

# Тема: Заместитель (Proху) или Сурогат (Surrogate)

Подготовили:  
Махиня Д.А.  
Кравченко В.С.



# Что представляет собой паттерн Proxy?

- Прокси - паттерн, контролирующий доступ к объектам, предоставляя более оптимальное их взаимодействие, перехватывая все вызовы (выполняет функцию контейнера).
- Прозрачно замещает объект и управляет доступом к нему. Не изменяет интерфейс или поведение.



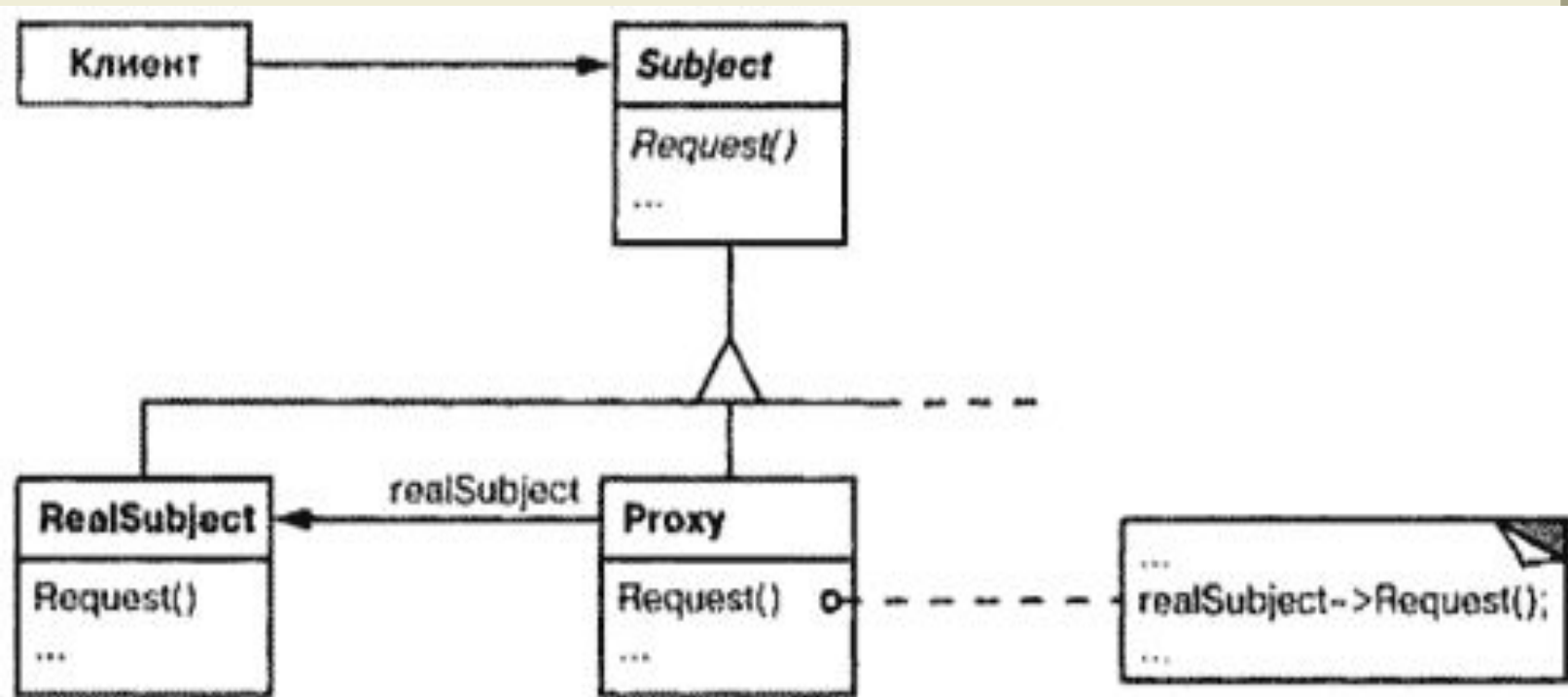
## Что представляет собой паттерн Proxy?

- Упрощает и оптимизирует работу с объектом. Может добавлять свою функциональность, скрывая ее от клиента.
- Содержит объект или ссылку на него, может управлять существованием замещенного объекта.



# Главная идея

Не переписывая, не вмешиваясь в код  
некой цельной системы обеспечить  
более оптимальное и экономичное  
взаимодействие участников этой  
системы.



# Что дает паттерн Proxy

1. Выявляются узкие места
2. Выявляются объекты, функционирование которых проходит не совсем оптимально
3. Вводятся самые объекты заместители, которые, дублируя интерфейс «проблемных» объектов, переадресовывают им вызовы клиента лишь тогда, когда это действительно необходимо, либо после некоторых оптимизационных действий.

