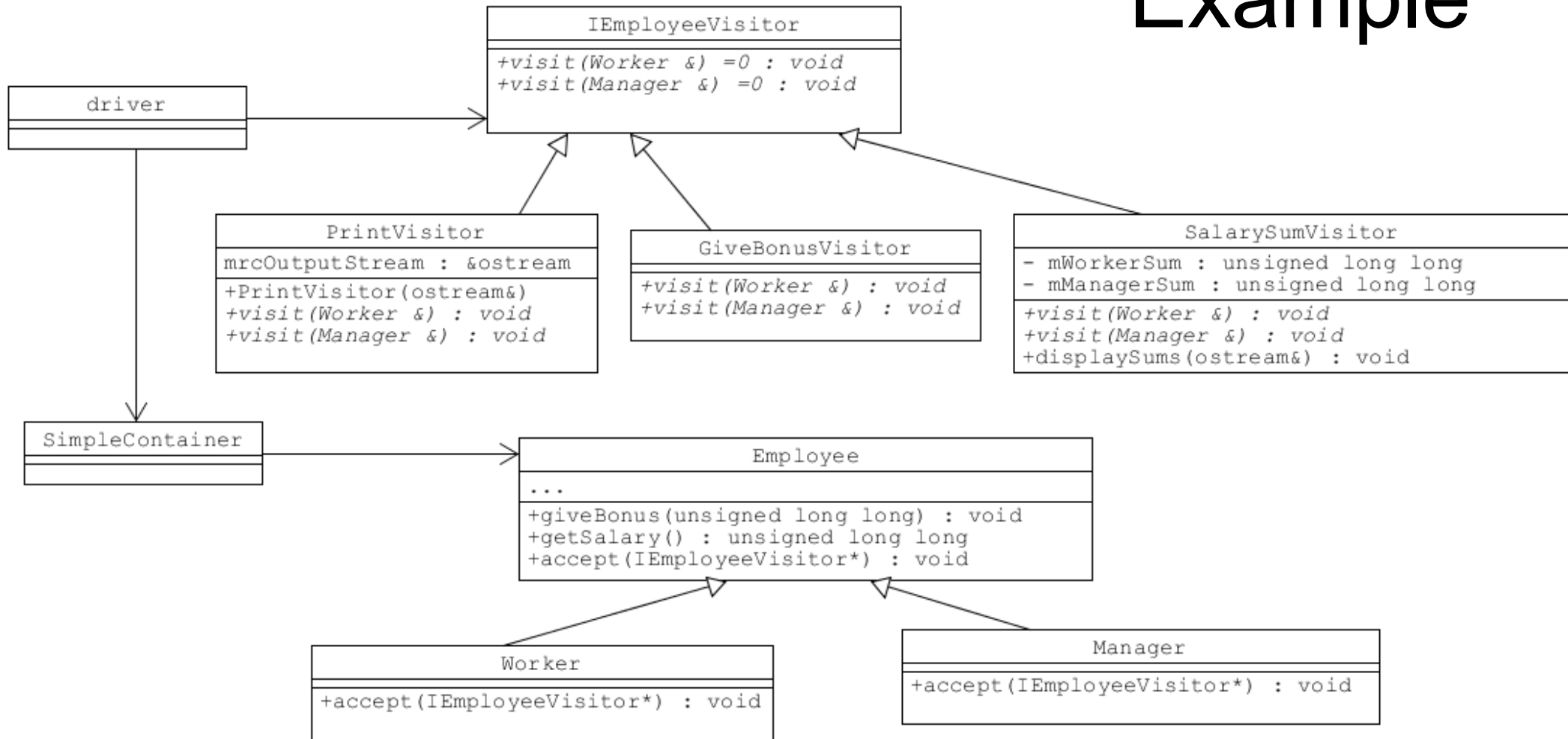
```
rcEmp.getBonus(cBonusGiver);
```



Double Dispatch

High coupling
best if the visited
elements are stable

Example



```
void Manager::accept (IEmployeeVisitor *pcVisitor)
{
    pcVisitor->visit (*this);
}
```

```
void GiveBonusVisitor::visit (Worker &rcWorker)
{
    rcWorker.giveBonus (50000);
}
```

Driver

```
EmpContainer::SimpleContainer cTheContainer;
PrintVisitor cPrintVisitor(std::cout);
GiveBonusVisitor cGiveBonusVisitor;
SalarySumVisitor cSalarySumVisitor;

cTheContainer.addEmployee (new Worker ("Bob", "Bib", 1000000));

EmpContainer::applyVisitor (cTheContainer, &cPrintVisitor);

EmpContainer::applyVisitor (cTheContainer, &cGiveBonusVisitor);

EmpContainer::applyVisitor (cTheContainer, &cPrintVisitor);

EmpContainer::applyVisitor (cTheContainer, &cSalarySumVisitor);
cSalarySumVisitor.displaySums (std::cout);
```


non-member, non-friend function

namespace!

```
void EmpContainer::applyVisitor (EmpContainer::SimpleContainer &rcCon,  
    IEmployeeVisitor *pcVisitor)  
{  
    for (std::size_t i = 0; i < rcCon.count (); ++i)  
    {  
        rcCon[i]->accept (pcVisitor);  
    }  
}
```