

# Методические указания по работе с программным обеспечением StarUML<sup>TM</sup>

21 июня 2021 г.

# Оглавление

<b>1</b>	<b>Краткий обзор StarUML™</b>	<b>2</b>
<b>2</b>	<b>Основные концепции</b>	<b>4</b>
<b>3</b>	<b>Управление проектом</b>	<b>7</b>
<b>4</b>	<b>Моделирование с помощью StarUML™</b>	<b>11</b>
<b>5</b>	<b>Справочник по интерфейсу пользователя</b>	<b>17</b>
<b>6</b>	<b>Построение диаграмм UML с помощью ПО StarUML™</b>	<b>30</b>
6.1	Особенности разработки диаграмм вариантов использования . . . . .	30
6.2	Особенности разработки диаграмм классов . . . . .	38
6.3	Особенности разработки диаграмм кооперации . . . . .	46
6.4	Особенности разработки диаграммы последовательности . . . . .	50
6.5	Особенности разработки диаграммы состояний . . . . .	57
6.6	Особенности разработки диаграммы деятельности . . . . .	62
6.7	Особенности разработки диаграммы компонентов . . . . .	70
6.8	Особенности разработки диаграммы развертывания . . . . .	75
	<b>СПИСОК ИЛЛЮСТРАТИВНОГО МАТЕРИАЛА . . . . .</b>	<b>81</b>
	<b>СПИСОК ТАБЛИЦ . . . . .</b>	<b>83</b>

# Глава 1

## Краткий обзор StarUML™

StarUML™ — программный инструмент моделирования, который поддерживает UML (Унифицированный язык моделирования). StarUML™ ориентирован на UML версии 1.4 и поддерживает одиннадцать различных типов диаграмм, принятых в нотации UML 2.0. Он активно поддерживает подход MDA (Модельно-управляемая архитектура), реализуя концепцию профилей UML. Среда разработки StarUML™ превосходно настраивается в соответствии с требованиями пользователя и имеет высокую степень расширяемости, особенно в области своих функциональных возможностей. Использование StarUML™, одного из ведущих программных инструментов моделирования, гарантирует достижение максимальной производительности и качества ваших программных проектов.

### Инструмент UML, который адаптируется к пользователю

StarUML™ предоставляет максимальную степень адаптации среды разработки пользователя, предлагая настройку параметров, которые могут влиять на методологию разработки программного обеспечения, проектную платформу и язык.

### Превосходная расширяемость и гибкость

StarUML™ обеспечивает превосходную расширяемость и гибкость. Он предоставляет механизм аддинов, чтобы расширять свои функциональные возможности. Этот механизм разработан специально, чтобы предоставлять доступ ко всем функциям модели/мета-модели посредством COM Automation и расширять меню и набор свойств элементов. Также, пользователи могут создавать собственные подходы и механизмы согласно своим собственным методологиям. Программа может также быть интегрирована с любыми внешними инструментальными средствами.

### Системные требования

Ниже указаны минимальные системные требования, необходимые для функционирования StarUML™.

- Intel ® Pentium ® 233MHz или выше
- Windows ® 2000, Windows XP ТМ, или выше
- Microsoft ® Internet Explorer 5.0 или выше
- 128 Мбайт RAM (256 МБ рекомендуется)
- 110 Мбайт на жестком диске (150 МБ рекомендуется)
- Устройство CD-ROM
- SVGA или монитор с более высокой разрешающей способностью (1024x768 рекомендуется)
- Мышь или другое устройство позиционирования

# Глава 2

## Основные концепции

Эта глава вводит фундаментальные концепции, которые требуется знать для эффективного использования StarUML™. Она содержит описание моделей, визуальных элементов и диаграмм, проектов, секций, подходов, фреймворков, модельных фрагментов, их различий относительно разных профилей UML.

- Модель,
- Представление (view) и Диаграмма
- Проект и проектная секция (unit)
- Модуль (module)

### Модель, Представление и Диаграмма

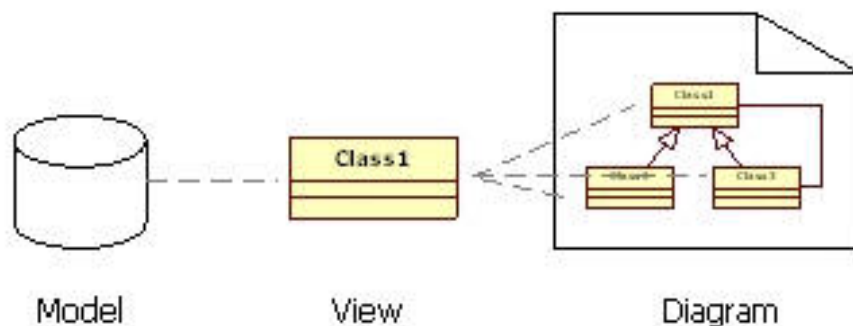


Рис. 2.1: Модель, представление, диаграмма

StarUML™ предполагает ясное понимание концептуального различия между моделями, представлениями и диаграммами. Модель — элемент, который содержит всю информацию о модели программы. Представление — визуальное выражение информации, содержащейся в модели, а Диаграмма — коллекция визуальных образов, которая отображает определенные аспекты проекта.

# **Проект и проектная секция**

## **Проект**

Проект – основная структурная единица в StarUML™. Проект может содержать одну или более программных моделей. Проект — корневой пакет верхнего уровня, который всегда существует в любой программной модели. В общем случае, один проект сохраняется в одном файле.

## **Структура проекта**

Проект содержит следующие суб-элементы:

Модель — элемент, который соответствует одной программной модели.

Подсистема — элемент, который соответствует модели подсистемы.

Пакет — самый общий элемент для группировки других элементов.

## **Проектный файл**

Проектные файлы сохраняются в формате XML и имеют расширение «.UML». Все модели, представления и диаграммы, созданные в StarUML™ сохраняются в одном проектном файле. Проект может также быть разделен и сохранен в нескольких проектных секциях. Проектный файл содержит следующую информацию:

- профиль UML, используемый в проекте
- файлы секций, на которые ссылается проект
- информация по всем моделям, содержащимся в проекте
- информация по всем диаграммам и представлениям, содержащимся в проекте

## **Секции**

Хотя проект обычно сохраняется в одном файле, бывают случаи, когда его целесообразно хранить в нескольких небольших файлах так, чтобы несколько разработчиков могли работать над проектом одновременно. В этом случае, проект представляется в виде набора секций. Секция может иметь иерархическую структуру; она может содержать несколько подсекций. Секции сохраняются как XML-файлы на которые ссылаются проектные файлы (.UML) или другие файлы секций (.UNT).

## **Состав секции**

Только пакет, подсистема или модель могут составлять секцию. Все элементы внутри пакетов этих типов сохраняются в соответствующем файле секции (.UNT).

## **Иерархическая структура секции**

Также, как проект может содержать много секций внутри себя, секция тоже может включать много подсекций. Так как родительская секция имеет ссылки на свои дочерние секции, всё множество секций имеет иерархическую структуру.

## **Фрагменты модели**

Фрагмент модели — часть проекта, сохраненная как отдельный файл. Только модель, подсистема или пакет может являться фрагментом модели. Файлы модельных фрагментов сохраняются с расширением «.MFG». Они могут быть легко включены в любой проект в любое время. Фрагменты модели существенно отличаются от секций, которые полностью едины с остальной частью проекта.

# Глава 3

## Управление проектом

Эта глава подробно описывает операции по управлению проектом: создание нового проекта, размещение части проекта в секции, создание и импорт фрагментов модели, импорт фреймворков, подключение и исключение профилей UML.

- Управление проектом
- Управление секциями
- Работа с фрагментами модели
- Импорт фреймворка
- Работа с профилями UML

### Управление проектом

#### Создание нового проекта

Чтобы начать разработку программного обеспечения, нужно инициировать новый проект. Вы можете начать абсолютно пустой проект или инициализировать новый проект согласно определённому подходу.

#### Процедура создания нового проекта №1 — New Project

1. выберите меню [File] → [New Project].
2. Новый проект будет создан в соответствии с подходом по умолчанию, ранее выбранным пользователем. В зависимости от подхода могут быть подгружены определённые профили и/или инструментарии.

#### Процедура создания нового проекта №2 — Select Select New Project

1. Выберите меню [File] → [Select New Project...]. В диалоговом окне New Project будет отображен список доступных подходов.
2. Выберите нужный из списка и нажмите кнопку [OK].



3. Новый проект будет создан и инициализирован согласно указанному подходу. В зависимости от подхода могут быть подгружены определённые профили и/или фреймворки.

### **Примечание**

- Список доступных подходов зависит от установленной среды разработки пользователя.
- Чтобы изменить заданный по умолчанию подход, откройте диалоговое окно «Select New Project», выберите нужный подход и затем установите опцию «Set As Default Approach» (Использовать как подход по умолчанию).

## **Открытие проекта**

Чтобы начать работать с существующим проектом, его проектный файл нужно открыть. Если проект включает более одной секции, все связанные секции будут также загружены вместе с проектом.

### **Процедура открытия проекта:**

1. Выберите меню [File] → [Open...].
2. В диалоговом окне «Open Project», выберите файл проекта (.UML) и нажмите кнопку [Open].
3. Выбранный проектный файл будет открыт.

**Примечание** Проекты могут также быть открыты через диалоговое окно «Select New Project».

## **Сохранение проекта**

Чтобы изменения, сделанные в проекте, не пропали, проектный файл должен быть должным образом сохранен. Ваша работа может быть сохранена в существующий проектный файл или в новый проектный файл. Когда проектный файл сохраняется, то вместе с ним сохраняются данные из связанных с ним секций.

### **Процедура сохранения проекта:**

1. Выберите меню [File] → [Save].
2. Если имя файла проекта не было определено, появится диалоговое окно «Save Project». Введите имя файла, и нажмите кнопку [Save].
3. Проектный файл будет сохранен.

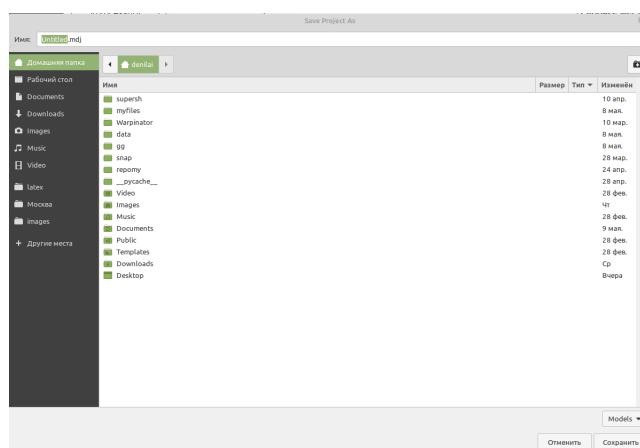


Рис. 3.1: Сохранение проекта

### Процедура сохранения проекта в новом файле:

1. Выберите меню [File] → [Save As...].
2. В диалоговом окне «Save As», введите новое имя файла, и нажмите кнопку [Save].
3. Проект будет сохранен в указанном файле.

### Примечание

- Если проект содержит одну или более секций, и есть изменённые секции, диалоговое окно будет запрашивать подтверждение на сохранение каждой измененной секции. Выберите [Yes], чтобы сохранить измененную секцию вместе с проектом.

### Заккрытие проекта

Проект может быть закрыт, если больше не требуется его редактирование.

### Процедура закрытия проекта:

1. Выберите меню [Файл] → [Close].
2. Если проект не был сохранен после внесения изменений, пользователю будет предложено сохранить изменения. Пользователь может выбрать «да», «нет» или «отмена».
3. После закрытия проект становится недоступным для редактирования.

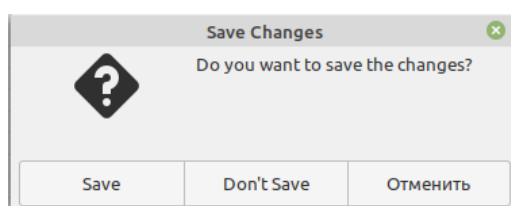


Рис. 3.2: Заккрытие проекта

## Управление элементами с помощью моделей, подсистем и пакетов

Программная модель состоит из многих элементов и диаграмм. Правильная группировка этих элементов и диаграмм очень важна для эффективного управления проектом. StarUML™ поддерживает три типа группирующих элементов (модели, подсистемы и пакеты), которые пользователь может использовать соответственно их назначению. Способы группировки элементов, реализованные в StarUML™.



Модель

Модель выражает физическую систему в определенном аспекте. Например, это может быть аспект анализа, аспект проекта, пользовательский аспект, и т.д.



Подсистема

Подсистема группирует элементы, которые составляют полную физическую систему или её части.



Пакет

Пакет логически группирует и содержит модельные элементы. Это чрезвычайно обобщенный элемент, который может использоваться только для того, чтобы как-то организовать модельные элементы.

## Глава 4

# Моделирование с помощью StarUML™

Эта глава подробно описывает, как создавать и редактировать элементы диаграммы, включая способы организации структуры модели с помощью навигатора модели.

- Редактирование элементов и диаграмм
- Организация структуры модели

### Редактирование элементов и диаграмм

#### Создание новой диаграммы

StarUML™™ поддерживает 11 типов диаграмм UML. Пользователь может свободно создавать и манипулировать диаграммами различных типов, как ему необходимо.

#### Процедура создания новой диаграммы:

1. Выберите в навигаторе модели или на диаграмме элемент, который будет содержать новую диаграмму.
2. Щелкните правой кнопкой мыши и выберите [Add Diagram]. Новая диаграмма будет создана после выбора типа диаграммы.

#### Доступные типы диаграмм

- **Диаграмма классов (Class diagram)**

Диаграмма классов — визуальное отображение различных статических отношений между класс-подобными элементами. Диаграмма классов может содержать не только классы, но также и интерфейсы, перечислимые типы, пакеты, различные отношения, инстанции и их связи.

- **Диаграмма прецедентов (Use case diagram)**

Диаграмма прецедентов — отображение отношений между вариантами использования (прецедентами) определенной системы или объекта и внешними акторами. Вариант ис-

пользования отображает функции системы и то, как эти функции взаимодействуют с внешними акторами.

- **Диаграмма сообщений (Sequence Diagram)**

Диаграмма сообщений отображает взаимодействие экземпляров. Она является прямым отображением множества взаимных воздействий (InteractionInstanceSet) между элементами множества экземпляров (CollaborationInstanceSet). В то время как Диаграмма сообщений роли ориентирована на классификаторы-роли, обычная Диаграмма сообщений — на экземпляры.

- **Диаграмма сообщений роли (Sequence Role Diagram)**

Диаграмма сообщений роли отображает взаимодействия в концепции ролей. Она является прямым отображением Interaction (множества взаимных сообщений между классификаторами-ролями) в пределах Collaboration. В то время как Диаграмма сообщений — отображение экземпляров, Диаграмма сообщений роли — отображение классификаторов-ролей.

- **Диаграмма коллаборации (Collaboration Diagram)**

Диаграмма коллаборации отображает взаимодействие между экземплярами. Она является прямым отображением модели взаимодействия экземпляров, входящих в CollaborationInstanceSet. В то время как диаграмма коллаборации ролей — отображение классификаторов-ролей, обычная диаграмма коллаборации — отображение экземпляров.

- **Диаграмма коллаборации ролей**

Диаграмма коллаборации ролей отображает взаимодействия между ролями. Она является прямым отображением модели взаимодействия классификаторов-ролей внутри коллаборации. В то время как обычная диаграмма коллаборации ориентирована на отображение экземпляров, диаграмма коллаборации ролей — отображение классификаторов-ролей.

- **Диаграмма состояний (Statechart Diagram)**

Диаграмма состояний выражает статическое поведение определенного объекта через состояния и переходы состояний. Хотя диаграмма состояний обычно используется, чтобы выразить поведение экземпляров классов, она может также использоваться, чтобы выражать поведение и других элементов.

- **Диаграмма действий (Activity Diagram)**

Диаграмма действий — специальная форма диаграммы состояний, которая является подходящей для того, чтобы отображать поток выполнения действий. Диаграмма действий в общем случае используется для отображения любых потоков обработки, но чаще всего применительно к объектам подобным классам, пакетам и операциям.

- **Диаграмма компонентов (Component Diagram)**

Диаграмма компонентов отображает зависимость между программными компонентами. Элементы, которые составляют программные компоненты и элементы, которые реализуют эти компоненты, могут быть отображены на диаграмме компонентов.

- **Диаграмма развертывания (Deployment Diagram)**

Диаграмма развертывания отображает аппаратные элементы компьютера, другие устрой-

ства и программные компоненты, а также процессы и объекты, которые им назначены.

- **Композиционная структурная диаграмма (Composite Structure Diagram)**

Композиционная структурная диаграмма — диаграмма, выражающая внутреннюю структуру классификатора. Она показывает его точки зрения взаимодействия с другими частями системы.

### **Примечание**

- Типы доступных диаграмм изменяются при переходе от одного типа элемента к другому.

## **Создание элемента на диаграмме**

Чтобы создать на диаграмме новый элемент, диаграмму сначала нужно открыть. Палитра элементов содержит различные типы элементов, доступных для создания в зависимости от типа диаграммы. Список доступных элементов изменяется при переходе от диаграммы одного типа к диаграмме другого типа.

### **Процедура создания элемента из палитры элементов:**

1. Выберите тип создаваемого элемента на палитре элементов.
2. Щёлкните желаемое место для нового элемента на диаграмме, чтобы создать там элемент. (Перетаскивайте указатель мыши, чтобы определить область и размер нового элемента. При создании элемента, который соединяет два других элемента, убедитесь, что соединение сделано правильно.)

### **Процедура одновременного создания нескольких однотипных элементов:**

1. Выберите тип создаваемого элемента на палитре элементов.
2. Нажмите [Lock] на палитре или тот же тип элемента еще раз.
3. Создайте несколько элементов подряд.
4. Снова нажмите элемент в палитре, когда создание группы элементов будет закончено.

### **Примечание**

- Создание элемента на диаграмме с помощью палитры элементов фактически означает создание как собственно модельного элемента, так и его визуального образа.

## **Редактирование элемента на диаграмме**

Элементы могут редактироваться непосредственно на диаграмме.

### **Процедура редактирования элемента:**

1. Дважды щелкните образ на диаграмме.
2. В «горячем диалоге» редактируйте имя элемента, область видимости и т.д., или нажмите кнопку, чтобы создать подчинённые элементы для выбранного элемента.
3. Нажмите [Enter] или щёлкните другое место на диаграмме, чтобы принять изменения.

### **Примечание**

- Для детального описания работы с горячим диалогом, см. Горячие диалоги.

### **Изменение размеров и перемещение**

Вы можете оптимизировать размер визуального образа и его позицию внутри диаграммы, а также Вы можете плавно менять его позицию и устанавливать размеры комбинацией клавиш <Специальная>+<Клавиша курсора>.

#### **Процедура изменения размера образа:**

1. Щёлкните образ на диаграмме.
2. Изменяйте размер образа перетаскиванием в нужном направлении одного из маркеров, которые показываются после выделения образа.

#### **Процедура изменения размеров образа, используя клавиатуру:**

1. Выделите образ, щёлкнув его на диаграмме.
2. Для изменения размеров образа пользователь может применить комбинацию *Shift + Cursor key*. Эта комбинация обеспечивает изменение шагами установленного размера, но Вы можете изменять размер образа плавно комбинацией *Shift + Alt + Cursor*.