В результате, не делая никаких изменений в архитектуре системы, мы получаем немалые возможности по улучшению качества функционирования.



## Когда выгодно использование данного паттерна

- работа с объектом не должна зависеть от того, где он реально расположен
- (или) нужно выполнять определенные действия при доступе к объекту;
- (или) необходимо оптимизировать взаимодействие объекта с клиентом.



## **Виды паттерна Proxy:**

- удаленный заместитель  
(Remote proxy)
- виртуальный заместитель  
(Virtual proxy)
- защищающий заместитель  
(Protection proxy)
- "умная" ссылка (Smart reference)



## **Удаленный заместитель**

Предоставление локального доступа к объекту в другом адресном пространстве или на удалённой машине.

Удаленный заместитель предоставляет локального представителя вместо целевого объекта, находящегося в другом адресном пространстве.





# Виртуальный заместитель

Обеспечивает «ленивую инициализацию»: создание реального субъекта только тогда, когда он действительно понадобится. Возможно при этом кешировать часть информации.

Реализация стратегии “копирование-при-записи”.



## **Защищающий заместитель**

Защищающий заместитель контролирует доступ к исходному объекту. Такие заместители полезны, когда для разных объектов определены различные права доступа.



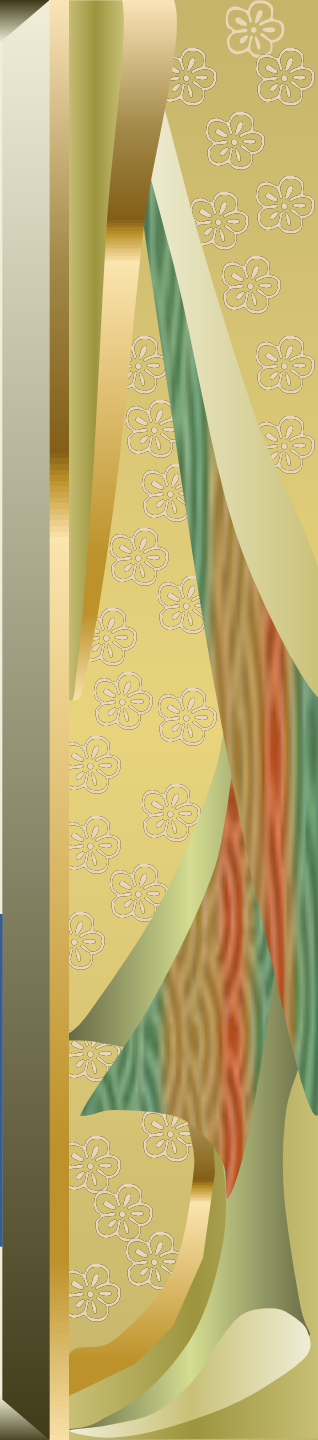
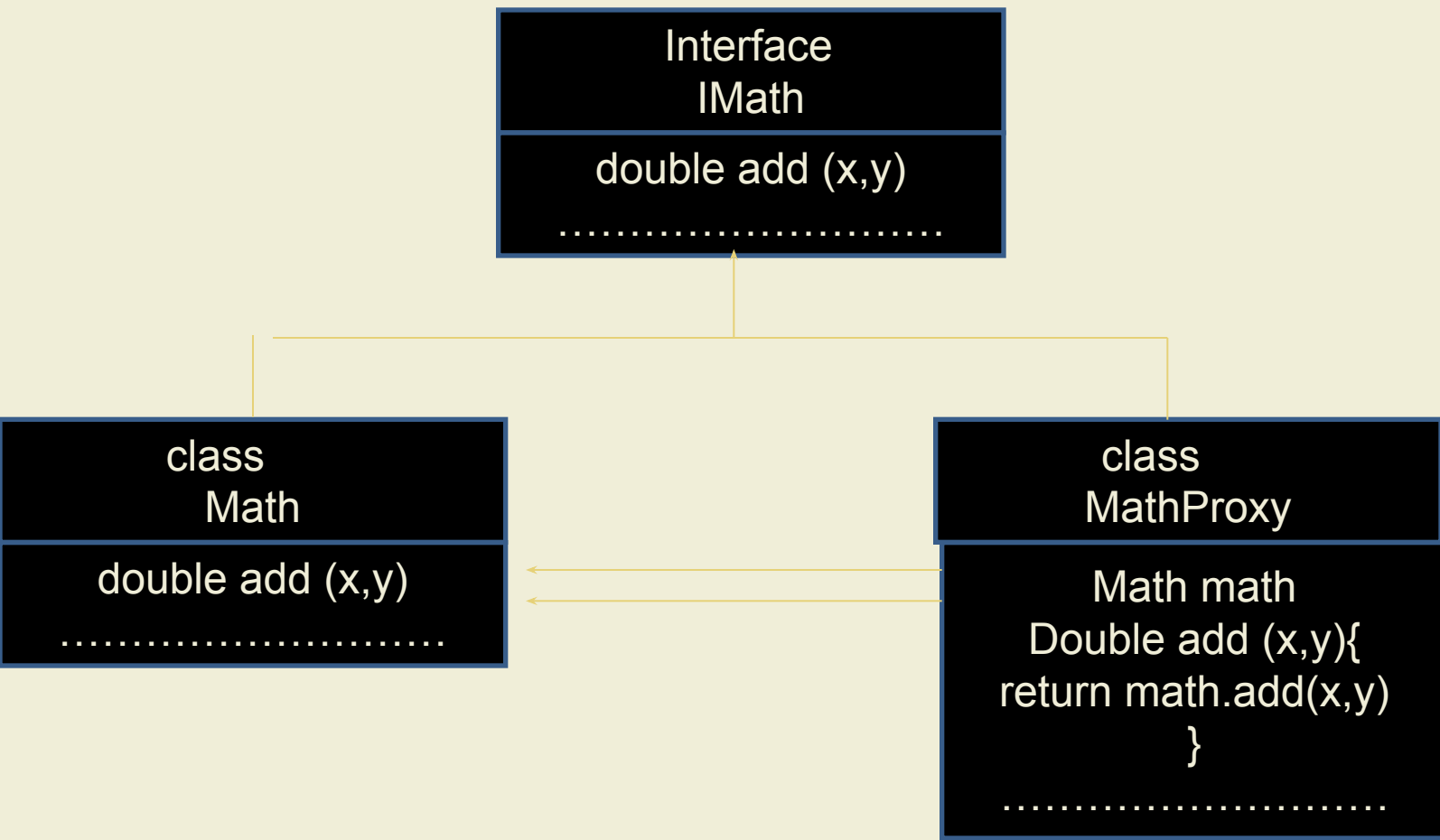
## «Умная ссылка»

