Лабораторная работа 2. Основы работы в Visual Paradigm CE.

Выполнил: Михалев Никита Романович М3311

## Цели и задачи:

В этой лабораторной работе необходимо изучить интерфейс Visual Paradigm Community Edition и с её помощью перерисовать диаграммы нескольких типов, сохраняя максимально возможное количество деталей.

Все диаграммы нужно сделать в одном проекте: открыв один файл, практик должен увидеть список из пяти диаграмм.

Все диаграммы ниже взяты с сайта <https://www.uml-diagrams.org/>.

## Диаграмма использования (Use Case Diagram)

Диаграмма, которую необходимо повторить:

Изображение выглядит как диаграмма, линия, текст

Автоматически созданное описание

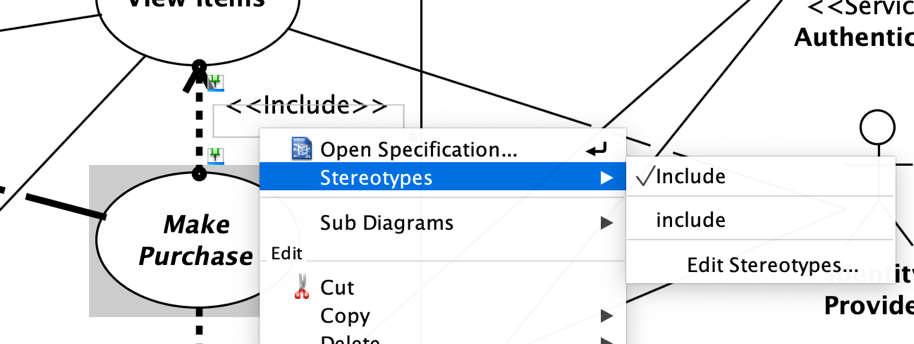
Изучив диаграмму, сделал вывод, что необходимо будет использовать стереотипы Subsystem, include и Service.

Использование стереотипа Subsystem:

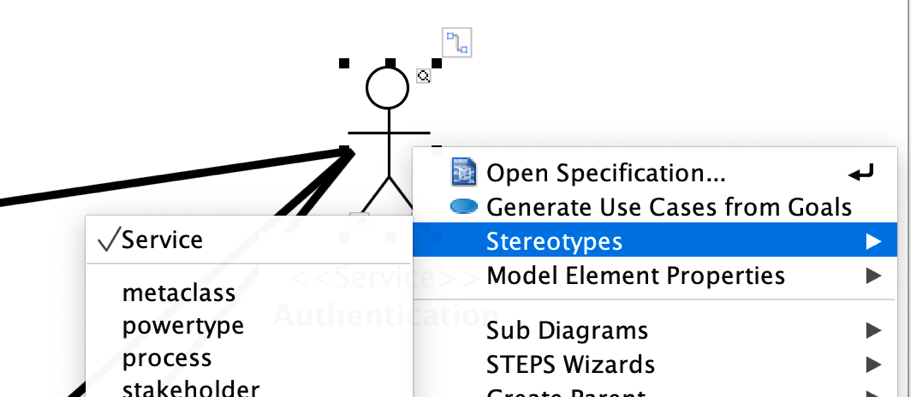
Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Использование стереотипа include:



Использование стереотипа Service:



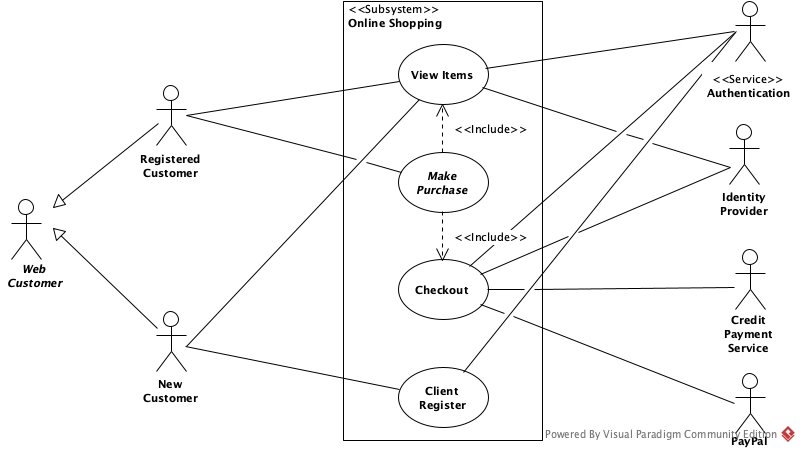
Также в диаграмме используется курсивный шрифт, обозначающий, что сущность является абстрактной, следовательно на диаграмме надо указать что соответствующие сущности будут абстрактными:

