

Паттерн «Bridge».

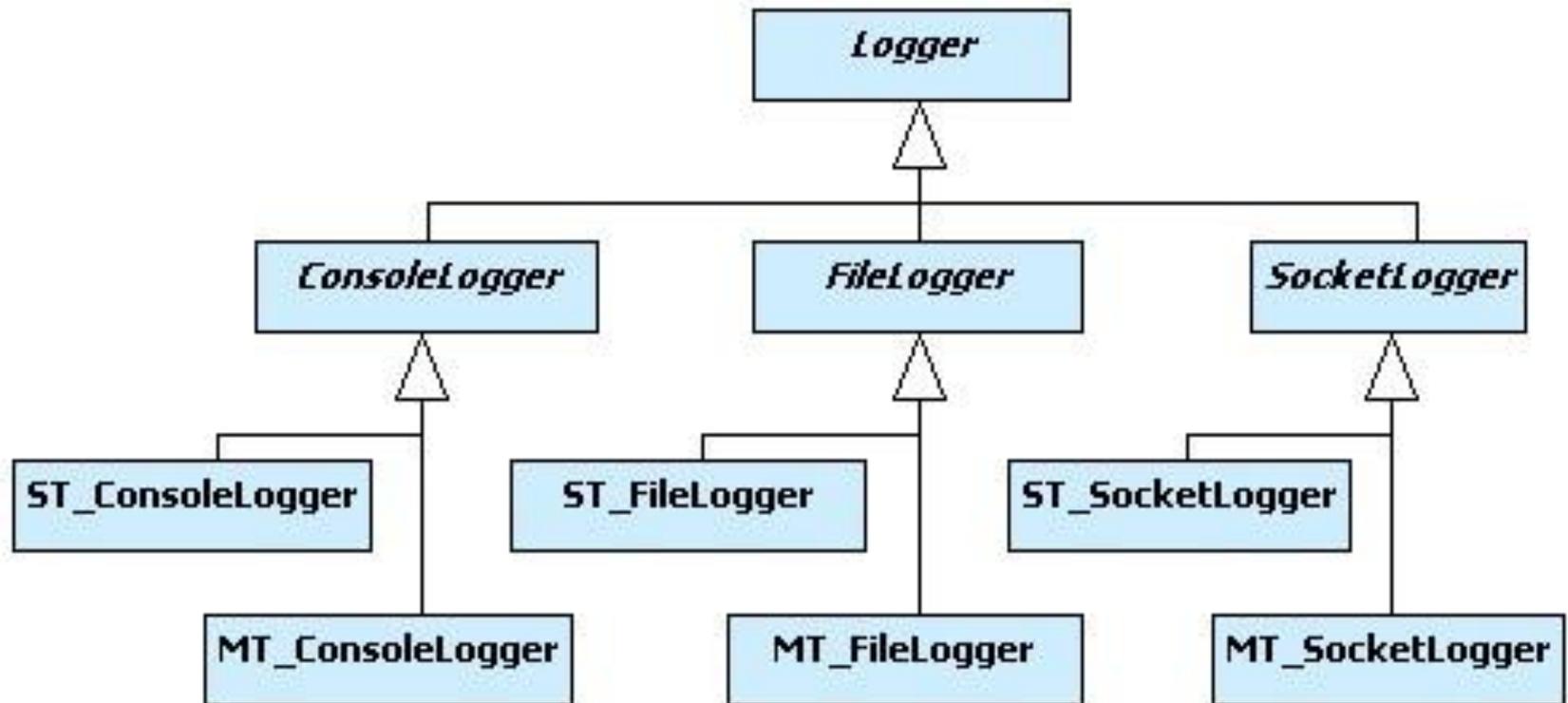
Назначение паттерна Bridge

Если в системе существуют классы, с отношением: абстрактный базовый класс объявляет интерфейс, а конкретные подклассы реализуют его нужным образом, возможны следующие недостатки:

- Система, построенная на основе наследования, является статичной. Реализация жестко привязана к интерфейсу. Изменить реализацию объекта некоторого типа в процессе выполнения программы уже невозможно
- Система становится трудно поддерживаемой, если число родственных производных классов становится большим