#### **Процедура перемещения образа:**

1. Выделите передвигаемый образ щёлкнув его на диаграмме. Если нужно выделить несколько образов, выделяйте их комбинацией *Shift + Click* или очертите мышью область на диаграмме.
2. Переместите образы в нужное место перетаскиванием, используя мышь.

#### **Процедура перемещения образа, используя клавиатуру:**

1. Выделите передвигаемый образ щёлкнув его на диаграмме. Если нужно выделить несколько образов, выделяйте их комбинацией *Shift + Click* или очертите мышью область на диаграмме.
2. Переместите образы куда нужно, используя *Ctrl + Cursor Key*. Эта комбинация обеспечивает передвижение шагами установленного размера, но Вы можете передвигать образ плавно комбинацией *Ctrl + Alt + Cursor Key*.

## **Создание элемента с использованием «горячего синтаксиса»**

Элементы также могут быть созданы без использования мыши, использованием горячего синтаксиса.

### **Процедура создания элемента с помощью горячего синтаксиса:**

1. Выберите образ на диаграмме.
2. Вызовите горячий диалог, нажав [Enter].
3. Введите нужный синтаксис в горячем диалоге.

### **Горячий синтаксис генерации**

Горячий синтаксис позволяет генерировать элементы модели и отношения через написание несложного текста. Основное правило горячего синтаксиса состоит в следующем. Напишите имя целевого модельного элемента и установленный нотационный знак, чтобы создать отношение соответствующее этому знаку между текущим и целевым элементом. Если целевое имя не указано, будет сгенерирован новый модельный элемент и указанное отношение. Нотация знаков отношений горячего синтаксиса приведена ниже:

### **Копирование и вставка**

При копировании в буфер модельных элементов для последующей вставки необходимо ясно понимать различие между модельными элементами и их визуальными образами. Если скопирован модельный элемент, он должен быть вставлен в модельный элемент. В этом случае, все подэлементы, содержащиеся в выбранном элементе, копируются вместе с ним. Визуальные образы могут копироваться в пределах той же самой диаграммы или в другие диаграммы. Скопированный в буфер визуальный образ может быть вставлен только в диаграмму; он не может быть вставлен в модельный элемент. Возможность копирования и вставки может быть ограничена в зависимости от типа визуального образа и типа диаграммы.

### **Процедура копирования и вставки модельных элементов:**

1. Выберите модельный элемент для копирования в навигаторе модели.
2. Щелкните правой кнопкой мыши и выберите меню [Copy]. Модельный элемент будет скопирован в буфер обмена.
3. Выберите в навигаторе модели модельный элемент, в который будет вставлен скопированный элемент.
4. Щелкните правой кнопкой мыши и выберите меню [Paste]. Скопированный модельный элемент будет вставлен из буфера обмена под выбранный элемент. Скопированные модельные элементы могут быть вставлены только в те элементы, которые могут их содержать.



### **Процедура копирования и вставки визуального образа в диаграмме:**

1. Выберите на диаграмме визуальный образ для копирования. (Вы можете выбрать несколько элементов, очертив мышью прямоугольную область, или щёлкая мышью образы, удерживая нажатой клавишу [Shift])
2. Щелкните правой кнопкой мыши и выберите меню [Copy]. Визуальные образы будут скопированы в буфер обмена.
3. Откройте диаграмму, в которую нужно вставить скопированные элементы. (Дважды щелкните визуальный образ в навигаторе модели или проводнике диаграмм, или выберите визуальный образ на вкладке диаграммы) .
4. Щелкните правой кнопкой мыши и выберите меню [Paste]. Скопированные визуальные образы будут вставлены в активную диаграмму.

### **Копирование/вставка для различных типов диаграмм**

- **Диаграмма классов**  
Элементы могут свободно копироваться или вставляться между диаграммами классов, прецедентов, компонентов, композиционной структуры и развертывания.
- **Диаграмма прецедентов**  
Элементы могут свободно копироваться или вставляться между диаграммами классов, прецедентов, компонентов, композиционной структуры и развертывания.
- **Диаграммы следования**  
Элементы не могут быть скопированы или вставлены
- **Диаграммы коллаборации**  
Элементы не могут быть скопированы или вставлены
- **Диаграмма состояний**  
Элементы могут быть скопированы или вставлены только между диаграммами в пределах той же самой модели состояний
- **Диаграмма действий**  
Элементы могут быть скопированы или вставлены только между диаграммами в пределах той же самой модели активности
- **Диаграмма компонентов**  
Элементы могут свободно копироваться или вставляться между диаграммами классов, прецедентов, компонентов, композиционной структуры и развертывания.
- **Диаграмма развертывания**  
Элементы могут свободно копироваться или вставляться между диаграммами классов, прецедентов, компонентов, композиционной структуры и развертывания.
- **Структурная диаграмма**  
Элементы могут свободно копироваться или вставляться между диаграммами классов, прецедентов, компонентов, композиционной структуры и развертывания.

## Глава 5

# Справочник по интерфейсу пользователя

Эта секция подробно описывает все окна StarUML™ :

- Главное окно
- Меню
- Инструментальные панели
- Окно
- Диалоговое окно
- Горячий диалог

### **Главное окно**

Главное окно StarUML™ состоит из следующих компонентов.

#### **Главное меню**

Главное меню — находится наверху экрана. Большинство функции StarUML™ доступны через главное меню.

#### **Инструментальные панели**

Инструментальные панели находятся ниже главного меню. Их кнопки дублируют часто используемые пункты меню.

#### **Область браузера**

Область браузера расположена в верхнем правом углу экрана. Эта область содержит инструменты, облегчающие просмотр составляющих элементов проекта. Эта область включает [Навигатор модели], который показывает модельные элементы в виде иерархической структуры, и [Навигатор диаграмм], который показывает диаграммы модели, сгруппированные по типам.

## Область инспектора

Область инспектора расположена в нижнем правом углу экрана. Эта область содержит инструменты, облегчающие редактирование детальной информации о модельных элементах. Эта область включает [Редактор свойств], который позволяет редактировать свойства, [Документационный редактор], который позволяет водить детальные описания элементов, и [Редактор вложений], который позволяет присоединять к элементам дополнительные файлы.

## Информационная область

Информационная область расположена в нижней части экрана. Эта область содержит инструменты, показывающие различные вспомогательные данные, касающиеся приложения StarUML™. Эта область включает [Окно Вывода], которое показывает регистрационную информацию, и [Окно Сообщений], которое показывает результаты поиска и проверки модели.

## Область диаграммы

Область диаграммы расположена в центре экрана. Эта область содержит инструментарий, позволяющий редактировать и управлять диаграммами.

## Палитра элементов

Расположена на левой стороне экрана. Палитра содержит инструменты для быстрого создания модельных элементов.

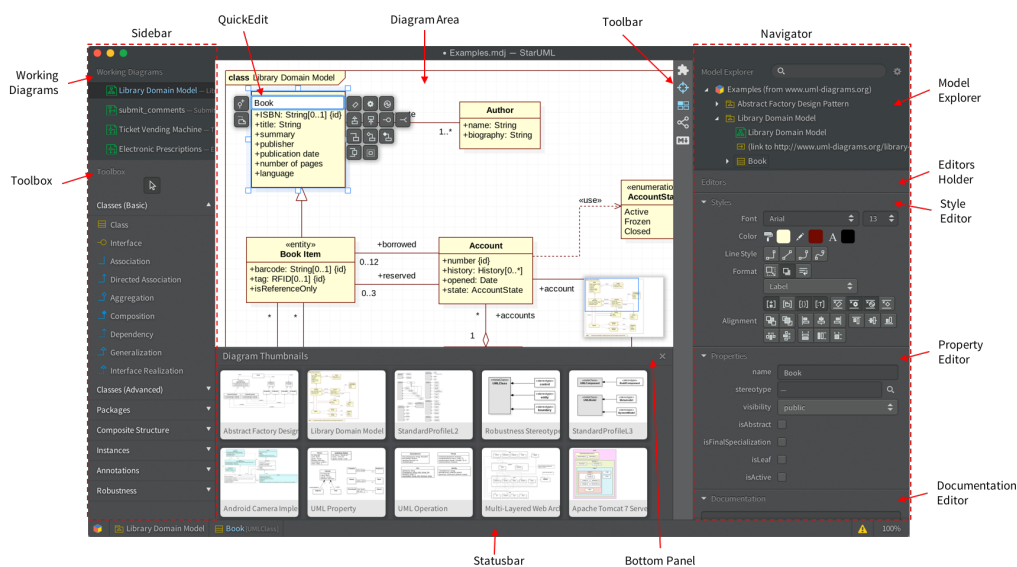


Рис. 5.1: Главное окно

## Меню

Эта секция описывает подробно все пункты в главном меню StarUML™.

- Меню File
- Меню Edit
- Меню Format
- Меню Model
- Меню View
- Меню Tools
- Меню Help Горячие клавиши

### Меню File

Меню File содержит следующие пункты меню.

Пункт меню	Описание
 New Project[Ctrl+N]	Создать новый проект.
New Project By Approach[Ctrl+I]	Открывает диалог Select New Project.
 Open[Ctrl+O]	Открывает файл проекта.
 Save[Ctrl+S]	Сохраняет файл проекта.
Save As[Ctrl+A]	Сохраняет проект в другом файле.
Close	Закрывает текущий проект.
Unit->Control Unit	Отделяет и сохраняет текущий выбранный элемент как секцию.
Unit->Uncontrol Unit	Объединяет выбранную секцию с родительской секцией (или проектом).
Unit->Delete Unit	Удаляет текущую секцию
Unit->Save Unit	Сохраняет текущую секцию как файл.
Unit->Save Unit As	Сохраняет текущую секцию как другой файл.
Import->Framework	Импортирует фреймворк в текущий проект.
Import->Model Fragment	Импортирует фрагмент модели в текущий проект.
Export->Model Fragment	Сохраняет текущий элемент как файл фрагмента модели.
Export Diagram[Shift+Ctrl+D]	Сохраняет текущую активную диаграмму как файл изображения.
Page Setup	Конфигурирует страницу для печати
 Print[Ctrl+P]	Печатает диаграмму.
Recent Files	Содержит список недавно открытых файлов.
Exit	Выходит из программы.

Рис. 5.2: Меню File

## Меню Edit

Меню Edit содержит следующие пункты меню.








Пункт меню	Описание
 Undo[Ctrl+Z]	Отменяет самое недавнее действие, выполненное пользователем.
 Redo[Ctrl+Y]	Повторяет самое недавнее действие, выполненное пользователем
 Cut[Ctrl+X]	Копирует выбранные элементы в буфер обмена и удаляет их из текущего местоположения.
Пункт меню	Описание
 Copy[Ctrl+C]	Копирует выбранные элементы в буфер обмена.
Copy Diagram[Shift+Ctrl+C]	Копирует текущую активную диаграмму в буфер обмена.
Copy Diagram as Bitmap[Shift+Ctrl+C]	Копирует текущую активную диаграмму в буфер обмена как растровый рисунок.
 Paste[Ctrl+V]	Вставляет содержание буфера обмена в текущий элемент (или диаграмму).
 Delete[Del]	Удаляет выбранные визуальные образы на диаграмме.
Delete From Model[Ctrl+Del]	Удаляет выбранные модельные элементы.
 Find[Ctrl+F]	Находит элемент.
Select All[Ctrl+A]	Выбирает все элементы в текущей диаграмме.

Рис. 5.3: Меню Edit

## Меню Format

Меню Format содержит следующие пункты меню.

Пункт меню	Описание
Font	Определяет шрифт для выбранных визуальных образов.
Line Color	Определяет цвет строки для выбранных визуальных образов.
Fill Color	Определяет цвет заливки для выбранных визуальных образов.
Line Style->Rectilinear[Ctrl+L]	Определяет тип линии выбранного элемента представления связи как прямолинейный.
Line Style->Oblique[Ctrl+B]	Определяет тип линии выбранного элемента представления связи как наклонный.
Stereotype Display->None[Shift+Ctrl+N]	Не показывать стереотип выбранных визуальных образов.
Stereotype Display->Textual[Shift+Ctrl+T]	Показывает стереотип выбранных визуальных образов в виде текста
Stereotype Display->Iconic[Shift+Ctrl+I]	Показывает стереотип выбранных визуальных образов как иконку.
Stereotype Display->Decoration[Shift+Ctrl+E]	Показывает стереотип выбранных визуальных образов как декорацию.
Suppress Attributes[Shift+Ctrl+A]	Подавляет секцию, которая отображает атрибуты для выбранных визуальных образов (например класс, прецедент, и т.д.).
Suppress Operations[Shift+Ctrl+O]	Подавляет секцию, которая отображает операции для выбранных визуальных образов (например класс, подсистема, и т.д.).
Suppress Literals[Shift+Ctrl+L]	Подавляет секцию, которая отображает атрибуты для выбранных перечислений.
Word Wrap Name	Разрешает разрывать имена выбранных визуальных образов.
Пункт меню	Описание
Show Parent Name	Показывает имя родителя в именах выбранных визуальных образов.
Show Operation Signature	Показывает сигнатуру операции выбранных визуальных образов (например класс, подсистема, и т.д.).
Show Properties	Показывает свойства элемента (например, теги и т.д.) в его визуальном образе.
Show Compartment Visibility	Показывает уровень видимости внутренних визуальных образов (например атрибутов, операций, и т.д.).
Show Compartment Stereotype	Показывает стереотипы внутренних визуальных образов (например атрибутов, операций, и т.д.).
Auto Resize	Автоматически изменяет размеры выбранных визуальных образов.
Alignment-> Bring to Front	Отображает выбранные элементы поверх остальных.
Alignment-> Send to Back	Отображает выбранные элементы под остальными.
Alignment-> Align Left	Выравнивает выбранные элементы по левому краю.
Alignment-> Align Right	Выравнивает выбранные элементы по правому краю
Alignment-> Align Middle	Выравнивает выбранные элементы горизонтально по центру
Alignment-> Align Top	Выравнивает выбранные элементы по верхнему краю.
Alignment-> Align Bottom	Выравнивает выбранные элементы по нижнему краю.
Alignment-> Align Center	Выравнивает выбранные элементы вертикально по центру .
Alignment-> Space Equally, Horizontally	Равномерно распределяет выбранные элементы горизонтально.
Alignment-> Space Equally, Vertically	Равномерно распределяет выбранные элементы вертикально.
Layout Diagram	Автоматически располагает элементы представления в текущей диаграмме.

Рис. 5.4: Меню Format

## Меню View

Меню Format содержит следующие пункты меню.

Пункт меню	Описание
Close Diagram	Закрывает текущую активную диаграмму.
Close All Diagrams	Закрывает все открытые диаграммы.
Select In Model Explorer	Показывает текущий элемент в навигаторе модели.
 Refresh	Обновляет текущую диаграмму.
Model Explorer	Переключает навигатор модели вкл. и выкл.
Diagram Explorer	Переключает навигатор диаграмм вкл. и выкл.
Properties	Переключает редактор свойств вкл. и выкл.
Documentations	Переключает документационный редактор вкл. и выкл.
Attachments	Переключает редактор вложений вкл. и выкл.
Output	Переключает окно вывода вкл. и выкл.
Messages	Переключает окно сообщения вкл. и выкл.
Toolbox	Переключает инструментарий вкл. и выкл.
Zoom->  Zoom In	Делает масштаб диаграммы большим.
Zoom->  Zoom Out	Делает масштаб диаграммы меньшим.
Zoom->  Fit To Window	Автоматически корректирует масштаб изображения так, чтобы вся диаграмма поместилась в окне.
Zoom->xxx%	Показывает текущую диаграмму в масштабе xxx%
Toolbars->Standard	Переключает инструментальная панель Standard вкл. и выкл.
Toolbars->Format	Переключает инструментальная панель Format вкл. и выкл.
Toolbars->View	Переключает инструментальная панель View вкл. и выкл.
Toolbars->Alignment	Переключает инструментальная панель Align вкл. и выкл.

Рис. 5.5: Меню View

## Горячие клавиши

StarUML™ обеспечивает горячие клавиши к функциям меню.

Command	MacOS	Windows
File > New	Cmd+N	Ctrl+N
File > Open	Cmd+O	Ctrl+O
File > Save	Cmd+S	Ctrl+S
File > Save As	Cmd+Shift+S	Ctrl+Shift+S
File > Preferences	Cmd+,	n/a
File > Quit	Cmd+Q	Ctrl+Q
Edit > Undo	Cmd+Z	Ctrl+Z
Продолжение на следующей странице		

**Таблица 5.1 – продолжение**

<b>Command</b>	<b>MacOS</b>	<b>Windows</b>
Edit > Redo	Cmd+Y	Ctrl+Y
Edit > Cut	Cmd+X	Ctrl+X
Edit > Copy	Cmd+C	Ctrl+C
Edit > Copy Diagram As Image	Cmd+Shift+C	Ctrl+Shift+C
Edit > Paste	Cmd+V	Ctrl+V
Edit > Delete	Delete	Delete
Edit > Delete from Model	Cmd+Delete	Ctrl+Delete
Edit > Move Up	Cmd+Shift+Up	Ctrl+Shift+Up
Edit > Move Down	Cmd+Shift+Down	Ctrl+Shift+Down
Edit > Select All	Cmd+A	Ctrl+A
Edit > Select In Explorer	Cmd+E	Ctrl+E
Edit > Select In Diagram	Cmd+D	Ctrl+D
Format > Font	Cmd+Shift+F	Ctrl+Shift+F
Format > Fill Color	Cmd+Shift+I	Ctrl+Shift+I
Format > Line Color	Cmd+Shift+L	Ctrl+Shift+L
Format > Line Style > Rectilinear	Cmd+L	Ctrl+L
Format > Line Style > Oblique	Cmd+B	Ctrl+B
Format > Line Style > RoundRect	Cmd+Option+L	Ctrl+Alt+L
Format > Line Style > Curve	Cmd+Option+B	Ctrl+Alt+B
Format > Auto Resize	Cmd+Shift+R	Ctrl+Shift+R
Format > Show Shadow	Cmd+Shift+H	Ctrl+Shift+H
Format > Stereotype Display > None	Cmd+Shift+0	Ctrl+Shift+0
Format > Stereotype Display > Label	Cmd+Shift+1	Ctrl+Shift+1
Format > Stereotype Display > Decoration	Cmd+Shift+2	Ctrl+Shift+2
Format > Stereotype Display > Decoration Label	Cmd+Shift+3	Ctrl+Shift+3
Format > Stereotype Display > Icon	Cmd+Shift+4	Ctrl+Shift+4
Format > Stereotype Display > Icon Label	Cmd+Shift+5	Ctrl+Shift+5
Format > Word Wrap	Cmd+Shift+W	Ctrl+Shift+W
Format > Show Visibility	Cmd+Shift+V	Ctrl+Shift+V
Format > Show Namespace	Cmd+Shift+N	Ctrl+Shift+N
Format > Show Property	Cmd+Shift+B	Ctrl+Shift+B
Format > Show Type	Cmd+Shift+Y	Ctrl+Shift+Y
Format > Show Multiplicity	Cmd+Shift+M	Ctrl+Shift+M
Format > Show Operation Signature	Cmd+Shift+G	Ctrl+Shift+G
Format > Suppress Attributes	Cmd+Shift+A	Ctrl+Shift+A
Format > Suppress Operations	Cmd+Shift+O	Ctrl+Shift+O
Format > Suppress Receptions	Cmd+Shift+E	Ctrl+Shift+E
Format > Suppress Literals	Cmd+Shift+T	Ctrl+Shift+T
Format > Suppress Columns	Cmd+Shift+U	Ctrl+Shift+U
Продолжение на следующей странице		



**Таблица 5.1 – продолжение**

<b>Command</b>	<b>MacOS</b>	<b>Windows</b>
Model > Find	Cmd+F	Ctrl+F
View > Command Palette...	Cmd-Shift-P	Ctrl-Shift-P
View > Close Diagram	F4	F4
View > Close Other Diagrams	Cmd+F4	Ctrl+F4
View > Close All Diagrams	Shift+F4	Shift+F4
View > Next Diagram	Cmd+Shift+]	Ctrl+Shift+]
View > Previous Diagram	Cmd+Shift+[	Ctrl+Shift+[
View > Zoom In	Cmd++	Ctrl++
View > Zoom Out	Cmd+-	Ctrl+-
View > Actual Size	Cmd+0	Ctrl+0
View > Fit To Window	Cmd+Option+1	Ctrl+Alt+1
View > Show Grid	Cmd+G	Ctrl+G
View > Sidebar	Cmd+1	Ctrl+1
View > Navigator	Cmd+2	Ctrl+2
View > Toolbar	Cmd+3	Ctrl+3
View > Statusbar	Cmd+4	Ctrl+4
View > Toolbox	Cmd+5	Ctrl+5
View > Editors	Cmd+6	Ctrl+6
View > Diagram Thumbnails	Cmd+Option+T	Ctrl+Alt+T
View > Markdown Documentation	Cmd+Option+D	Ctrl+Alt+D
View > Minimap	Cmd+Option+M	Ctrl+Alt+M
View > Relationships	Cmd+Option+R	Ctrl+Alt+R
Debug > Show DevTools	Shift+Option+T	Shift+Alt+T
Debug > Reload	Cmd+R	Ctrl+R

## **Панели инструментов**

Эта секция описывает все элементы инструментальной панели StarUML™ TM.

