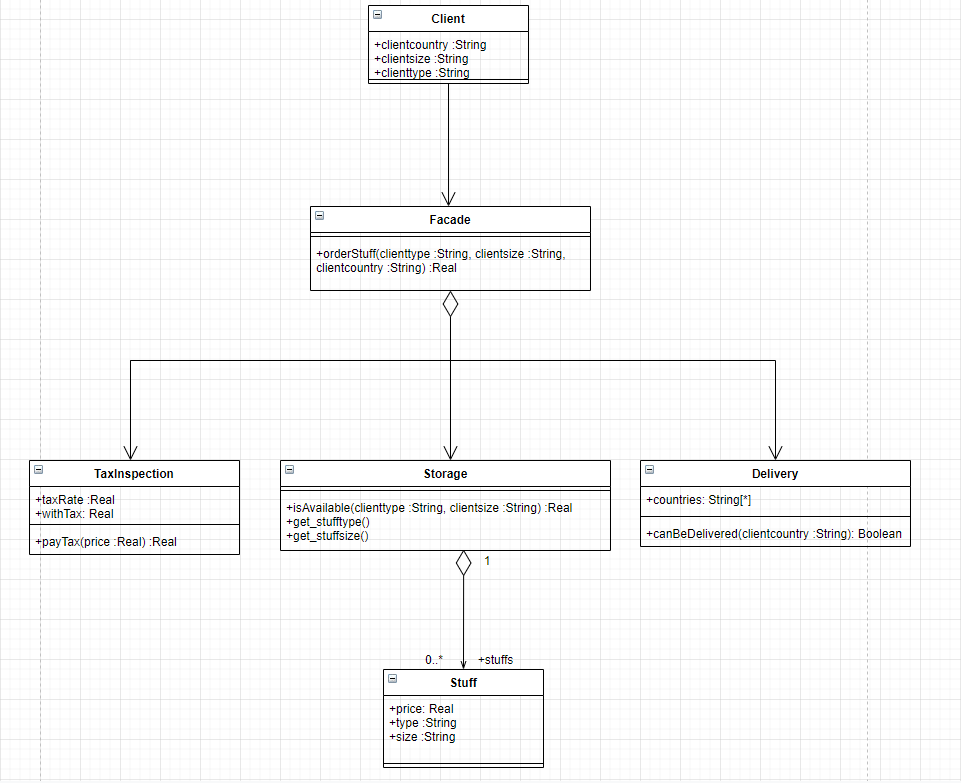
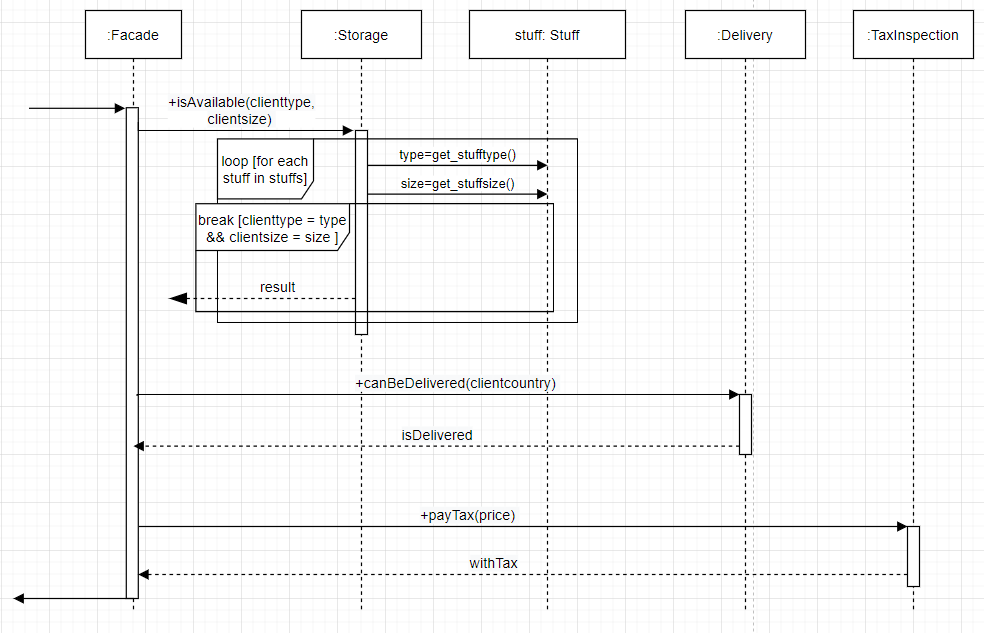
**Документация к паттерну «Фасад»**

**Диаграмма классов**



**Диаграмма последовательностей**



**Задумка**

Несмотря на то, что было сказано попытаться изложить основную суть на диаграммах, рискну написать какова все же была основная задумка. =)

Решил реализовать что-то похожее на интернет-магазин (с оооочень сильно урезанным функционалом). Совершая какие-либо покупки онлайн, со стороны клиента все выглядит довольно просто, но на самом деле по ту сторону происходит множество различных процессов, которые спрятаны от клиента. Клиент вносит следующую информацию:

* Необходимый тип вещи ("jeans", "shirt", "trousers", "jacket", "coat", "scarf")
* Размер – S/M/L
* Страна (доставка осуществляется в ("Russia", "Germany", "USA", "UK", "France")

Происходит ряд проверок:

* Проверка на наличие соответствующего товара выбранного размера (тут реализована «заглушка» - предполагаем, что на складе имеется всего 5 вещей (рандомные вещи рандомного размера), ведь, по сути, все это хранится в базе данных/1C) – если товара нет, возвращаем -1
* Проверка на возможность доставки в выбранную покупателем страну, если нет, то возвращаем 0
* Начисление НДС на товар дороже 5000

**Почему применил паттерн**

В моем случае применение паттерна «защищает» клиента от необходимости обращения сначала ко складу, затем к службе доставки, а затем еще и к налоговой. Это все делает фасад.

P.S. если я как-то криво скинул работу в мудле, добавляю ссылку на работу на Github: https://github.com/ParadiZzzze/Reshetnikov.931802.Individual-project