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Получившаяся диаграмма:



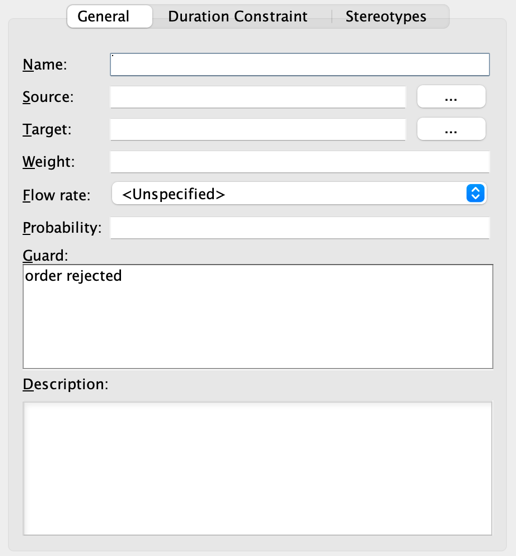
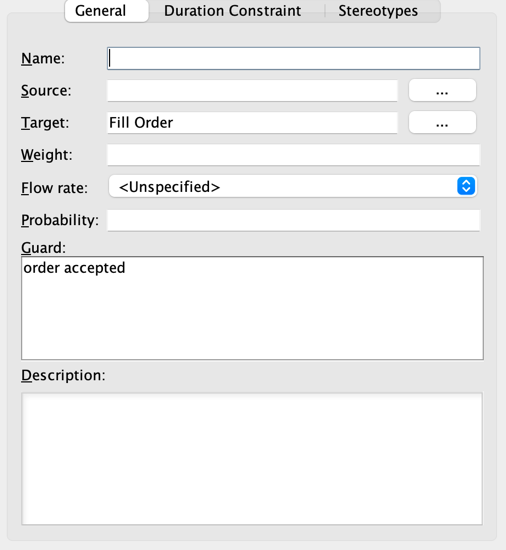
## Диаграмма деятельности (Activity Diagram)

Диаграмма, которую необходимо повторить:

Изображение выглядит как диаграмма, линия, План, Технический чертеж

Автоматически созданное описание

Изучив диаграмму, сделал вывод, что для выбора цепочки действий необходимо использовать guard на Control Flow:



Получившаяся диаграмма:

Изображение выглядит как диаграмма, линия, снимок экрана, текст

Автоматически созданное описание

## Диаграмма классов (Class Diagram)

Диаграмма, которую необходимо повторить:

Изображение выглядит как текст, диаграмма, План, Технический чертеж

Автоматически созданное описание

Изучив диаграмму, сделал вывод, что для описания отношений между классами необходимо будет использовать association end from/to:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, программное обеспечение, число