- Панель Standard. Включает команды меню File, Edit и Model.
- Панель Format. Включает команды меню Format.
- Панель View . Включает команды меню View.
- Панель Align. Включает команды выравнивания из меню Format.
- Панель Pallet. Содержит клавиши для размещения визуальных образов на диаграммах.

### **Общие инструменты палитры**

Следующие функции всегда доступны в инструментальной палитре независимо от типов диаграмм.








Функция	Описание
 Select	Самый основной инструмент, который выбирает, перемещает или изменяет размеры элемента в диаграмме.
 Note	Создает элемент примечания в текущей диаграмме
 Note Link	Связывает примечание с элементом диаграммы
 Text	Создает строковый элемент в текущей диаграмме
 Rectangle	Создает прямоугольник в текущей диаграмме.
 Ellipse	Создает эллипс в текущей диаграмме.
 Rounded Rectangle	Создает округленный прямоугольник в текущей диаграмме.

Рис. 5.6: Общие инструменты

## Инструменты палитры ориентированные на разные типы диаграмм

Следующие функции создают элементы для диаграмм разных типов.












Функция	Описание	Все диаграммы
 Subsystem	Создаёт подсистему в текущей диаграмме	Диаграмма классов
 Package	Создаёт пакет в текущей диаграмме.	Диаграмма классов, Диаграмма компонентов, Диаграмма развёртывания, Диаграмма прецедентов
 Class	Создаёт класс в текущей диаграмме	Диаграмма классов, Композиционная диаграмма
 Interface	Создаёт интерфейс в текущей диаграмме	Диаграмма классов, Диаграмма компонентов, Композиционная диаграмма
 Enumeration	Создаёт перечислимый тип в текущей диаграмме	Диаграмма классов
 Signal	Создаёт сигнал в текущей диаграмме	Диаграмма классов
 Excerpt	Создаёт прерывание в текущей диаграмме	Диаграмма классов
 Component	Создаёт компонент в текущей диаграмме	Диаграмма компонентов
 ComponentInstance	Создаёт инстанцию компонента в текущей диаграмме	Диаграмма компонентов
 Node	Создаёт узел в текущей диаграмме	Диаграмма развёртывания
 NodeInstance	Создаёт инстанцию узла в текущей диаграмме	Диаграмма компонентов, Диаграмма развёртывания

Рис. 5.7: Особые инструменты. Ч1




















 Artifact	Создаёт артефакт в текущей диаграмме	Диаграмма прецедентов
 UseCase	Создаёт прецедент в текущей диаграмме	Диаграмма прецедентов
 Actor	Создаёт актора в текущей диаграмме	Диаграмма прецедентов
 SystemBoundary	Создаёт системную границу в текущей диаграмме	Диаграмма прецедентов
 Object	Создаёт объект в текущей диаграмме	Диаграмма классов, Диаграмма сообщений, Диаграмма коллабораций
 Part	Создаёт часть классификатора в текущей диаграмме	Диаграмма классов, Диаграмма компонентов, Диаграмма развёртывания, Композиционная диаграмма
 Port	Создаёт порт классификатора в текущей диаграмме	Диаграмма классов, Диаграмма компонентов, Диаграмма развёртывания, Композиционная диаграмма
 ClassifierRole	Создаёт роль классификатора в текущей диаграмме	Диаграмма сообщений ролей, Диаграмма коллаборации ролей
 Combined Fragment	Создаёт комбинированный фрагмент в текущей диаграмме	Диаграмма сообщений, Диаграмма сообщений ролей, Диаграмма коллабораций, Диаграмма коллаборации ролей
 Interaction Operand	Создаёт операнд взаимодействия с комбинированным фрагментом в текущей диаграмме	Диаграмма сообщений, Диаграмма сообщений ролей, Диаграмма коллабораций, Диаграмма коллаборации ролей
 Frame	Создаёт фрейм в текущей диаграмме	Диаграмма сообщений, Диаграмма сообщений ролей, Диаграмма коллабораций, Диаграмма коллаборации ролей
 CompositeState	Создаёт композитное состояние в текущей диаграмме	Диаграмма состояний
 SubmachineState	Создаёт а суб-модель состояний в текущей диаграмме	Диаграмма состояний
 InitialState	Создаёт начальное состояние (Псевдосостояние) в текущей диаграмме	Диаграмма состояний, Диаграмма действий
 FinalState	Создаёт конечное состояние в текущей диаграмме	Диаграмма состояний, Диаграмма действий
 Flow Final	Создаёт глубокую хронологию(Состояние конечного потока) в текущей диаграмме	Диаграмма состояний, Диаграмма действий
 ChoicePoint	Создаёт выбор (Псевдосостояние) в текущей диаграмме	Диаграмма состояний
 JunctionPoint	Создаёт соединение (Псевдосостояние) element в текущей диаграмме	Диаграмма состояний
 ShallowHistory	Создаёт поверхностную хронологию (Псевдосостояние) в текущей диаграмме	Диаграмма состояний

Рис. 5.8: Особые инструменты. Ч2

 DeepHistory	Создаёт глубокую хронологию (Псевдосостояние) в текущей диаграмме	Диаграмма состояний
 Synchronization	Создаёт синхронизацию (Псевдосостояние) в текущей диаграмме	Диаграмма состояний, Диаграмма действий
 ActionState	Создаёт действие в текущей диаграмме	Диаграмма действий
 SubactivityState	Создаёт суб-действие в текущей диаграмме	Диаграмма действий
 Decision	Создаёт решение (Псевдосостояние) в текущей диаграмме	Диаграмма действий
 ObjectFlow	Создаёт объектный поток в текущей диаграмме	Диаграмма действий
 Signal Accept State	Создаёт состояние принятия сигнала в текущей диаграмме	Диаграмма действий
 Signal Send State	Создаёт состояние послыки сигнала в текущей диаграмме	Диаграмма действий
 Swimlane (Vertical)	Создаёт вертикальную область действия в текущей диаграмме	Диаграмма действий
 Swimlane (Horizontal)	Создаёт горизонтальную область действия в текущей диаграмме	Диаграмма действий
 Association	Проводит семантическую ассоциацию между двумя классами в текущей диаграмме	Диаграмма классов, Диаграмма компонентов, Диаграмма развёртывания, Диаграмма прецедентов
 Directed Association	Проводит семантическую ассоциацию между двумя классами в текущей диаграмме	Диаграмма классов, Диаграмма развёртывания, Диаграмма прецедентов
 Aggregation	Проводит семантическую ассоциацию между двумя классами в текущей диаграмме	Диаграмма классов
 Composition	Проводит семантическую ассоциацию между двумя классами в текущей диаграмме	Диаграмма классов
 Generalization	Соединяет обобщаемый и специализируемый элементы отношением обобщения в текущей диаграмме	Диаграмма классов, Диаграмма прецедентов
 Dependency	Соединяет два элемента отношением зависимости в текущей диаграмме	Диаграмма классов, Диаграмма компонентов, Диаграмма развёртывания, Диаграмма прецедентов, Композиционная диаграмма
 Realization	Соединяет спецификацию элемента и его реализацию отношением реализации в текущей диаграмме	Диаграмма классов, Диаграмма компонентов, Композиционная диаграмма
 AssociationClass	Соединяет класс и ассоциацию в текущей диаграмме так, что ассоциация сама становится подобной классу.	Диаграмма классов
 Include	Соединяет два прецедента отношением включения в текущей диаграмме так, что один прецедент включает поведение другого.	Диаграмма прецедентов

Рис. 5.9: Особые инструменты. ЧЗ

 DeepHistory	Создаёт глубокую хронологию (Псевдосостояние) в текущей диаграмме	Диаграмма состояний
 Synchronization	Создаёт синхронизацию (Псевдосостояние) в текущей диаграмме	Диаграмма состояний, Диаграмма действий
 ActionState	Создаёт действие в текущей диаграмме	Диаграмма действий
 SubactivityState	Создаёт суб-действие в текущей диаграмме	Диаграмма действий
 Decision	Создаёт решение (Псевдосостояние) в текущей диаграмме	Диаграмма действий
 ObjectFlow	Создаёт объектный поток в текущей диаграмме	Диаграмма действий
 Signal Accept State	Создаёт состояние принятия сигнала в текущей диаграмме	Диаграмма действий
 Signal Send State	Создаёт состояние послыки сигнала в текущей диаграмме	Диаграмма действий
 Swimlane (Vertical)	Создаёт вертикальную область действия в текущей диаграмме	Диаграмма действий
 Swimlane (Horizontal)	Создаёт горизонтальную область действия в текущей диаграмме	Диаграмма действий
 Association	Проводит семантическую ассоциацию между двумя классами в текущей диаграмме	Диаграмма классов, Диаграмма компонентов, Диаграмма развёртывания, Диаграмма прецедентов
 Directed Association	Проводит семантическую ассоциацию между двумя классами в текущей диаграмме	Диаграмма классов, Диаграмма развёртывания, Диаграмма прецедентов
 Aggregation	Проводит семантическую ассоциацию между двумя классами в текущей диаграмме	Диаграмма классов
 Composition	Проводит семантическую ассоциацию между двумя классами в текущей диаграмме	Диаграмма классов
 Generalization	Соединяет обобщаемый и специализируемый элементы отношением обобщения в текущей диаграмме	Диаграмма классов, Диаграмма прецедентов
 Dependency	Соединяет два элемента отношением зависимости в текущей диаграмме	Диаграмма классов, Диаграмма компонентов, Диаграмма развёртывания, Диаграмма прецедентов, Композиционная диаграмма
 Realization	Соединяет спецификацию элемента и его реализацию отношением реализации в текущей диаграмме	Диаграмма классов, Диаграмма компонентов, Композиционная диаграмма
 AssociationClass	Соединяет класс и ассоциацию в текущей диаграмме так, что ассоциация сама становится подобной классу.	Диаграмма классов
 Include	Соединяет два прецедента отношением включения в текущей диаграмме так, что один прецедент включает поведение другого.	Диаграмма прецедентов

Рис. 5.10: Особые инструменты. Ч4









 Extend	Соединят два прецедента отношением расширения в текущей диаграмме так, что один прецедент может быть расширен поведением другого.	Диаграмма прецедентов
 AssociationRole	Соединяет две роли ассоциацией ролей в текущей диаграмме	Диаграмма коллаборации ролей
 SelfAssociationRole	Создаёт ассоциацию роли к себе самой в текущей диаграмме	Диаграмма коллаборации ролей
 Link	Соединяет два объекта в текущей диаграмме	Диаграмма классов, Диаграмма коллабораций
 SelfLink	Соединяет объект с самим собой в текущей диаграмме	Диаграмма классов, Диаграмма коллабораций
 ForwardMessage	Определяет сообщение между двумя ролями в текущей диаграмме	Диаграмма сообщений ролей, Диаграмма коллаборации ролей
 ReverseMessage	Определяет сообщение между двумя ролями в текущей диаграмме	Диаграмма сообщений ролей, Диаграмма коллаборации ролей
 SelfMessage	Создаёт сообщение роли себе самой в текущей диаграмме	Диаграмма сообщений ролей, Диаграмма коллаборации

Рис. 5.11: Особые инструменты. Ч5



## Глава 6

# Построение диаграмм UML с помощью ПО StarUML™

### 6.1 Особенности разработки диаграмм вариантов использования

Работа над моделью в среде SrarUML™ начинается с общего анализа проблемы и построения диаграммы вариантов использования, которая отражает функциональное назначение проектируемой программной системы. В качестве проекта далее будет рассматриваться процесс создания новой модели велосипеда на промышленном предприятии.

Для изменения имени проекта, предложенного программой по умолчанию, следует сохранить модель во внешнем файле на диске, например, под именем CycleComp.mdj. В этом случае изменится имя в строке заголовка и имя проекта в иерархическом представлении модели в браузере проекта. Настройка шрифтов, цвета линий и графических элементов производится или в меню «Format». В левом нижнем углу в блоке Editors. Характерной особенностью среды является возможность работы с символами кириллицы. Однако следует заметить, что при спецификации элементов модели с последующей генерацией текста программного кода следует записывать имена и свойства классов, ассоциаций, атрибутов, операций и компонентов символами того языка, который поддерживается соответствующим языком программирования. Для разработки диаграммы вариантов использования модели в среде StarUML™ необходимо активизировать соответствующую диаграмму в окне диаграммы. Это можно сделать следующими способами:

- Перейдя в пункт меню *Model → AddDiagram → UseCaseDiagram*
- Нажав правой кнопкой мыши на рабочей области и перейдя по *AddDiagram → UseCaseDiagram*

При этом появляется новое окно с чистым рабочим листом диаграммы вариантов использования и специальная панель инструментов, содержащая кнопки с изображением графических элементов, необходимых для разработки *диаграммы вариантов использования*. См рис. 6.1. Назначение кнопок приведены в таблице 6.1.

## Добавление актера на диаграмму вариантов использования и редактирование его свойств

Для добавления актера на диаграмму варианта использования нужно с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы актера на специальной панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы. На диаграмме появится изображение актера с маркерами изменения его геометрических размеров и предложенным программой именем по умолчанию *Actor*. Для разрабатываемой модели предложенное программой имя актера следует изменить на *Инженер-конструктор*. См. рис. 6.2.

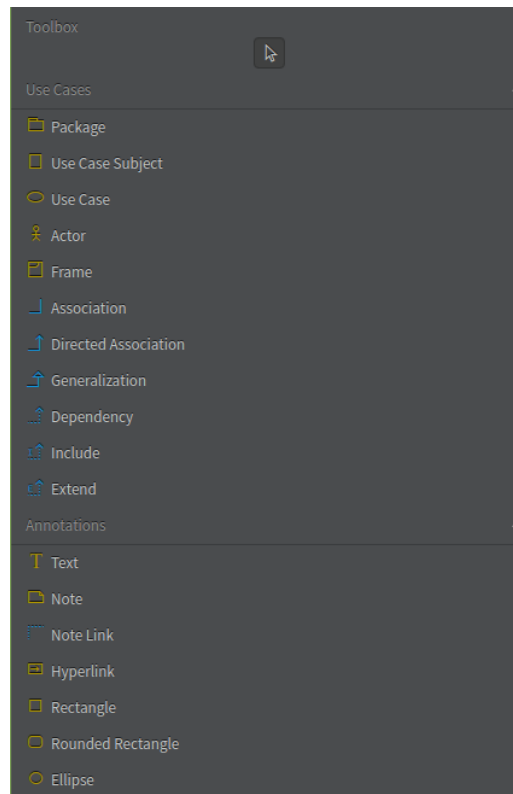


Рис. 6.1: Use Case: Панель инструментов



Рис. 6.2: Use Case: Actor



Всплывающая подсказка	Назначение кнопки
Selection Tool	Превращает изображение курсора в форму стрелки для последующего выделения элементов на диаграмме
Text Box	Добавляет на диаграмму текстовую область
Note	Добавляет на диаграмму примечание
Anchor Note to Item	Добавляет на диаграмму связь примечания с соответствующим графическим элементом диаграммы
Package	Добавляет на диаграмму пакет
Use Case	Добавляет на диаграмму вариант использования
Actor	Добавляет на диаграмму актера
Unidirectional Association	Добавляет на диаграмму направленную ассоциацию
Dependency or Instantiates	Добавляет на диаграмму отношение зависимости
Generalization	Добавляет на диаграмму отношение обобщения

Таблица 6.1: Use Case: Назначение кнопок

Чтобы изменить графические размеры изображения элемента модели, прежде всего, следует щелчком левой кнопки мыши выделить его в рабочей области диаграммы.

Имя размещенного на диаграмму элемента разработчик может изменить либо сразу после добавления элемента на диаграмму, либо в ходе последующей работы над проектом. Для любого графического элемента модели по щелчку правой кнопкой мыши на выбранном элементе вызывается контекстное меню данного элемента. Подробнее ознакомиться с параметрами выбранного элемента можно в области *Editors*, расположенной справа от рабочей области. Для добавленного актера «*Инженер-конструктор*» окно спецификации свойств выглядит следующим образом. См рис. 6.3.

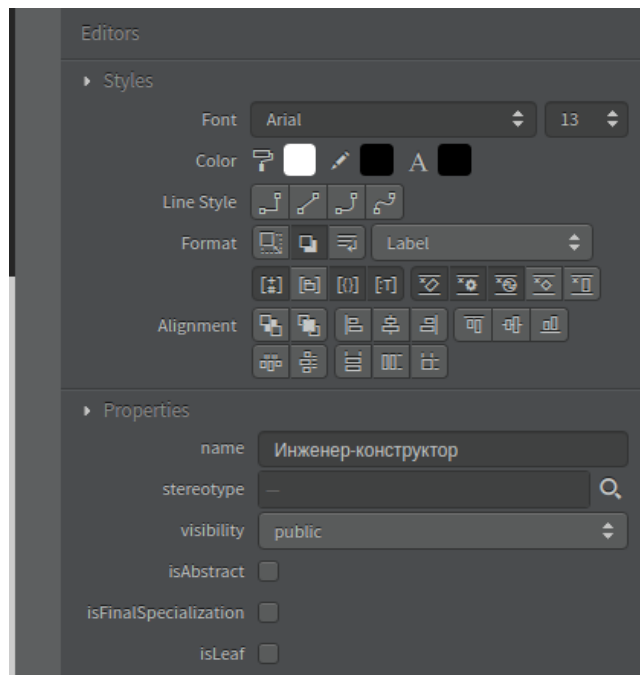


Рис. 6.3: Спецификация актера

В среде StarUML™ актер является классом, для него некорректно специфицировать атрибуты и операции, поскольку актер является внешней по отношению к разрабатываемой системе сущностью. Для актера *Инженер-конструктор* можно уточнить его назначение в модели. С этой целью следует изменить его стереотип и добавить текст документации. Для добавления текста документации в секцию Documentation следует ввести текст: «Физическое лицо, осуществляющее инженерную деятельность» и нажать кнопку Apply (Применить) или ОК. После изменения данных свойств актера *Инженер-конструктор* окно спецификации свойств будет выглядеть следующим образом 6.4.