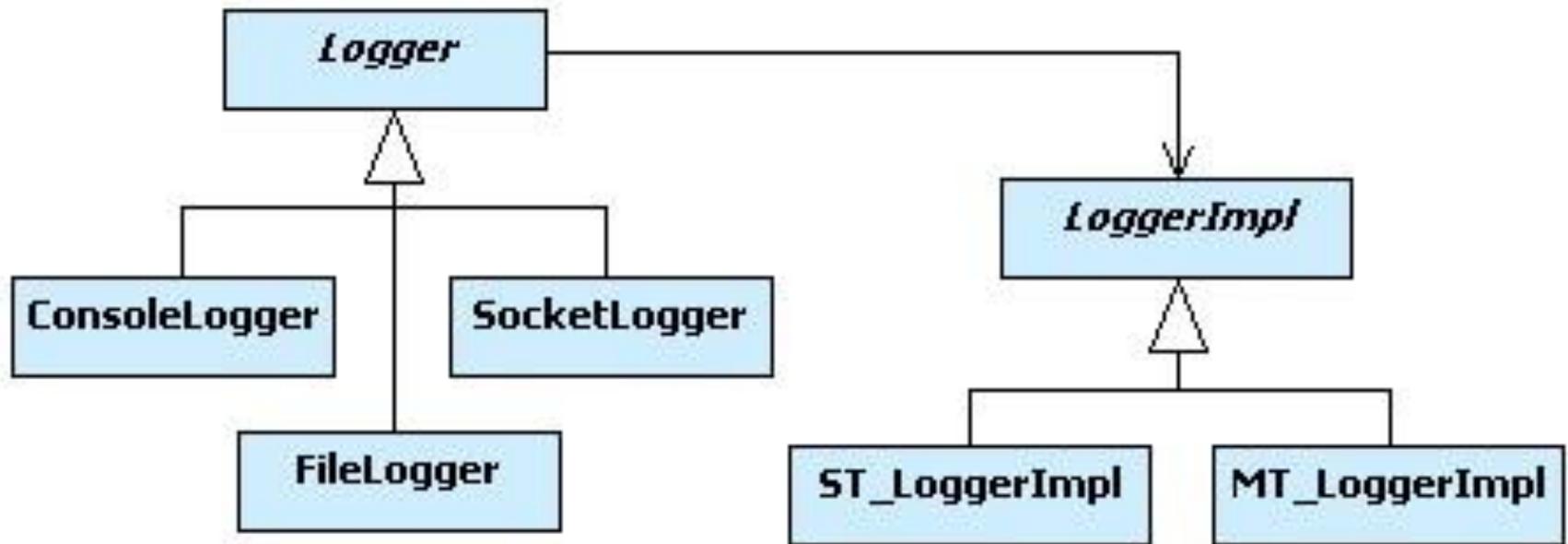
Пример (1)

Стандартный подход на основе полиморфизма



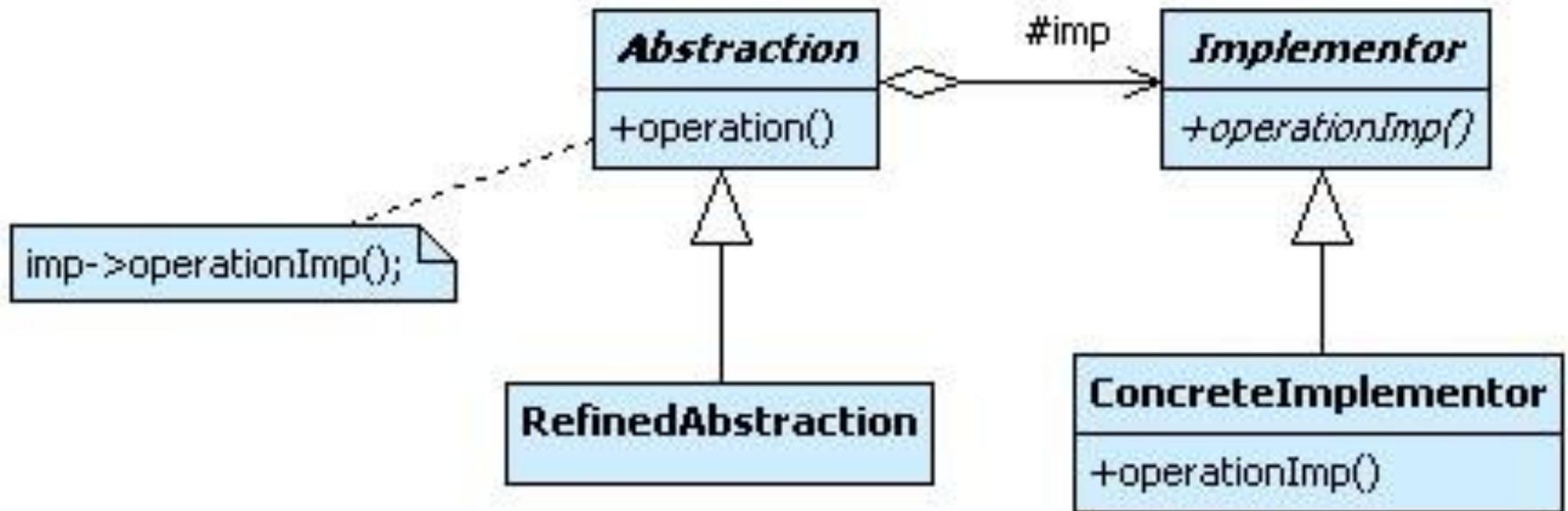
Пример (2)

Описание паттерна Bridge



Структура паттерна Bridge

(UML-диаграмма)



Достоинства паттерна Bridge

- Проще расширять систему новыми типами за счет сокращения общего числа родственных подклассов
- Возможность динамического изменения реализации в процессе выполнения программы
- Паттерн Bridge полностью скрывает реализацию от клиента. В случае модификации реализации пользовательский код не требует перекомпиляции