Это замена обычного указателя. Она позволяет выполнить дополнительные действия при доступе к объекту.

Производит подсчет ссылок, автоматическое освобождение памяти и др.



# Пример использования Proxy



```
public interface IMath {
    public double add(double x, double y);
    public double sub(double x, double y);
    public double mul(double x, double y);
    public double div(double x, double y); }
/** * "Real Subject" */
public class Math implements IMath {
    public double add(double x, double y)
        { return x + y; }
    public double sub(double x, double y)
        { return x - y; }
    public double mul(double x, double y)
        { return x * y; }
    public double div(double x, double y)
        { return x / y; }}
/** * "Proxy Object" */
public class MathProxy implements IMath {
    private Math math;
    public MathProxy()
        { math = new Math(); }
    public double add(double x, double y)
        { return math.add(x, y); }
    public double sub(double x, double y)
        { return math.sub(x, y); }
    public double mul(double x, double y)
        { return math.mul(x, y); }
    public double div(double x, double y)
        { return math.div(x, y); }
}
```



```
public class Main {  
  
    public static void main(String[] args) {  
        // Create math proxy  
        IMath p = new MathProxy();  
  
        // Do the math  
        System.out.println("4 + 2 = " + p.add(4, 2));  
        System.out.println("4 - 2 = " + p.sub(4, 2));  
        System.out.println("4 * 2 = " + p.mul(4, 2));  
        System.out.println("4 / 2 = " + p.div(4, 2));  
    }  
}
```



# Преимущества и недостатки

## Преимущества:

- удаленный заместитель;
- виртуальный заместитель может выполнять оптимизацию;
- защищающий заместитель;
- "умная" ссылка;

## Недостатки

- резкое увеличение времени отклика.



## **Вопросы**

- **Что представляет собой паттерн Proxy?**
  - а) паттерн, который предоставляет необходимые функции, но не поддерживает нужного интерфейса
  - б) разделитель объектов на абстракцию и реализацию так, чтобы они могли изменяться независимо друг от друга
  - в) паттерн, контролирующий доступ к объектам, перехватывая все вызовы.
- **Назовите виды прокси.**





## Вопросы

- **Для чего нужен виртуальный заместитель?**
  - а) Обеспечивает создание реального субъекта только тогда, когда он действительно понадобится
  - б) Контролирует доступ к исходному объекту
  - в) Является заменой обычного указателя