### Добавление и редактирование варианта использования

Для добавления варианта использования на диаграмму нужно с помощью левой кнопки мыши нажать кнопку с изображением варианта использования на специальной панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте диаграммы. На диаграмме появится изображение варианта использования с маркерами изменения его геометрических размеров и предложенным программой именем по умолчанию *UseCase*. Для разрабатываемой модели предприятия предложенное программой имя варианта использования следует изменить на *Спроектировать изделие* (рис. 6.5).

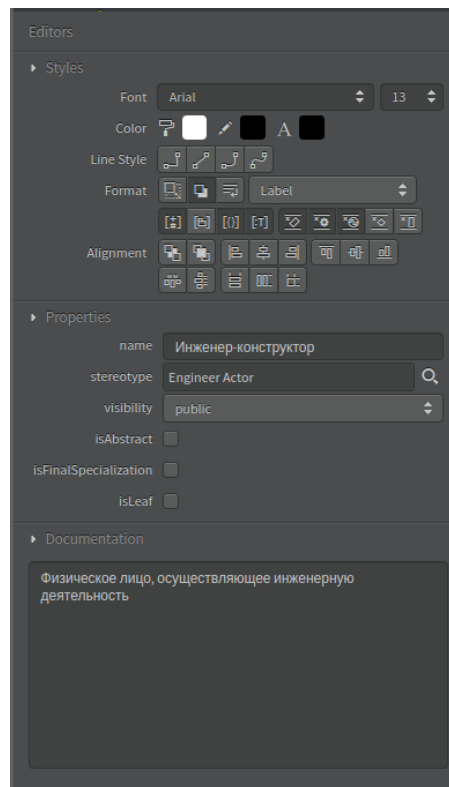


Рис. 6.4: Измененная спецификация актера

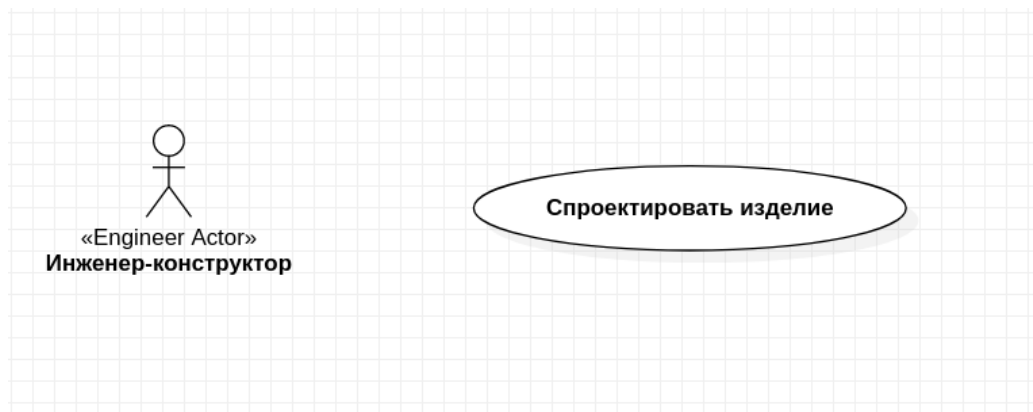


Рис. 6.5: Вариант использования

Для уточнения свойств данного варианта использования следует открыть диалоговое окно спецификации его свойств, например, с помощью двойного щелчка левой кнопкой мыши на изображении этого элемента на диаграмме. Для изменения стереотипа во вложенном списке Stereotype нужно выбрать строку Engineer Use Case. Для добавления текста документации в секцию Documentation следует ввести текст: «Основной вариант использования для разрабатываемой модели предприятия» и нажать кнопку Apply (Применить) или OK. После изменения данных свойств варианта использования окно спецификации его свойств будет выглядеть следующим образом (рис. 6.6).

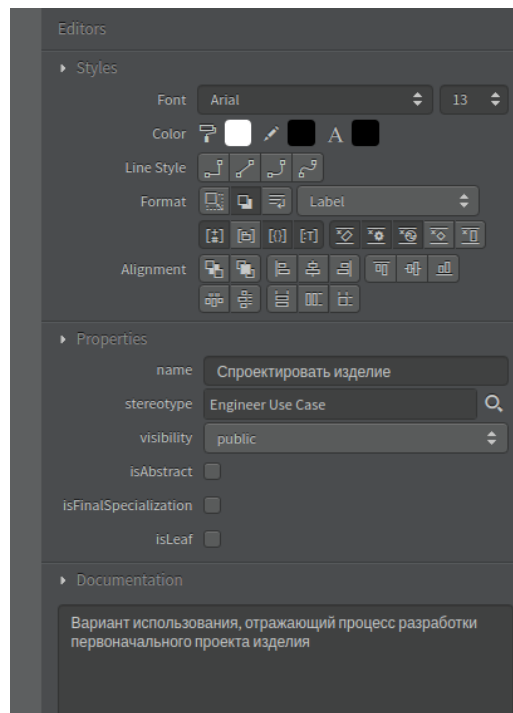


Рис. 6.6: Дополненная спецификация варианта использования

### Добавление ассоциации

Для добавления ассоциации между актером и вариантом использования на диаграмму нужно с помощью левой кнопки мыши нажать на специальной панели инструментов кнопку с изображением пиктограммы направленной ассоциации, отпустить левую кнопку мыши, щелкнуть левой кнопкой мыши на изображении актера на диаграмме и отпустить ее на изображении варианта использования. В результате этих действий на диаграмме появится изображение ассоциации, соединяющей актера с вариантом использования. См. рис. 6.7. При

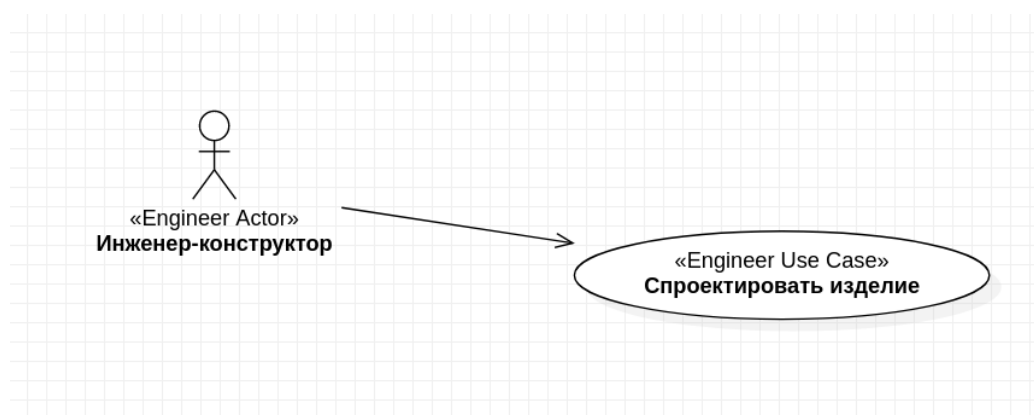


Рис. 6.7: Ассоциация актера с вариантом использования

необходимости можно сделать направленную ассоциацию ненаправленной, для чего следует воспользоваться секцией *Editors*, расположенной справа, под Обозревателем Модели (*Model Explorer*). Следует убрать отметку строки выбора *Navigable* (Навигация) (рис. 6.8).

Рис. 6.8: Свойство Navigable

### Добавление отношения зависимости и редактирование его свойств

Для добавления отношения зависимости между двумя вариантами использования на диаграмму необходимо предварительно рассмотренным выше способом добавить второй вариант использования с именем *Изготовить тестовый образец*. После этого с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы зависимости на специальной панели инструментов, отпустить левую кнопку мыши, щелкнуть левой кнопкой мыши на изображении варианта использования *Спроектировать изделие* и отпустить ее на изображении варианта использования *Изготовить тестовый образец*.

В результате этих действий на диаграмме появится изображение отношения зависимости, которое соединяет два выбранных варианта использования. Поскольку вариант использования *Спроектировать изделие* выполняется всегда, для добавленного отношения зависимости дополнительно следует указать текстовый стереотип «include». Выполнить это можно уже известным способом с помощью диалогового окна спецификации свойств этого отношения и выбора нужного стереотипа из предлагаемого списка.

После задания для данного отношения зависимости стереотипа «include» текст этого стереотипа в угловых скобках появится рядом с изображением пунктирной линии зависимости, связывающей соответствующие варианты использования (рис. 6.9). С целью лучшей визуализации диаграммы текстовую область стереотипа можно переместить в нужное место диаграммы. Выполнить это можно с помощью общего способа перемещения графических элементов модели, который был рассмотрен ранее применительно к актеру *Инженер-конструктор*.



Рис. 6.9: Отношение зависимости между вариантами использования

Аналогичным образом могут быть добавлены на диаграмму вариантов использования отношения зависимости со стереотипом «extend», которые применяются для моделирования исключений при выполнении отдельных вариантов использования.

### Удаление графического элемента

Для удаления любого графического элемента с диаграммы его следует выделить на диаграмме и нажать клавишу *Delete* на клавиатуре. При этом выделенный элемент будет удален с активной диаграммы, но не из модели. Для удаления элемента не только из диаграммы, но и из модели проекта необходимо выбрать пункт *Delete From Model* в диалоговом окне, появляющимся при попытке удаления объекта.

### Построение диаграммы

Завершим построение *Диаграммы Вариантов использования*, добавив необходимых актеров и соответствующие варианты использования. См. рис. 6.10.

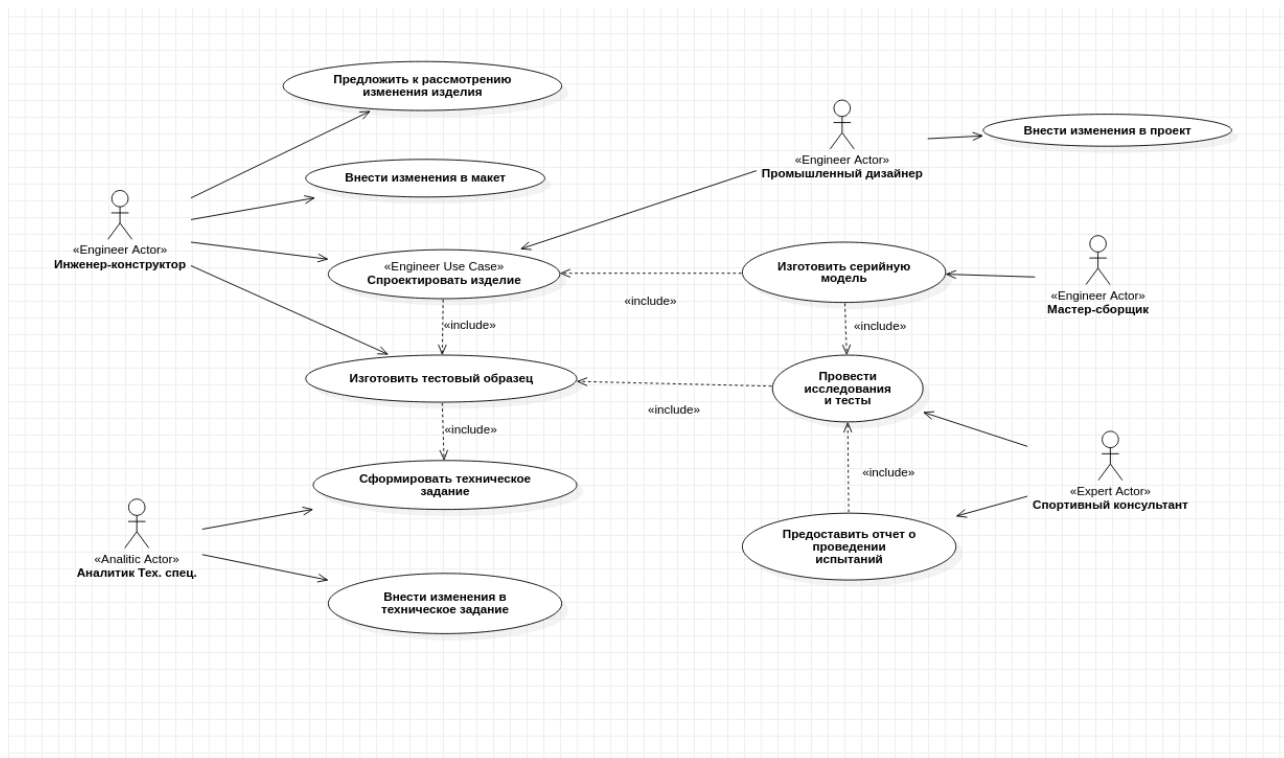


Рис. 6.10: Диаграмма вариантов использования

Напомним, что диаграмма вариантов использования является высокоуровневым концептуальным представлением модели, поэтому она не должна содержать слишком много вариантов использования и актеров. В последующем построенная диаграмма может быть изменена посредством добавления новых элементов, таких как варианты использования и актеры, или их удаления.

При работе с отношениями на диаграмме вариантов использования следует помнить о назначении соответствующих отношений в нотации языка UML. Речь идет о том, что если для двух элементов выбранный вид отношения не является допустимым, то в большинстве случаев программа StarUML™ сообщит об этом разработчику, и соответствующая линия связи не будет добавлена на диаграмму. После окончания сеанса работы над проектом выполненную работу необходимо сохранить в файле проекта с расширением «.mdj».

## 6.2 Особенности разработки диаграмм классов

Диаграмма классов является основным логическим представлением модели и содержит детальную информацию о внутреннем устройстве объектно-ориентированной программной системы или, используя современную терминологию, об архитектуре программной системы. Активизировать рабочее окно диаграммы классов можно несколькими способами:

- окно диаграммы классов появляется по умолчанию в рабочем окне диаграммы после создания нового проекта;
- выбрать соответствующий вариант, перейдя по следующим пунктам меню:  
*Model* → *AddDiagram* → *ClassDiagram*;

- нажать правую кнопку мыши на рабочей области и выбрать аналогичные пункты контекстного меню: *AddDiagram* → *ClassDiagram*;

При этом появляется новое окно с чистым рабочим листом диаграммы классов и специальная панель инструментов, содержащая кнопки с изображением графических примитивов, необходимых для разработки диаграммы классов (рис. 6.11). Назначение отдельных кнопок панели можно узнать также из всплывающих подсказок. Приведем назначение некоторых инструментов в таблице 6.2.

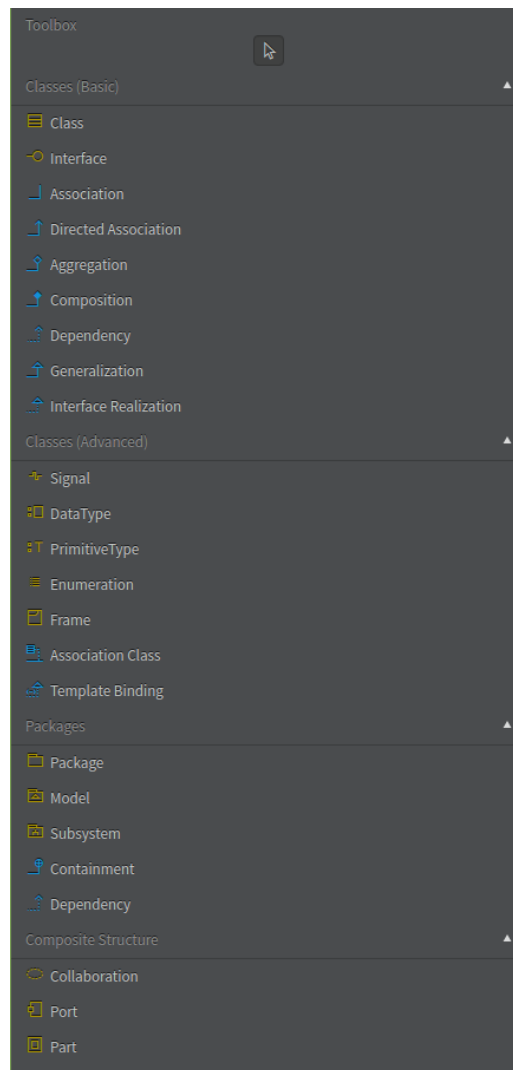


Рис. 6.11: Class Diagram:Панель инструментов

### Добавление класса на диаграмму классов и редактирование его свойств

Для добавления класса на диаграмму классов нужно с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы класса на специальной панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы. На диаграмме появится изображение класса с маркерами изменения его геометрических размеров и предложенным средой именем по умолчанию *Class*.



Всплывающая подсказка	Назначение кнопки
Selection Tool	Превращает изображение курсора в форму стрелки для последующего выделения элементов на диаграмме
Text Box	Добавляет на диаграмму текстовую область
Note	Добавляет на диаграмму примечание
Anchor Note to Item	Добавляет на диаграмму связь примечания с соответствующим графическим элементом диаграммы
Class	Добавляет на диаграмму класс
Interface	Добавляет на диаграмму интерфейс
Unidirectional Association	Добавляет на диаграмму направленную ассоциацию
Association Class	Добавляет на диаграмму ассоциацию класс
Package	Добавляет на диаграмму пакет
Dependency or Instantiates	Добавляет на диаграмму отношение зависимости
Generalization	Добавляет на диаграмму отношение обобщения
Realize	Добавляет на диаграмму отношение реализации

Таблица 6.2: Class Diagram: Назначение пунктов меню

Продолжая разработку модели предприятия по производству велосипедов в качестве сквозного примера проекта, построим для этой модели следующую каноническую диаграмму — диаграмму классов. С этой целью следует изменить предложенное по умолчанию имя диаграммы Main на ClassDiagramCycle, а имя добавленного на диаграмму класса — на *Конструкторский отдел* 6.12.

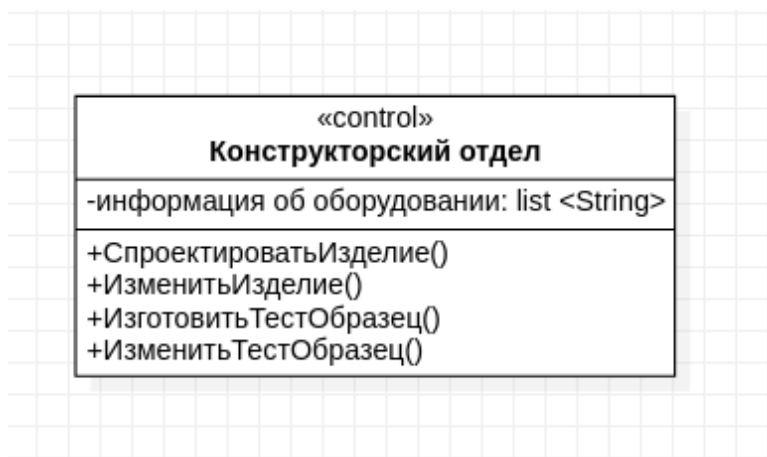


Рис. 6.12: Класс «Конструкторский отдел»

Поскольку разрабатываемая модель предприятия на начальных этапах работы над проектом используется для анализа общей архитектуры проекта и согласования ее с различными участниками рабочей группы, имена классов, их атрибутов и операций для большей наглядности и понимания задаются на русском языке с пробелами и записываются символами кириллицы.