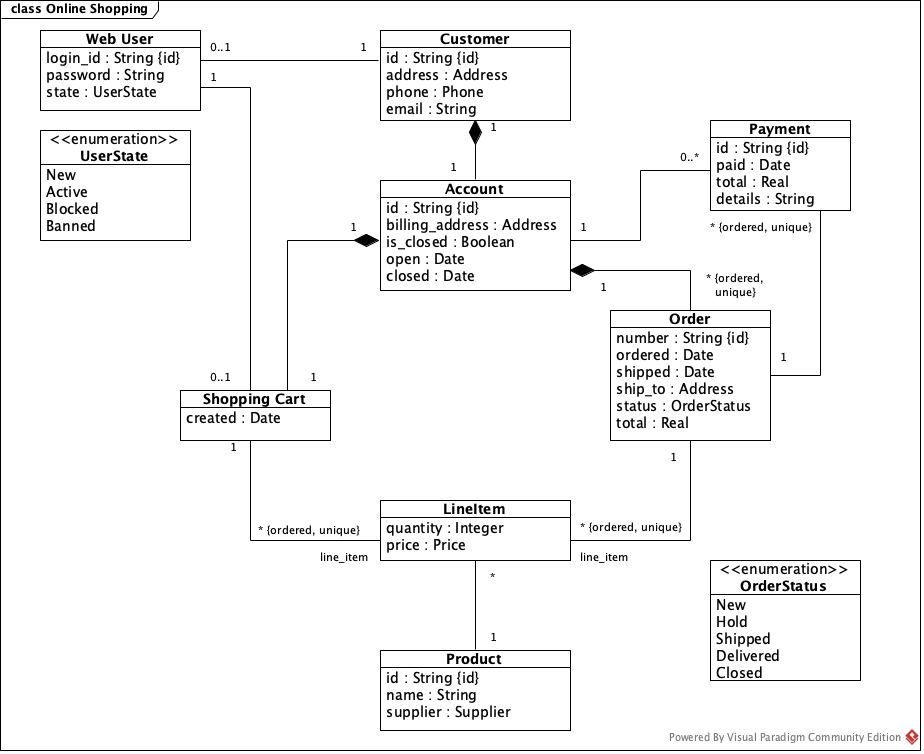
Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Также для классов перечислений необходимо использовать стереотип enumeration:Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Получившаяся диаграмма:



## Диаграмма последовательности (Sequence Diagram)

Диаграмма, которую необходимо повторить:

Изображение выглядит как текст, диаграмма, Параллельный, План

Автоматически созданное описание

Изучив диаграмму, сделал вывод, что необходимо будет использовать message различных типов: Call, Create, Destroy и Reply:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

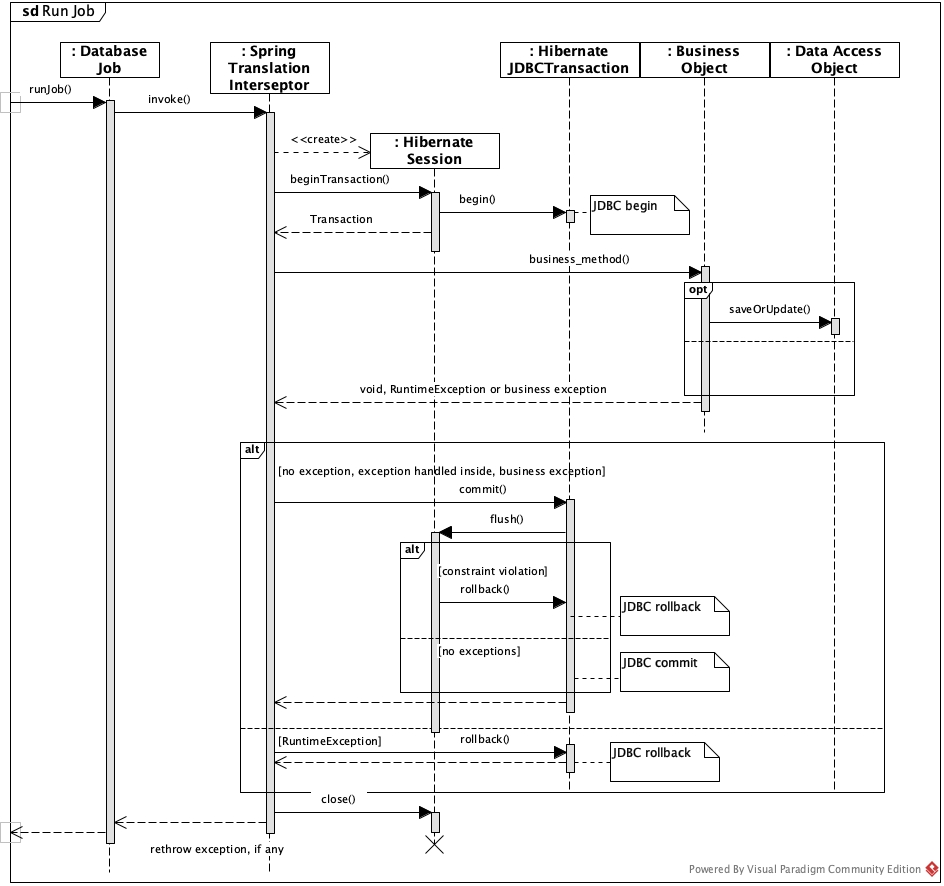
Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Получившаяся диаграмма:



## Диаграмма размещения (Deployment Diagram)

Диаграмма, которую необходимо повторить:

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Артефакт web-tools-lib.jar и схемы СУБД можно показать, как обычные артефакты, в случае схем нужно добавить соответствующий стереотип. Однако при желании можно найти иконки и сделать диаграмму один в один.

