

Design Pattern

Visitor

```
public interface IElement {
    int accept(IShoppingCartVisitor visitor);
}

public interface IShoppingCartVisitor {
    int visit(Book book);
    int visit(Fruit fruit);
}

public class Book implements IElement {
    private String name;
    private int price;
    private String isbnNumber;

    public Book(String name,int price,String isbnNumber) {
        this.name = name;
        this.price = price;
        this.isbnNumber = isbnNumber;
    }

    public String getISBNNumber() {
        return isbnNumber;
    }

    public void setISBNNumber(String isbnNumber) {
        this.isbnNumber = isbnNumber;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public int accept(IShoppingCartVisitor visitor) {
        return visitor.visit(this);
    }
}
```

```

public class Fruit implements IElement {
    private String name;
    private int price;

    public Fruit(String name,int price) {
        this.name = name;
        this.price = price;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public int accept(ISHoppingCartVisitor visitor) {
        return visitor.visit(this);
    }
}

public class ShoppingCart implements ISHoppingCartVisitor {
    public int visit(Book book) {
        int cost = 0;

        if (book.getPrice() > 50)
            cost = book.getPrice() - 5;
        else
            cost = book.getPrice();

        System.out.println("book ISBN   : " + book.getISBNNumber() +
                           " - cost = " + cost);

        return cost;
    }

    public int visit(Fruit fruit) {
        int cost;

        cost = fruit.getPrice();

        System.out.println("fruit name : " + fruit.getName() +
                           " - cost = " + cost);

        return cost;
    }
}

```

```

public class Application {
    public static void main(String... args) {
        IElement[] items = new IElement[]{new Book("book1",10,"isbn1"),
                                           new Book("book2",20,"isbn2"),
                                           new Fruit("fruit1",10)};

        IShoppingCartVisitor visitor = new ShoppingCart();

        int sum = 0;

        for (IElement item : items) {
            int variable = item.accept(visitor);
            sum+= variable;
        }

        System.out.println("sum : " + sum);
    }
}

```