В последующем по мере выполнения проекта и реализации модели на некотором языке программирования, имена соответствующих классов, атрибутов и операций должны быть преобразованы в символы латиницы. При этом имена этих элементов модели должны быть

записаны без пробелов. В контексте управляемой моделью архитектуры первую модель еще называют независимой от платформы реализации, а вторую — зависимой от платформы реализации.

Для класса *Конструкторский отдел* можно уточнить его назначение в модели с помощью указания стереотипа и пояснительного текста в форме документации. С этой целью щелчком левой кнопкой мыши на изображении этого класса на диаграмме или в браузере проекта открыть секцию *Editors* этого класса и в разделе *Properties* в строке *stereotype* указать стереотип «control». При этом стереотип отобразится над именем класса в угловых кавычках (рис. 6.13).

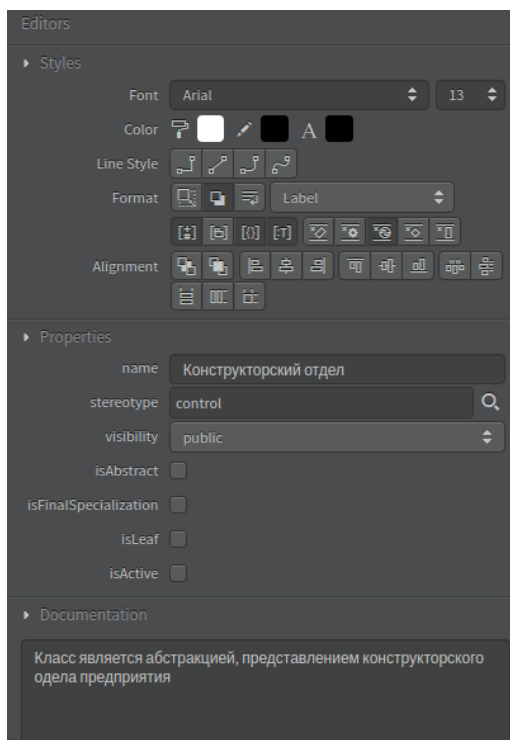


Рис. 6.13: Секция Editors

Выбор данного стереотипа означает, что соответствующий класс предназначен для проведения каких-либо процессов, изменяющих состояние сущностей. Данный класс также отвечает за координацию действий других классов.

Далее в секцию документации данного класса можно ввести поясняющий текст: «Класс является абстракцией, представлением конструкторского отдела предприятия» и нажать кнопку *Apply* или *OK*, чтобы сохранить результаты редактирования свойств выбранного класса.

В разделе *Properties* секции *Editors* можно настроить дополнительные характеристики класса:

- Настроить видимость класса характеристикой *visibility* (public, private, protected, package);
- Указать класс как абстрактный (*isAbstract*);
- Указать реализацию класса как финальную (*isLeaf*);
- Указать класс активным (*isActive*).

В той же секции, кликнув на метод конкретного класса можно специфицировать условия

на возможность реализации объектов класса в параллельных потоках управления. Для выбора могут быть использованы следующие свойства:

- Sequential (Последовательный) — свойство по умолчанию, которое означает, что объекты класса будут вести себя нормально только при наличии одного потока управления, т. е. соответствующие операции объектов должны выполняться последовательно. В то же время при наличии нескольких потоков управления стабильное поведение объектов класса не гарантируется.
- Guarded (Безопасный) — означает, что при наличии нескольких потоков управления объекты класса будут вести себя ожидаемым от них образом. Для этого объекты в различных потоках должны взаимодействовать друг с другом для того, чтобы гарантировать отсутствие конфликта между ними.
- Concurrent (Параллельный) — означает, что объекты класса будут вести себя ожидаемым от них образом при наличии нескольких потоков управления. При этом нет необходимости во взаимодействии объектов в различных потоках управления, поскольку объекты данного класса могут самостоятельно разрешать возможные конфликты.

Продолжая разработку модели предприятия, добавим на диаграмму второй класс с именем *Управляющий отдел*, для которого в окне спецификации свойств выберем стереотип «*boundary*» (граничный класс), а в качестве документации введем текст: «Реализует поведение аналитического отдела. Иницирует работу над новым изделием и вводит изделие в модельный ряд».

### Добавление ассоциации на диаграмму классов и редактирование ее свойств

Добавление на диаграмму ассоциации между двумя классами выполняется следующим образом. На специальной панели инструментов необходимо нажать кнопку с изображением пиктограммы направленной ассоциации и отпустить левую кнопку мыши. Если ассоциация направленная, то на диаграмме классов надо выделить первый элемент ассоциации или источник, от которого исходит стрелка, и, не отпуская нажатую левую кнопку мыши, переместить ее указатель ко второму элементу отношения или приемнику, к которому направлена стрелка.

После перемещения ко второму элементу кнопку мыши следует отпустить, в результате чего на диаграмму классов будет добавлена направленная ассоциация между двумя выбранными классами. Продолжая разработку диаграммы классов модели предприятия, добавим на нее описанным способом направленную ассоциацию между классами *Управляющий отдел* и *Конструкторский отдел*.

Укажем имя для данной связи. Это можно выполнить с помощью окна спецификации свойств ассоциации. Доступ к секции *Editors* можно получить после выделения линии ассоциации на диаграмме классов или в обозревателе проекта и двойного щелчка левой кнопкой мыши.

Для отдельного класса можно уточнить также и другие его свойства, доступные для редактирования на вкладке *Editors* окна спецификации свойств этого класса. Например, на этой

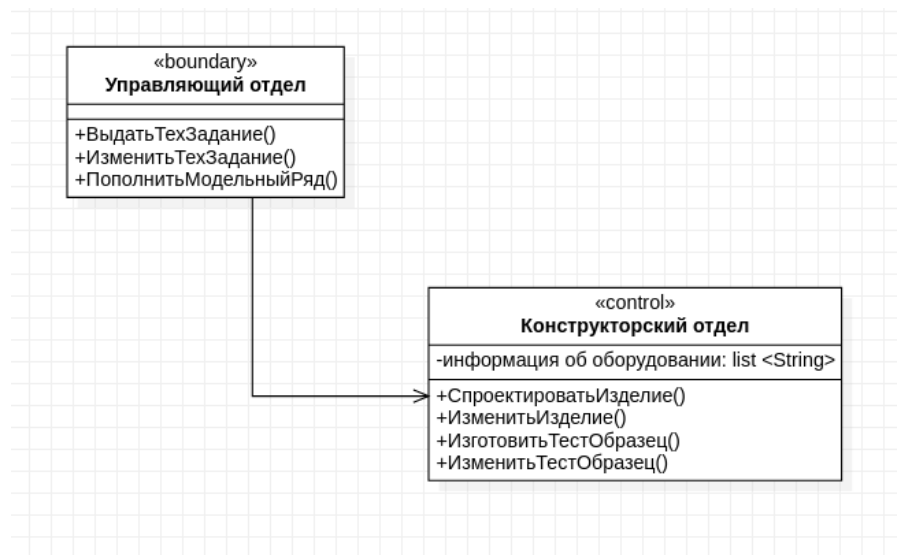


Рис. 6.14: Ассоциация

вкладке с помощью атрибута *Multiplicity* (Кратность) можно задать количество объектов или экземпляров данного класса. В данном случае для классов *Конструкторский отдел* и *Управляющий отдел*, связанных ассоциацией, укажем кратность «один ко многим» (рис. 6.15). Данная кратность указывает на то, что в ведении аналитического отдела может находиться один или более конструкторских отделов. Но при этом каждый конструкторский отдел получает задания только от одного аналитического отдела.

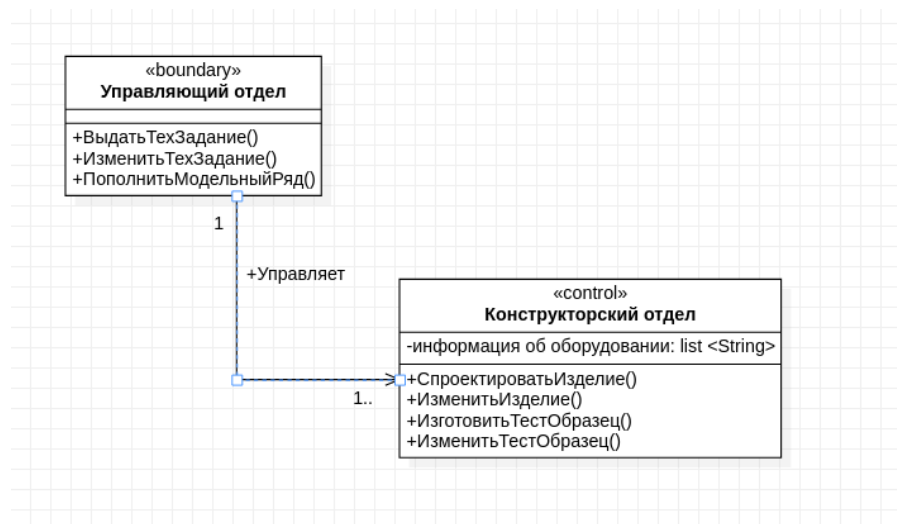


Рис. 6.15: Кратность ассоциации

## Добавление отношения обобщения на диаграмму классов и редактирование ее свойств

Добавление на диаграмму отношения обобщения между двумя классами выполняется следующим образом. На специальной панели инструментов необходимо нажать кнопку с изображением пиктограммы обобщения и отпустить левую кнопку мыши.

Далее на диаграмме классов надо выделить первый элемент обобщения или потомок,

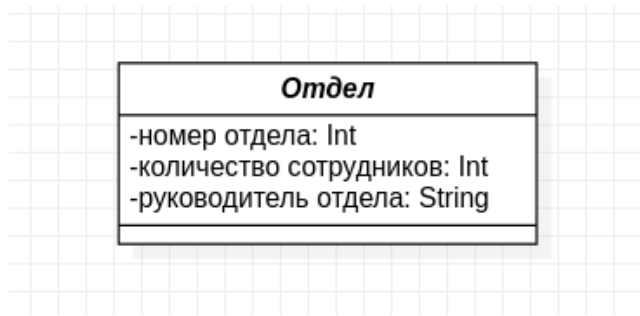


Рис. 6.16: Абстрактный класс

от которого исходит стрелка, и, не отпуская нажатую левую кнопку мыши, переместить ее указатель ко второму элементу отношения или предку, к которому направлена стрелка. После перемещения ко второму элементу кнопку мыши следует отпустить, в результате чего на диаграмму классов будет добавлена линия обобщения между двумя выбранными классами.

Продолжая разработку модели предприятия, добавим на диаграмму третий, абстрактный класс с именем *Отдел*, который будет являться родительским для всех классов, соответственно добавим на диаграмму связи обобщения (*Generalization*).

Добавим в описание класса следующие закрытые (*private*) атрибуты:

- Номер отдела: *Int*
- Количество сотрудников: *Int*
- Руководитель отдела: *String*

Последний класс может быть предназначен для спецификации системных атрибутов и операций, необходимых при исполнении соответствующей программы. Напомним, что на абстрактный характер класса указывает написание курсивом его имени, а для спецификации данного свойства класса необходимо на вкладке *Properties* окна спецификации свойств класса *Отдел* выставить отметку в строке выбора *isAbstract* (рис. 6.16).

### Добавление и редактирование атрибутов классов

Из всех графических элементов среды StarUML™ класс обладает максимальным набором свойств, главными из которых являются его атрибуты и операции, поскольку именно диаграмма классов используется для генерации программного кода. Добавить атрибут к созданному ранее классу можно одним из следующих способов:

- С помощью операции контекстного меню *Add Attribute* (Добавить атрибут) или воспользоваться сочетанием клавиш (*Ctrl+Enter*) для класса, выделенного на диаграмме классов. В этом случае активизируется курсор ввода текста в области графического изображения класса на диаграмме;
- С помощью операции контекстного меню: *Add → Attribute* для класса, нажав правой кнопкой мыши на графическое изображение класса;
- Нажав правой кнопкой мыши на названии класса в секции *Model Explorer* и выбрав пункты контекстного меню: *Add → Attribute*.

## Добавление и редактирование операций классов

Функционирование предприятия основано на выполнении его отделами тех или иных действий (процессов). В модели структуры предприятия все действия представляются с помощью операций классов. Таким образом, следующий этап разработки диаграммы классов связан со спецификацией операций классов.

Добавить операцию к созданному ранее классу можно одним из следующих способов:

- С помощью операции контекстного меню *Add Operation* (Добавить метод) или воспользоваться сочетанием клавиш (Ctrl+Sift+Enter) для класса, выделенного на диаграмме классов. В этом случае активизируется курсор ввода текста в области графического изображения класса на диаграмме;
- С помощью операции контекстного меню: *Add → Operation* для класса, нажав правой кнопкой мыши на графическое изображение класса;
- Нажав правой копкой мыши на названии класса в секции *Model Explorer* и выбрав пункты контекстного меню: *Add → Operation*.

## Окончательное построение диаграммы классов модели предприятия

Для окончательного построения диаграммы классов рассматриваемой модели предприятия следует описанным выше способом добавить оставшиеся классы и ассоциации, а также специфицировать стереотипы, атрибуты и операции этих классов.

Построенная в результате указанных действий диаграмма классов будет иметь следующий вид (рис. 6.17).

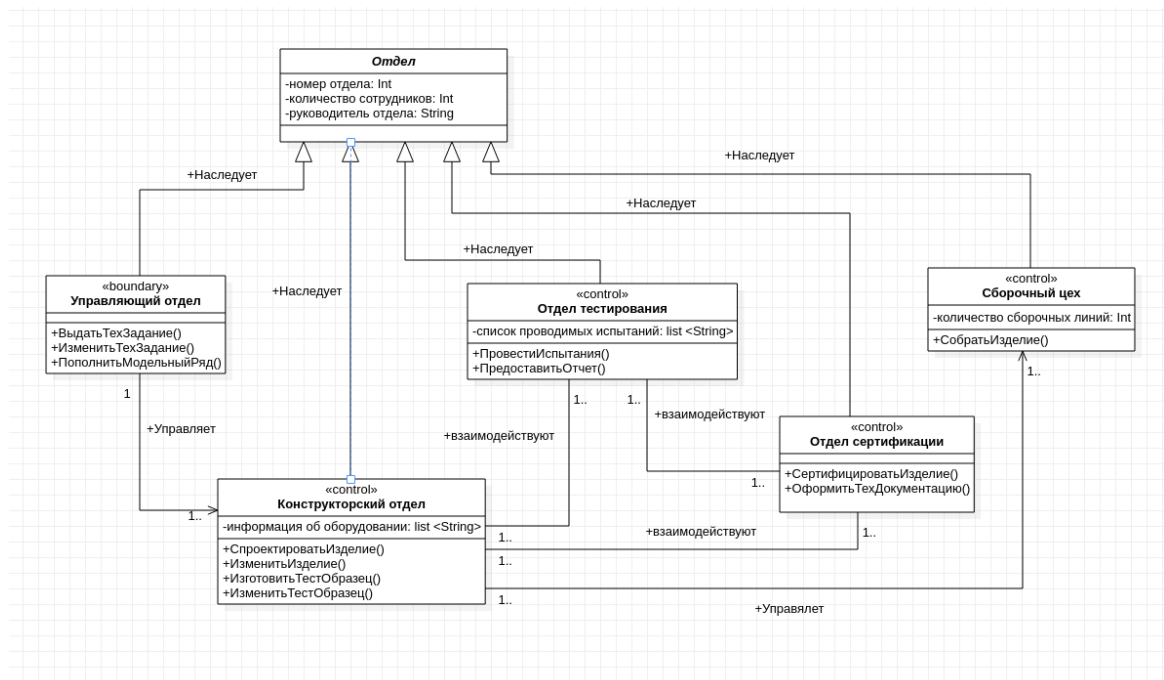


Рис. 6.17: Диаграмма Классов

Диаграмма классов является логическим представлением структуры модели, поэтому она

должна содержать столько классов, сколько необходимо для реализации всего проекта. При этом для полного представления структуры модели необходимо установить и специфицировать отношения между классами.

## 6.3 Особенности разработки диаграмм кооперации

Диаграмма кооперации является разновидностью диаграммы взаимодействия, и в контексте языка UML описывает динамический аспект взаимодействия объектов при реализации отдельных вариантов использования. Активизировать рабочее окно диаграммы кооперации в программе StarUML™ можно несколькими способами:

- выбрать соответствующий вариант, перейдя по следующим пунктам меню: *Model → AddDiagram → CommunicationDiagram*;
- нажать правую кнопку мыши на рабочей области и выбрать аналогичные пункты контекстного меню: *AddDiagram → CommunicationDiagram*;

При этом появляется новое окно с чистым рабочим листом диаграммы кооперации и специальная панель инструментов, содержащая кнопки с изображением графических примитивов, необходимых для разработки диаграммы кооперации (рис. 6.18). Назначение отдельных кнопок панели можно узнать из всплывающих подсказок.

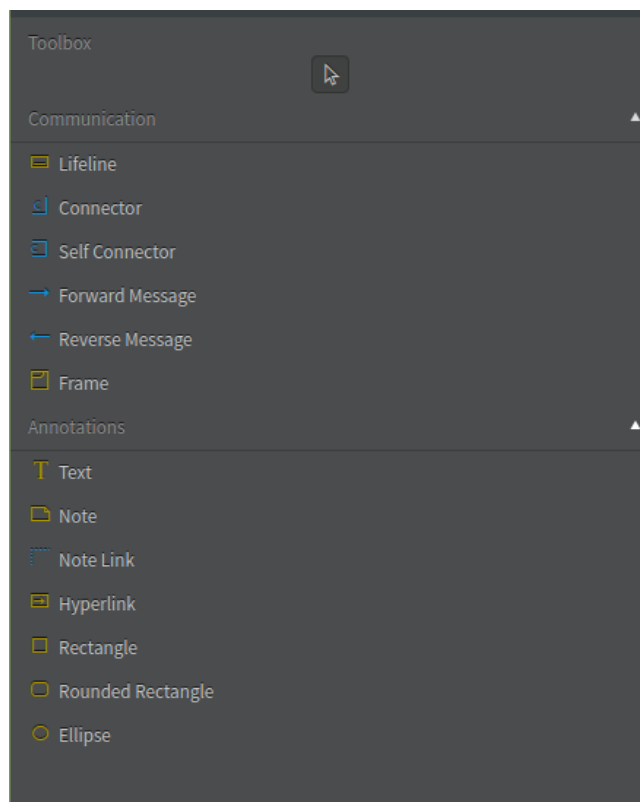


Рис. 6.18: Communication Diagram: Панель инструментов

В качестве примера рассматривается процесс построения диаграммы кооперации, которая представляет собой реализацию варианта использования «Разработки новой модели велоси-

педа» применительно к разрабатываемому проекту описания работы велосипедного предприятия. В модели данная диаграмма кооперации соответствует этому варианту использования и может быть размещена в представлении вариантов использования (Use Case View).

После активизации новой диаграммы кооперации одним из описанных выше способов следует в качестве имени данной диаграммы задать: «Разработка новой модели велосипеда».

В общем случае работа с диаграммой кооперации состоит в добавлении объектов, связей и сообщений, а также редактировании их свойств.

### Добавление объекта на диаграмму кооперации и редактирование его свойств

Добавить объект на диаграмму кооперации можно стандартным образом с помощью соответствующей кнопки на специальной панели инструментов. Однако, в случае наличия построенной ранее диаграммы классов, более удобным представляется следующий способ.