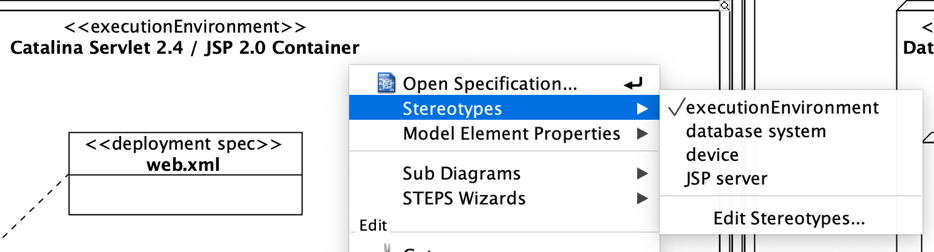
Изучив диаграмму, сделал вывод, что для Application Server необходимо использовать три node, со стереотипами device, JSP server и executionEnvironment:

Изображение выглядит как текст, Шрифт, линия, число

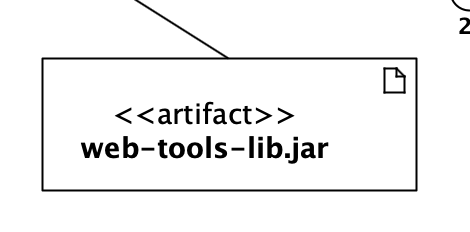
Автоматически созданное описание

Изображение выглядит как текст, линия, снимок экрана, Шрифт

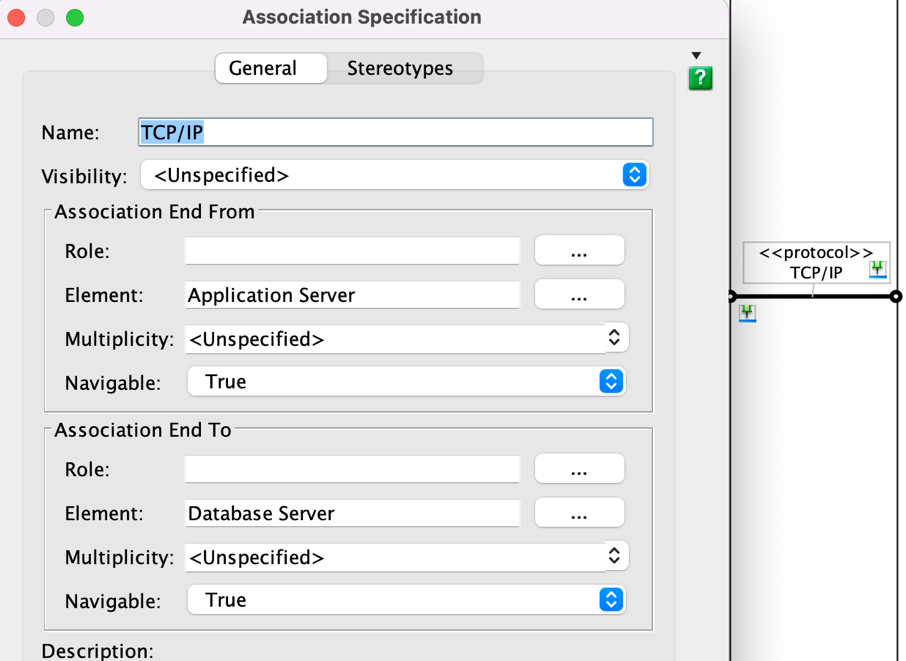
Автоматически созданное описание



Вместо web-tool-lib.jar использовал обычный artifact:



Для связи между нодами использовал association specification, со стереотипом protocol:



Получившаяся диаграмма:

