

Visitor: C++ Code Examples

22nd LECTURE

엄현상(Eom, Hyeonsang)
School of Computer Science and Engineering
Seoul National University

Outline

- **Visitor: C++ Code Examples**

Code Examples

- Visitor

```
#include <iostream>

using namespace std;

class Item;
class ItemA;
class ItemB;

class ItemVisitor;
class Pricing1Visitor;
class Pricing2Visitor;
```

“Design Patterns: Elements of Reusable
Object-Oriented Software,” Erich Gamma,
Richard Helm, Ralph Johnson, John
Vlissides, Addison Wesley, 1995

Code Examples Cont'd

- Visitor Cont'd

```
class Item {
public:
    virtual ~Item() {};

    void SetPrice(double p) { _price = p; };
    double GetPrice() { return _price; };

    virtual double NetPrice() { return _price; };
    virtual double DiscountPrice() { return 0.6*_price; };

    virtual void Accept(ItemVisitor&) = 0;
protected:
    Item() {};
private:
    double _price;
};
```

Code Examples Cont'd

- Visitor Cont'd

```
class ItemA : public Item {
public:
    ItemA() {};
    virtual void Accept(ItemVisitor& v) { v.VisitItemA(this); }
};

class ItemB : public Item {
public:
    ItemB() {};
    virtual void Accept(ItemVisitor& v) { v.VisitItemB(this); }
};
```

Code Examples Cont'd

- Visitor Cont'd

```
class ItemVisitor {  
public:  
    virtual void VisitItemA(Item*) {};  
    virtual void VisitItemB(Item*) {};  
protected:  
    ItemVisitor() {};  
};
```

“Design Patterns: Elements of Reusable
Object-Oriented Software,” Erich Gamma,
Richard Helm, Ralph Johnson, John
Vlissides, Addison Wesley, 1995

Code Examples Cont'd

- Visitor Cont'd

```
class Pricing1Visitor : public ItemVisitor {
public:
    Pricing1Visitor() { _total = 0.0; };

    double GetTotalPrice() { return _total; };

    virtual void VisitItemA(ItemA* pa) { _total += pa->NetPrice(); };
    virtual void VisitItemB(ItemB* pb) { _total += pb-
>DiscountPrice(); };

private:
    double _total;
};
```

Code Examples Cont'd

- Visitor Cont'd

```
class Pricing2Visitor : public ItemVisitor {
public:
    Pricing2Visitor() { _total = 0.0; };

    double GetTotalPrice() { return _total; };

    virtual void VisitItemA(ItemA* pa) { _total += pa-
>DiscountPrice(); };
    virtual void VisitItemB(ItemB* pb) { _total += pb->NetPrice(); };

private:
    double _total;
};
```


Code Examples Cont'd

- Visitor Cont'd

```
int main () {
    ItemA a;
    a.SetPrice(10);
    ItemB b;
    b.SetPrice(20);

    Pricing1Visitor p1;
    a.Accept(&p1); // p1.VisitItemA(&a);
    b.Accept(&p1); // p1.VisitItemB(&b);
    cout << "Pricing: p1.GetTotalPrice() = for a & b"
    << p1.GetTotalPrice() << endl;

    Pricing2Visitor p2;
    a.Accept(&p2); // p2.VisitItemA(&a);
    b.Accept(&p2); // p2.VisitItemB(&b);
    cout << "Pricing: p2.GetTotalPrice() = for b & b"
    << p2.GetTotalPrice() << endl;
}
```