В обозревателе проекта выделить необходимый класс и, удерживая нажатой левую кнопку мыши, перетащить изображение пиктограммы класса из браузера на свободное место рабочего листа диаграммы кооперации. В результате этих действий на диаграмме кооперации появится изображение объекта с именем класса и маркерами изменения его геометрических размеров (рис. 6.19).

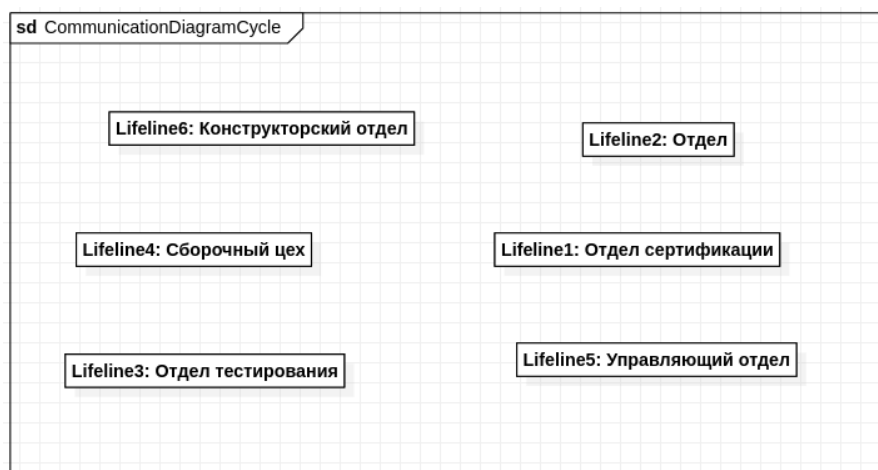


Рис. 6.19: Объекты диаграммы кооперации

По умолчанию каждый добавляемый объект считается анонимным. При необходимости можно задать собственное имя объекта, для чего двойным щелчком на изображении объекта на диаграмме кооперации следует вызвать контекстное меню свойств этого объекта (рис. 6.20).

Для объекта выбранного класса можно задавать: собственное имя объекта, особенности его реализации и множественность экземпляров. При необходимости можно представить объект в форме мультиобъекта. Для этого следует поставить галочку у свойства *isMultinstance* (Несколько экземпляров). Однако для объекта класса *Управляющий отдел* и это свойство следует оставить пустым, поскольку данный объект присутствует в модели в единственном экземпляре.



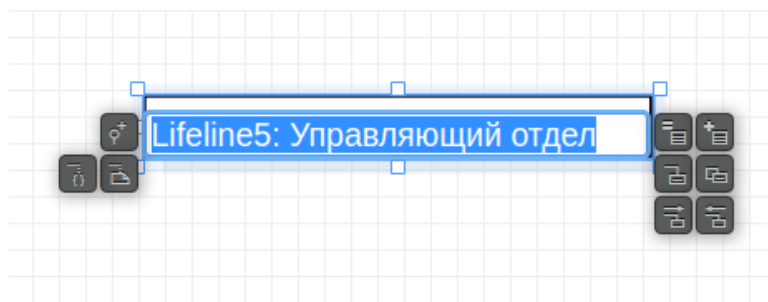


Рис. 6.20: Контекстное меню

### Добавление сообщения и редактирование его свойств

Для добавления сообщения между предварительно размещенными на диаграмме объектами нужно с помощью левой кнопки мыши нажать кнопку с изображением связи на специальной панели инструментов, отпустить левую кнопку мыши, щелкнуть левой кнопкой мыши на изображении одного объекта на диаграмме и отпустить ее на изображении другого объекта.

В результате этих действий на диаграмме появится изображение связи, например, соединяющей объект класса *Управляющий Отдел* с объектом класса *Конструкторский отдел* (рис. 6.21). После этого следует выполнить операцию *Select Operation* контекстного меню сообщения, в результате чего появляется вложенный список с предложением выбрать одну из операций целевого класса для спецификации имени сообщения (рис. 6.22).



Рис. 6.21: Изображение связи

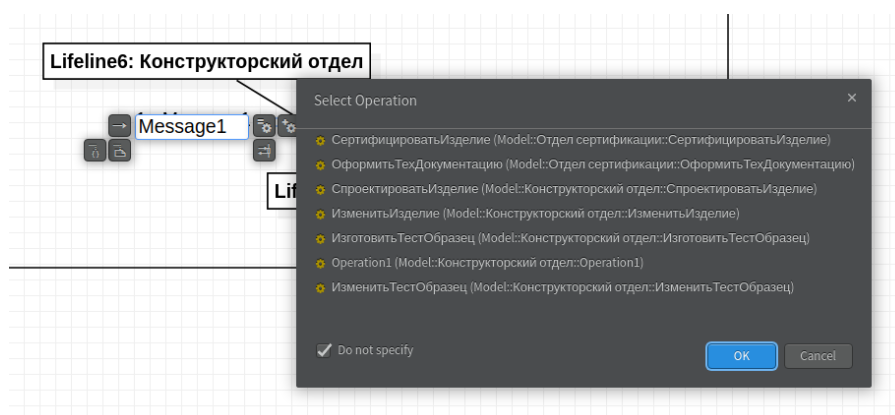


Рис. 6.22: Выбор сообщения (метода)

Для рассматриваемой модели предприятия для первого сообщения следует выбрать опе-

рацию *ОформитьТехДокументацию()*. После выбора операции для данного сообщения оно добавляется в список сообщений данной связи, а рядом с линией связи на диаграмме кооперации появится стрелка с номером и именем этого сообщения (рис. 6.23).

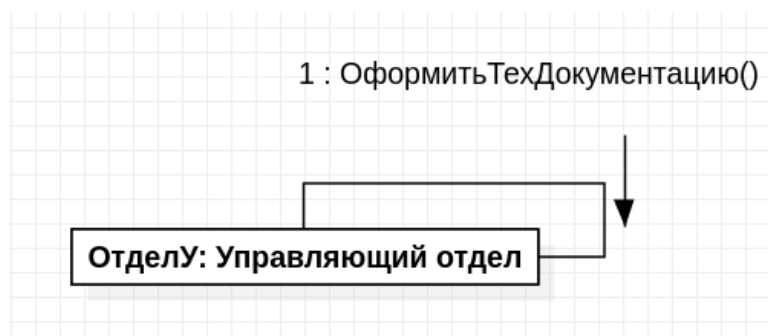


Рис. 6.23: Рефлексивное сообщение

Характеристика отдельных свойств синхронизации сообщений и графическое изображение соответствующих стрелок сообщений приводится в следующей таблице (табл. 6.3).

Название свойства	Назначение свойства
Simple (Простое)	Данное сообщение выполняется в одном потоке управления. Это свойство задается добавляемому на диаграмму сообщению по умолчанию
Synchronous (Синхронное)	После передачи данного сообщения клиент ожидает ответа от объекта-приемника о результате выполнения соответствующей операции
Balking (С отказом)	После передачи данного сообщения объект-приемник отказывает клиенту в выполнении соответствующей операции, если он занят выполнением других операций
Timeout (С ожиданием)	После передачи данного сообщения объект-приемник может поместить данное сообщение в очередь с ограниченным временем ожидания, если он занят выполнением других операций
Procedure Call (Вызов процедуры)	Клиент посылает данное сообщение объекту-приемнику и, чтобы продолжить свою работу ожидает, пока вся дальнейшая вложенная последовательность сообщений не будет обработана приемником
Asynchronous (Асинхронное)	Клиент посылает данное сообщение и продолжает свою работу, не ожидая подтверждения от объекта-приемника о получении этого сообщения. При этом соответствующая операция может быть как выполнена, так и не выполнена
Return (Возврат)	Данное сообщение посылается клиенту после окончания выполнения вызова процедуры

Таблица 6.3: Свойства сообщений

## Окончательное построение диаграммы последовательности модели предприятия по производству велосипедов

Для завершения построения диаграммы последовательности рассматриваемого примера следует описанным выше способом добавить оставшиеся объекты и сообщения. С этой целью следует выполнить следующие действия:

1. Добавить объекты классов *Отдел сертификации*, *Сборочный цех*, *Отдел тестирования*;
2. Указать объекты всех классов как мультиобъекты, выбрав соответствующий параметр в секции *Editors*;
3. Добавить следующие сообщения между классами:
  - ОформитьТехЗадание()
  - СпроектироватьИзделие()
  - ИзготовитьТестОбразец()
  - ПровестиИспытания()
  - ПредоставитьОтчет()
  - СертифицироватьИзделие()
  - ОформитьТехДокументацию()
  - СобратьИзделие()
  - ПополнитьМодельныйРяд()
4. Указать стереотипы для классов. В случае данного варианта использования класс *Управляющий отдел* будет иметь стереотип «*border*», так как он является точкой инициализации и точкой окончания процесса. Все остальные классы составляют внутреннюю систему и имеют стереотип «*control*».

При необходимости можно изменить порядок следования сообщений и их спецификацию, а также установить дополнительную синхронизацию сообщений и связать с сообщениями примечания, следует обратиться к Обозревателю Модели (*Model Explorer*), в котором можно изменить порядок следования сообщений, выбрав пункт (*Move Up/Move Down*) или воспользоваться сочетанием клавиш **Ctrl+Shift+Стрелка вверх**/**Ctrl+Shift+Стрелка вниз**.

Диаграмма кооперации, описывающая реализацию типичного хода событий варианта использования *Разработка новой модели велосипеда* показана на рис. 6.24.

## 6.4 Особенности разработки диаграммы последовательности

Диаграмма последовательности является другой формой визуализации взаимодействия в модели и, как и диаграмма кооперации, оперирует объектами и сообщениями.

Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни» (англ. *lifeline*), отображающие течение времени, прямоугольники, отражающие деятельность объекта или испол-

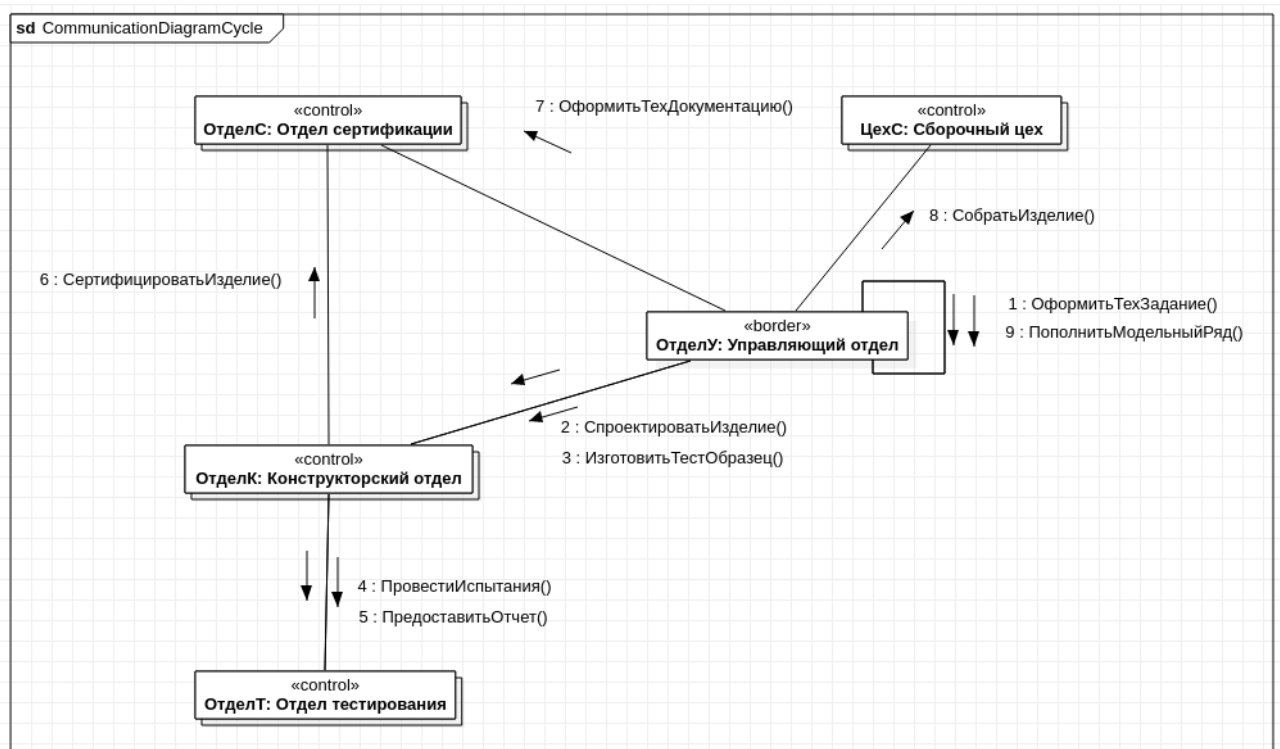


Рис. 6.24: Диаграмма кооперации

нение им определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами.

Активизировать рабочее окно диаграммы последовательности в среде разработки StarUML™ можно несколькими способами:

- выбрать соответствующий вариант, перейдя по следующим пунктам меню:  
*Model* → *AddDiagram* → *SequenceDiagram*;
- нажать правую кнопку мыши на рабочей области и выбрать аналогичные пункты контекстного меню: *AddDiagram* → *SequenceDiagram*;

При этом появляется новое окно с чистым рабочим листом диаграммы классов и специальная панель инструментов, содержащая кнопки с изображением графических примитивов, необходимых для разработки диаграммы последовательности (табл. 6.4). Назначение отдельных кнопок панели можно узнать из всплывающих подсказок.

### Добавление объекта на диаграмму последовательности и редактирование его свойств

Добавить объект на диаграмму последовательности можно как стандартным образом с помощью соответствующей кнопки на специальной панели инструментов, так и более удобным способом — с помощью перетаскивания изображения пиктограммы класса из браузера на свободное место рабочего листа диаграммы последовательности.

В результате этих действий на диаграмме последовательности появится изображение объекта с именем класса, маркерами изменения его геометрических размеров и вертикальной пунктирной линией, означающей линию жизни этого объекта (рис. 6.25).

Всплывающая подсказка	Назначение кнопки
Selection Tool	Превращает изображение курсора в форму стрелки для последующего выделения элементов на диаграмме
Text Box	Добавляет на диаграмму текстовую область
Note	Добавляет на диаграмму примечание
Anchor Note to Item	Добавляет на диаграмму связь примечания с соответствующим графическим элементом диаграммы
Object	Добавляет на диаграмму объект
Object Message	Добавляет на диаграмму простое сообщение
Message To Self	Добавляет на диаграмму рефлексивное сообщение
Return Message	Добавляет на диаграмму сообщение типа возврата из вызова процедуры
Destruction Marker	Добавляет на диаграмму символ уничтожения объекта
Procedure Call	Добавляет на диаграмму сообщение типа вызова процедуры
Asynchronous Message	Добавляет на диаграмму асинхронное сообщение

Таблица 6.4: Диаграмма последовательности: Панель инструментов

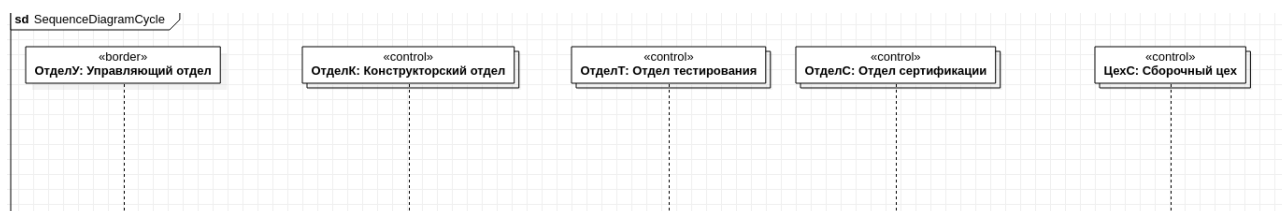


Рис. 6.25: Объекты диаграммы последовательности

Так же как и для диаграммы кооперации, для диаграммы последовательности каждый добавляемый объект по умолчанию считается анонимным. При необходимости можно задать собственное имя объекта, для чего уже известным способом (например, двойным щелчком на изображении объекта на диаграмме) следует вызвать диалоговое окно свойств объекта или обратиться к секции *Editors*.

При необходимости можно представить объект в форме мультиобъекта. Для этого следует выбрать отметку у свойства `isMultinstance` (Несколько экземпляров), аналогично тому, как это было достигнуто при проектировании **диаграммы кооперации**.

### Добавление связи и редактирование ее свойств

Для добавления рефлексивной связи (отправки рефлексивного сообщения) следует выбрать соответствующий инструмент (*Self Message*) на панели и нажать левой кнопкой мыши на объект класса в рабочей области диаграммы последовательности. Далее следует внести название связи в предлагаемое поле. В конкретном примере показан процесс создания рефлексивной связи класса *Управляющий отдел* (рис. 6.26).

Для добавления связи между предварительно размещенными на диаграмме объектами нужно с помощью левой кнопки мыши нажать кнопку с изображением связи на специальной панели инструментов, отпустить левую кнопку мыши, щелкнуть левой кнопкой мыши на

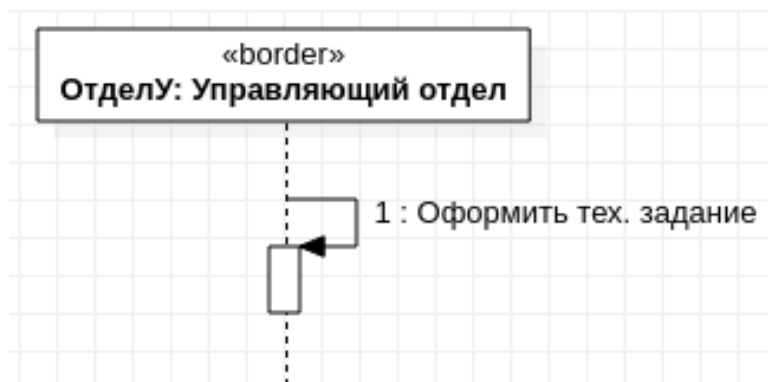


Рис. 6.26: Рефлексивная связь

изображении одного объекта на диаграмме и отпустить ее на изображении другого объекта.

В результате этих действий на диаграмме появится изображение связи, например, соединяющей объект класса *Управляющий отдел* с объектом класса *Конструкторский отдел* (рис. 6.27). При этом изображение линии жизни у соответствующей пары объектов изменится на изображение фокуса управления.



Рис. 6.27: Отображение связи между объектами

После добавления имени связи можно перейти к дополнительным параметрам. Например, задать имя ассоциации, видимость соответствующей пары объектов и наличие общих ролей. Для редактирования свойств связи следует обратиться к вспомогательной секции *Editors*, расположенной справа от рабочей области (рис. 6.28).

Построение диаграммы последовательности сводится к добавлению и редактированию свойств отдельных объектов и сообщений. Доступ к окну спецификации свойств соответствующих элементов возможен также либо через контекстное меню, либо с помощью секции меню *Editors*. При добавлении сообщений на диаграмму последовательности они получают по умолчанию свой номер в общей последовательности сообщений.

Для изменении порядка следования сообщений следует обратиться к Обозревателю Модели (*Model Explorer*), выбрав пункт (*Move Up/Move Down*) или воспользоваться сочетанием клавиш **Ctrl+Shift+Стрелка вверх**/**Ctrl+Shift+Стрелка вниз**. При этом список сообщений автоматически перестроен. См. рис. 6.29.

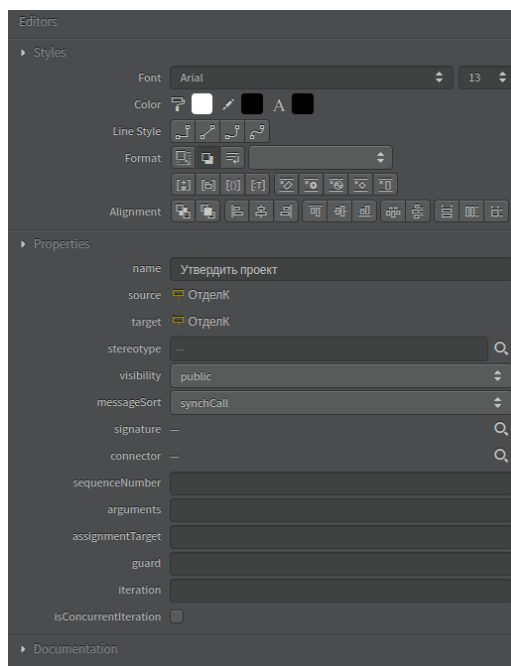


Рис. 6.28: Свойства связи

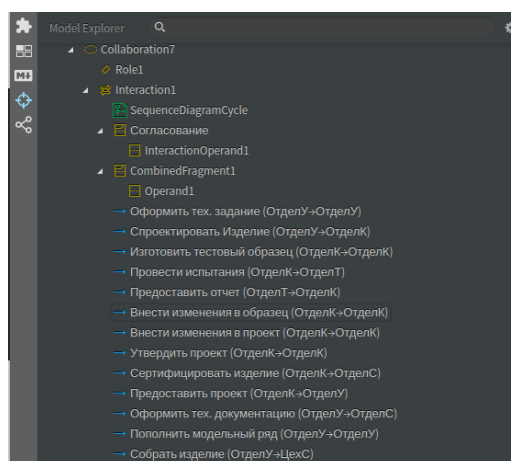


Рис. 6.29: Меню обозревателя модели

## Управление сложными взаимодействиями с фрагментами последовательности

Диаграмма последовательности в нотации UML предусматривает добавление сложных фрагментов для реализации таких сценариев использования как циклическое действие, альтернативное действие, справочный сегмент. StarUML™ также располагает средствами для использования данных фрагментов.

Рассмотрим правила использования данных фрагментов на примере системы «Банкомат».

Фрагмент последовательности представлен в виде рамки, обрамляющей участок взаимодействия между объектами (как показано в примерах ниже) на диаграмме последовательности.

Он используется для более структурированного отображения сложных взаимодействий, таких как альтернативные потоки и циклы. В верхнем левом углу фрагмента расположен оператор. Этот оператор (оператор фрагмента) указывает на вид фрагмента.

**Альтернативы** Фрагмент альтернативной комбинации используется, когда необходимо сделать выбор между двумя или более последовательностями сообщений. Он моделирует логику «если, то еще».

Альтернативный фрагмент представляет собой большой прямоугольник или кадр, который задается упоминанием 'alt' в окошке с названием кадра (так называемый оператор фрагмента).

Чтобы показать две или более альтернативы, больший прямоугольник затем делится на то, что называется операндов взаимодействия, используя пунктирную линию, как показано на примере диаграммы последовательности выше. Каждый операнд имеет защитное ограждение, которое проверяется и размещается в верхнем левом углу операндов.

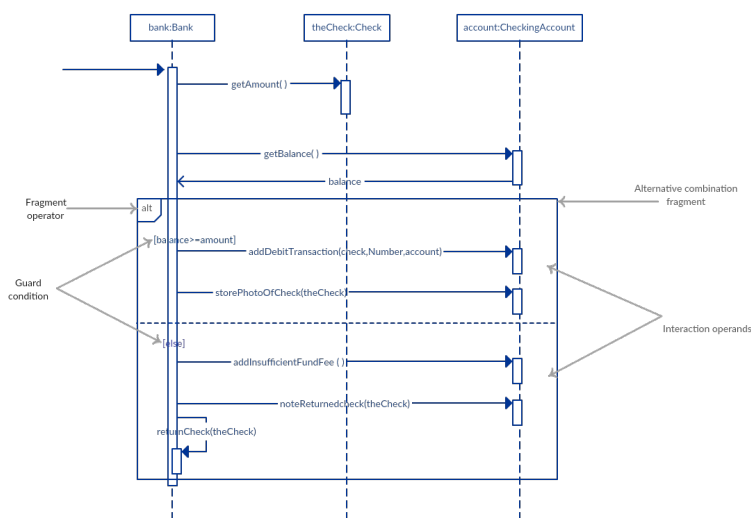


Рис. 6.30: Пример альтернативного фрагмента

**Варианты** Фрагмент комбинации опций используется для указания последовательности, которая будет происходить только при определенном условии, в противном случае последовательность не будет происходить. Оно моделирует утверждение “если тогда”.

Как и альтернативный фрагмент, фрагмент опции также представлен прямоугольной рамкой, в которой в окошке с названием помещается 'opt'.

В отличие от альтернативного фрагмента, фрагмент опции не делится на два или более операндов. Вариант защиты расположен в верхнем левом углу.

**Loops** Loop фрагмент используется для представления повторяющейся последовательности.

В квадратных скобках секции Loop можно разместить условия выхода из цикла или защитные условия (*Guard Condition*).

Если это защита от минимальных итераций, то шлейф должен выполняться не меньше указанного числа, а если это защита от максимальных итераций, то шлейф не должен выполняться больше указанного числа.

(Пример фрагмента цикла приведен ниже в шаблонах последовательных диаграмм и разделе с примерами)



**Справочный фрагмент** Вы можете использовать фрагмент ссылки для управления размером больших последовательных диаграмм. Это позволяет повторно использовать часть одной диаграммы последовательностей в другой, или, другими словами, можно ссылаться на часть диаграммы в другой диаграмме, используя фрагмент ref-фрагмента.

Для указания ссылочного фрагмента необходимо в окошке с названием кадра указать 'ref', а внутри кадра — название диаграммы последовательности, на которую делается ссылка.

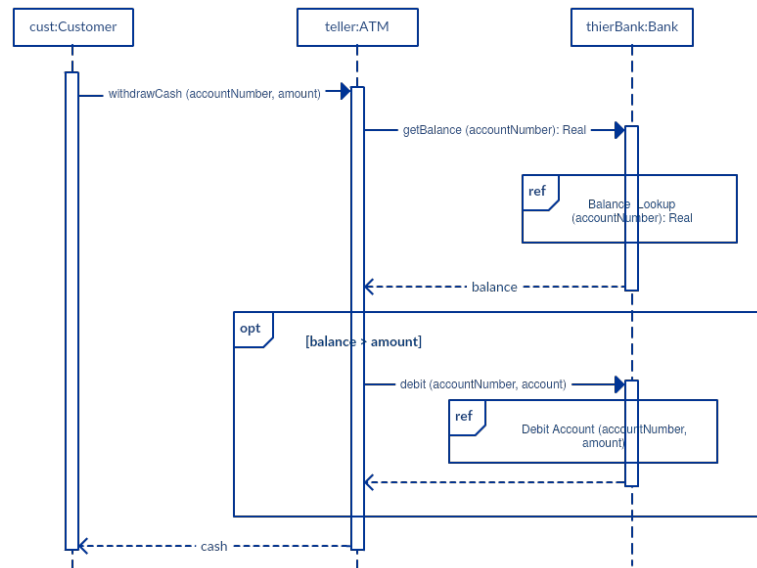


Рис. 6.31: Пример справочного и вариантного фрагмента

### Добавление фрагментов последовательности

При разработке рассматриваемой диаграммы последовательности, описывающей вариант использования «Разработка новой модели велосипеда» мы также использовали фрагменты для отражения альтернативного выбора и циклически выполняемых действий.

Для добавления такого фрагмента следует выбрать инструмент *Combined Fragment* на панели и разместить его на рабочей области. Изменить тип фрагмента можно в секции *Editors*, выбрав один из вариантов, предлагаемых в выпадающем списке. См. рис. 6.32.

### Окончательное построение диаграммы последовательности модели предприятия

Для завершения построения диаграммы последовательности рассматриваемого примера следует описанным выше способом добавить оставшиеся объекты и сообщения, которые упоминались в диаграмме кооперации.

Фрагмент диаграммы последовательности, описывающая реализацию типичного хода событий варианта использования «Разработка новой модели велосипеда» для модели предприятия, показан на рис. 6.33.

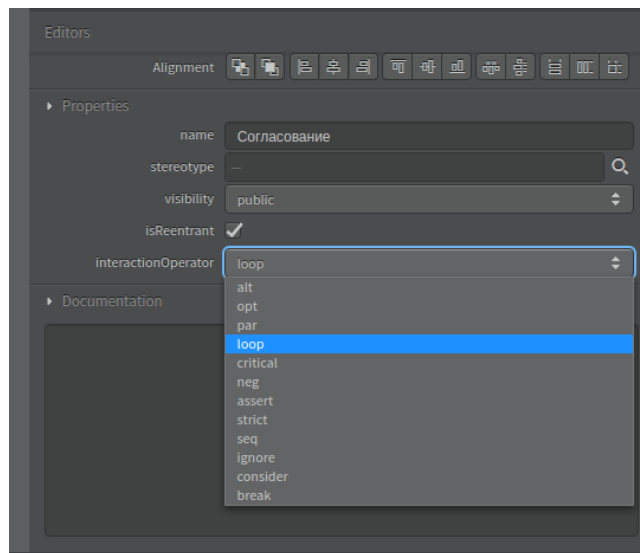


Рис. 6.32: Настройки фрагмента

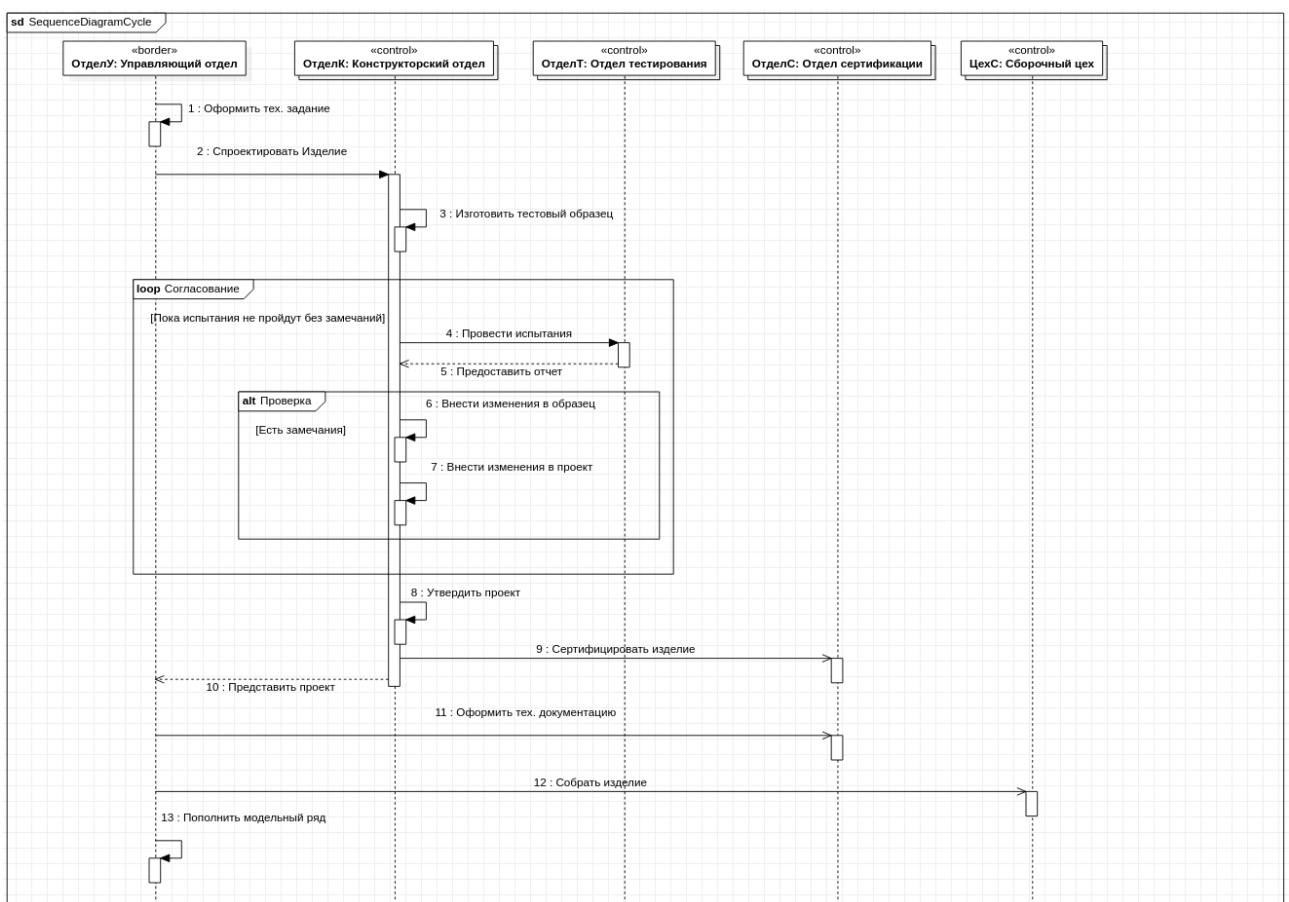


Рис. 6.33: Диаграмма последовательности

## 6.5 Особенности разработки диаграммы состояний

Диаграмма состояний — это, по существу, диаграмма состояний из теории автоматов со стандартизированными условными обозначениями, которая может определять множество систем от компьютерных программ до бизнес-процессов. Используются следующие условные обозначения:

- Круг, обозначающий начальное состояние;
- Окружность с маленьким кругом внутри, обозначающая конечное состояние (если есть);
- Скруглённый прямоугольник, обозначающий состояние. Верхушка прямоугольника содержит название состояния. В середине может быть горизонтальная линия, под которой записываются активности, происходящие в данном состоянии;
- Стрелка, обозначающая переход. Название события (если есть), вызывающего переход, отмечается рядом со стрелкой. Охраняющее выражение может быть добавлено перед «/» и заключено в квадратные скобки, что значит, что это выражение должно быть истинным, чтобы переход имел место. Если при переходе производится какое-то действие, то оно добавляется после «/» (название\_события[охраняющее\_выражение]/действие);
- Толстая горизонтальная линия с либо множеством входящих линий и одной выходящей, либо одной входящей линией и множеством выходящих. Это обозначает объединение и разветвление соответственно.

Начать построение диаграммы состояний для выбранного элемента модели или моделируемой системы в целом можно одним из следующих способов:

- выбрать соответствующий вариант, перейдя по следующим пунктам меню:  
*Model → AddDiagram → StatechartDiagram;*
- нажать правую кнопку мыши на рабочей области и выбрать аналогичные пункты контекстного меню: *AddDiagram → StatechartDiagram;*

Продолжая разработку проекта по моделированию предприятия по производству велосипедов, можно приступить к разработке новой диаграммы состояний. С этой целью для диаграммы состояний модели предприятия зададим имя *StatechartDiagramCycle*.

В результате выполнения этих действий появляется новое окно с чистым рабочим листом диаграммы состояний и специальная панель инструментов, содержащая кнопки с изображением графических элементов модели, необходимых для разработки диаграммы состояний. Приведем назначение основных инструментов в таблице 6.5. Назначение отдельных кнопок панели можно узнать из всплывающих подсказок.

Всплывающая подсказка	Назначение кнопки
Text Box	Добавляет на диаграмму текстовую область
Note	Добавляет на диаграмму примечание
Simple State	Добавляет на диаграмму состояние
Initial State	Добавляет на диаграмму начальное состояние
Final State	Добавляет на диаграмму конечное состояние
Transition	Добавляет на диаграмму переход
Self Transition	Добавляет на диаграмму рефлексивный переход
Join/Fork	Добавляет на диаграмму символ для описания параллельного перехода
Choice	Добавляет на диаграмму символ принятия решения для альтернативных переходов

Таблица 6.5: Диаграмма состояний: Назначения инструментов

Полный список используемых инструментов можно увидеть на рисунке 6.34.

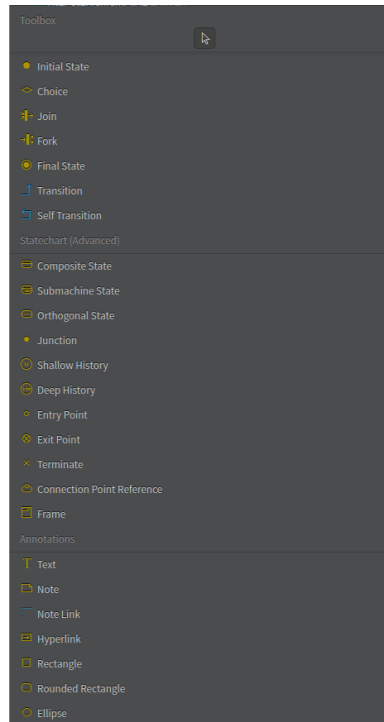


Рис. 6.34: Диаграмма состояний: Панель инструментов

### Добавление состояния на диаграмму состояний и редактирование его свойств

Для добавления состояния на диаграмму состояний необходимо с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы состояния на специальной панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы. На диаграмме появится изображение состояния с маркерами изменения его геометрических размеров и предложенным средой именем по умолчанию, которое разработчику следует изменить.

Для диаграммы состояний модели предприятия в качестве имени первого добавленного состояния изменим предложенное программой по умолчанию имя *State1* на *Разработка Технического задания* (рис 6.35). Задать имя элемента можно либо непосредственно при добавлении нового состояния на диаграмму состояний, либо открыв окно спецификации свойств нового состояния.

### Добавление перехода и редактирование его свойств

Для добавления перехода между двумя состояниями нужно с помощью левой кнопки мыши нажать кнопку с изображением перехода на специальной панели инструментов, отпустить левую кнопку мыши, щелкнуть левой кнопкой мыши на изображении исходного состояния на диаграмме и отпустить ее на изображении целевого состояния.

В результате этих действий на диаграмме появится изображение перехода, соединяющего два выбранных состояния. Продолжая разработку модели предприятия, добавим на диаграм-

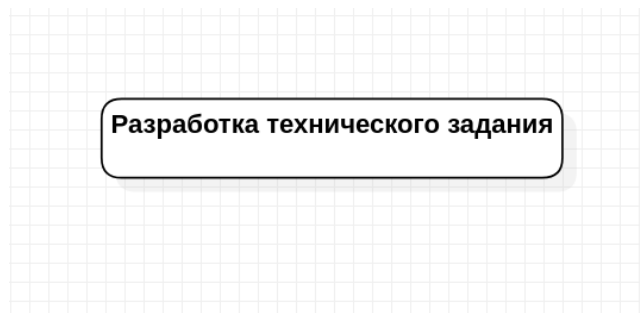


Рис. 6.35: Добавление состояния на диаграмму

му состояний начальное состояние (*Initial State*) и соединим его переходом с состоянием *Разработка Технического задания* (рис. 6.36). Для добавления переходу имени следует дважды кликнуть левой кнопкой мыши на изображении стрелки и ввести условия, названия, охранные выражения присущие данному переходу.



Рис. 6.36: Добавление перехода на диаграмму

Состояние может содержать только имя или имя и дополнительно список внутренних действий. Список внутренних действий содержит перечень действий или деятельности, которые выполняются во время нахождения объекта в данном состоянии. Данный список фиксированный. Список основных действий включает следующие значения:

- **entry** — действие, которое выполняется в момент входа в данное состояние (входное действие);
- **exit** — действие, которое выполняется в момент выхода из данного состояния (выходное действие);
- **do** — выполняющаяся деятельность («do activity») в течение всего времени, пока объект находится в данном состоянии
- **defer** — событие, обработка которого предписывается в другом состоянии, но после того, как все операции в текущем будут завершены.

### Узлы ветвления и объединения

Узлы ветвления и объединения аналогичны узлам на диаграмме деятельности. На диаграмме состояний обычно данные подсостояния используются для распараллеливания переходов в композитных состояниях, о которых речь пойдет позже. После срабатывания перехода моделируемый объект одновременно будет находиться во всех целевых состояниях этого перехода.

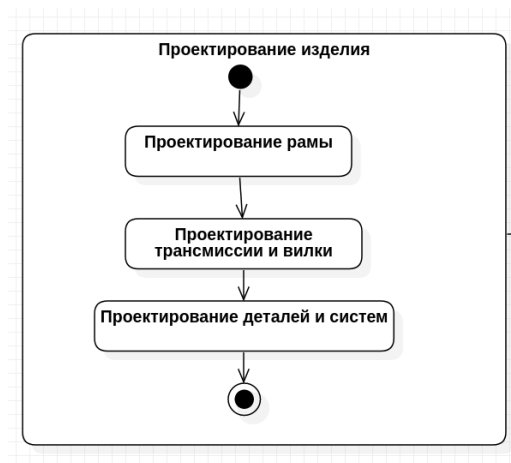


Рис. 6.37: Составное состояние

Варианты принятия решений на диаграмме состояний могут быть показаны также как и на диаграмме деятельности с помощью узла выбора. При этом переход в состояние выбора должен быть триггерным и содержать имя события. Переходы из псевдосостояния выбора в целевые состояния должны содержать сторожевые условия. Переход, который должен срабатывать, если ни одно из условий не примет значение «истина» должен содержать метку «else».

В дальнейшем мы используем узел ветвления при построении диаграммы состояний для модели рассматриваемого предприятия по производству велосипедов.

### Сложные и простые состояния

На диаграмме могут быть представлены как простые состояния, так и сложные состояния. Сложные или составные состояния (*composite state*) включают в себя вложенные подсостояния. Декомпозиция сложного состояния может осуществляться как на основной диаграмме, так и отдельно. Приведем пример сложного состояния, описанного при построении модели рассматриваемого предприятия (рис. 6.37).

### Последовательные и параллельные состояния

**Последовательные подсостояния/состояния** (*sequential substates*) используются для моделирования такого поведения объекта, во время которого в каждый момент времени объект может находиться в одном и только одном подсостоянии. Поведение объекта в этом случае представляет собой последовательную смену подсостояний, начиная от начального и заканчивая конечным подсостояниями.

**Параллельные подсостояния/состояния** (*concurrent substates*) позволяют специфицировать два и более подавтомата, которые могут выполняться параллельно внутри составного события. Каждый из подавтоматов занимает некоторую область (регион) внутри составного состояния.

Переходы могут осуществляться как в само композитное состояние, так и в одно из его

подсостояний. Таким образом, переход, стрелка которого соединена с границей некоторого составного состояния, обозначает переход в составное состояние. Он эквивалентен переходу в начальное состояние каждого из подавтоматов.

Переход, выходящий из составного состояния относится к каждому из вложенных под-состояний. Это означает, что объект может покинуть составное суперсостояние, находясь в любом из его подсостояний. Если необходимо указать конкретное подсостояние из которого может осуществиться выход из композитного состояния, достаточно добавить переход от подсостояния в целевое состояние.

Для данной системы состояния «Изготовление вилки» и «Изготовление рамы» могут быть рассмотрены как те, в которых система может находиться одновременно. С этой целью, используя инструменты *Fork* и *Join* построим узел параллельного перехода. См. рис. 6.38.

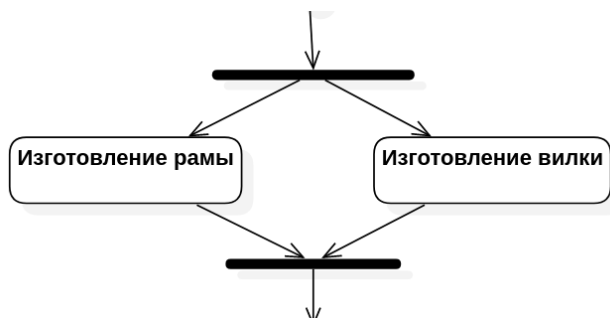


Рис. 6.38: параллельные состояния

### Окончательное построение диаграммы состояний модели предприятия

Для завершения построения диаграммы состояний рассматриваемого примера следует, основываясь на построенных диаграммах последовательности и кооперации, описанным выше способом добавить оставшиеся состояния и описать переходы между ними.

Диаграмма состояний для рассматриваемой модели предприятия будет иметь следующий вид (рис. 6.39).

## 6.6 Особенности разработки диаграммы деятельности

Диаграмма деятельности в среде StarUML™, так же как и диаграмма состояний, может относиться к отдельному классу, операции класса, варианту использования, пакету или представлению. Начать построение диаграммы деятельности для выбранного элемента модели или моделируемой системы в целом можно одним из следующих способов:

- выбрать соответствующий вариант, перейдя по следующим пунктам меню:  
*Model* → *AddDiagram* → *ActivityDiagram*;
- нажать правую кнопку мыши на рабочей области и выбрать аналогичные пункты контекстного меню: *AddDiagram* → *ActivityDiagram*;

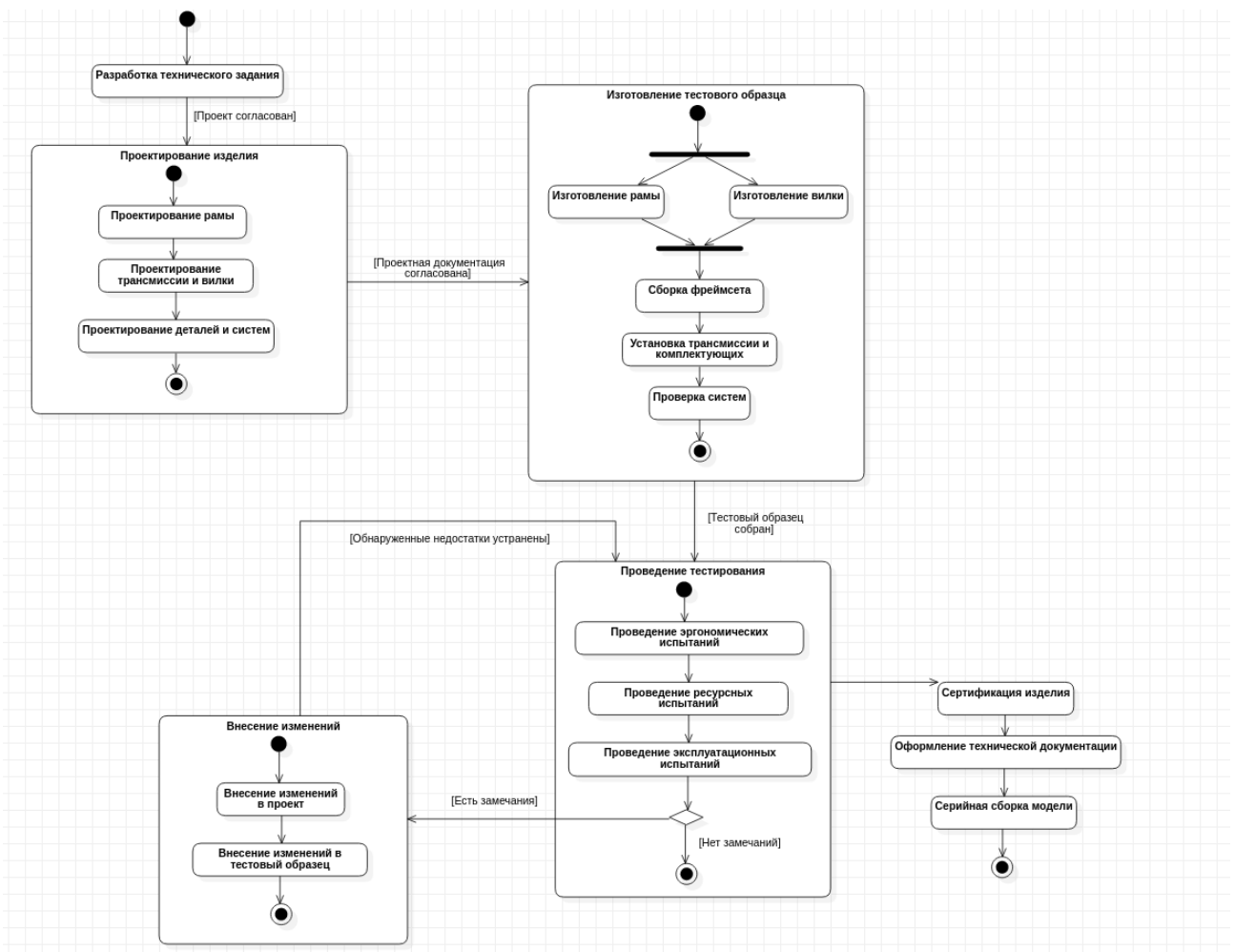


Рис. 6.39: Диаграмма состояний

В результате выполнения этих действий появляется новое окно с чистым рабочим листом диаграммы деятельности и специальная панель инструментов, содержащая кнопки с изображением графических элементов, необходимых для разработки диаграммы деятельности (табл. 6.6). Назначение отдельных кнопок панели можно узнать из всплывающих подсказок.

Более полный список предоставляемых инструментов можно видеть на рисунке 6.40.

Следует отметить, что в проектах реинжиниринга и документирования бизнес-процессов диаграмма деятельности является основным средством визуализации бизнес-процессов в контексте языка UML.

Так как изучаемый процесс по производству новой модели велосипеда на производстве может быть представлен как бизнес-процесс, то мы можем рассмотреть особенности разработки диаграмм деятельности в среде StarUML™ на данном примере.

Повторимся, что наиболее подходящим типом диаграмм для визуального представления схем выполнения бизнес-процессов являются диаграммы деятельности, на которых дополнительно размещаются так называемые дорожки (*Swimlane*). Назначение **дорожек** состоит в том, чтобы указать зоны ответственности за выполнения отдельных деятельностей в рамках моделируемого бизнес-процесса. В качестве имен дорожек используются либо названия подразделений (департаментов) рассматриваемой компании, либо названия отдельных долж-



Всплывающая подсказка	Назначение кнопки
Selection Tool	Превращает изображение курсора в форму стрелки для последующего выделения элементов на диаграмме
Text Box	Добавляет на диаграмму текстовую область
Note	Добавляет на диаграмму примечание
Note Link	Добавляет на диаграмму связь примечания с соответствующим графическим элементом диаграммы
Action	Добавляет на диаграмму состояние
Initial	Добавляет на диаграмму начальное состояние
Final	Добавляет на диаграмму конечное состояние
Join/Fork	Добавляет на диаграмму символ для описания параллельного процесса
Merge/Decison	Добавляет на диаграмму символ принятия решения для альтернативных переходов

Таблица 6.6: Диаграмма активности: Назначение основных инструментов

ностей сотрудников тех или иных подразделений.

Проекты по моделированию бизнес-процессов могут выполняться либо с целью реорганизации или реинжиниринга компании, либо с целью документирования бизнес-процессов. Особенности данных проектов заключаются в том, что в обоих случаях необходимо построить модели бизнес-процессов некоторой существующей компании. Чтобы акцентировать внимание на подобных проектах, их часто называют проектами типа *«As is»* (*«Как есть»*). Соответственно проекты по разработке новых продуктов или моделей новых систем называют проектами типа *«To be»* (*«Как должно быть»*).

Выполнение проектов типа *«Как есть»* по моделированию бизнес-процессов в большинстве случаев начинают с построения диаграмм деятельности, которые служат для графического представления схем выполнения бизнес-процессов и документооборота рассматриваемой компании. После этого, исходя из требований проекта, разрабатывается модель диаграммы вариантов использования и выполняется реорганизация бизнес-процессов. Наконец, в случае необходимости разработки или внедрения корпоративной информационной системы, строятся диаграмма классов, диаграммы взаимодействия и компонентов, которые служат основой для программной реализации соответствующего проекта.

Таким образом, первый этап выполнения проектов типа *«Как есть»* связан с построением моделей существующих бизнес-процессов компании в форме диаграмм деятельности. В качестве примера проекта этого типа рассматривается модель бизнес-процесса по разработке новой серийной модели велосипеда на предприятии. Хотя данный пример имеет упрощенный характер, он позволяет наглядно представить основные особенности моделирования бизнес-процессов в нотации языка UML с использованием средства StarUML™.

### Добавление деятельности на диаграмму деятельности и редактирование ее свойств

Для добавления деятельности на диаграмму деятельности нужно с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы деятельности (*Action*) на специальной

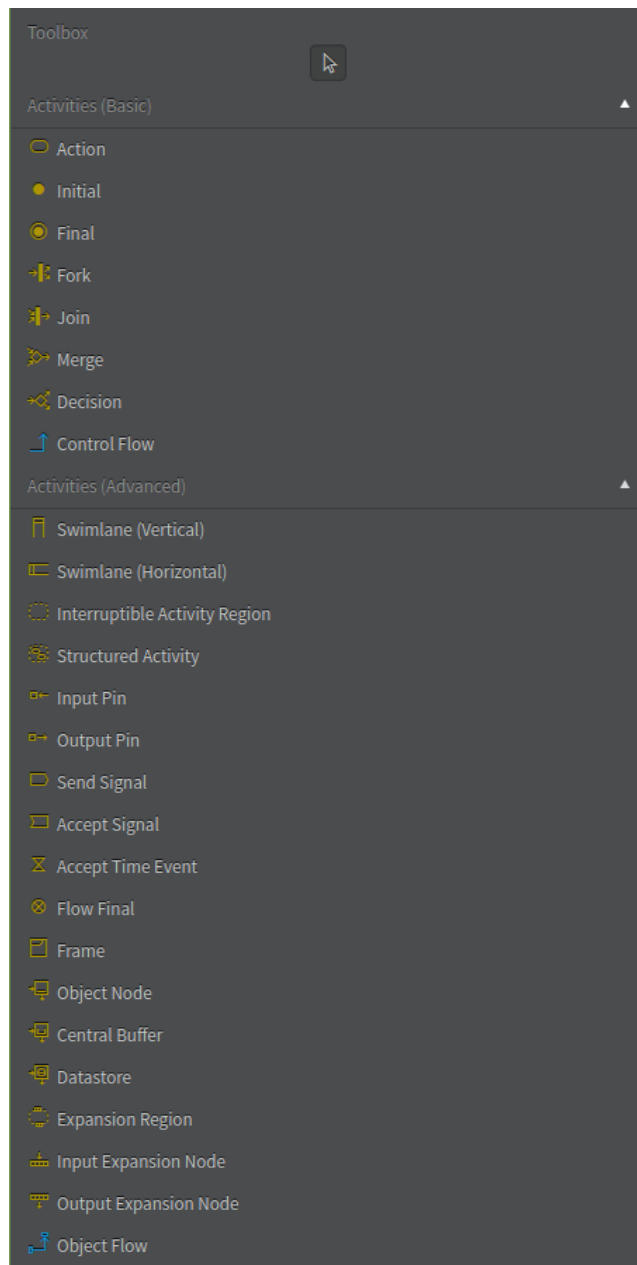


Рис. 6.40: Диаграмма Активности: Панель инструментов

панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы.

На диаграмме появится изображение деятельности с маркерами изменения его геометрических размеров и предложенным средой именем по умолчанию, которое разработчику следует изменить. В результате этих действий на диаграмме появится изображение деятельности с именем *Action1*, предложенное программой по умолчанию. Начиная построение диаграммы деятельности модели предприятия, для первой добавленной деятельности зададим имя *Оформить тех. задание* (рис. 6.41).

После добавления деятельности на диаграмму можно открыть меню *Editors*, где определить дополнительные свойства, доступные на соответствующих вкладках (рис. 6.42).



Рис. 6.41: Диаграмма активности. Добавление активности

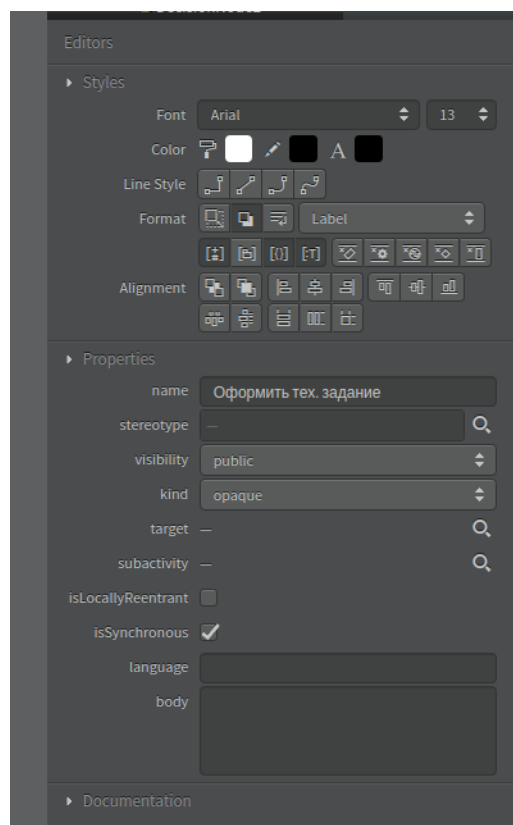


Рис. 6.42: Диаграмма активности. Меню *Editors*

### Добавление перехода и редактирование его свойств

Добавление перехода на диаграмму деятельности полностью аналогично диаграмме состояний. А именно, для добавления перехода между двумя деятельностями нужно с помощью левой кнопки мыши нажать кнопку с изображением перехода (*Control Flow*) на специальной панели инструментов, отпустить левую кнопку мыши, щелкнуть левой кнопкой мыши на изображении исходной деятельности на диаграмме и отпустить ее на изображении целевой деятельности (рис. 6.43.).)

В результате этих действий на диаграмме появится изображение перехода, соединяющего две выбранные деятельности. Если в качестве одной из деятельности является символ ветвления или соединения, то порядок добавления перехода сохраняется прежним.

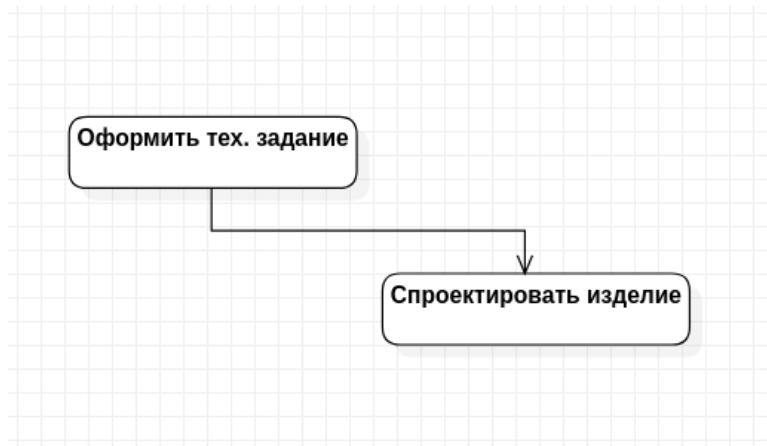


Рис. 6.43: Диаграмма активности. Добавление перехода

После добавления перехода на диаграмму деятельности становятся доступными для редактирования его свойства в специальном диалоговом окне, которое можно открыть по двойному щелчку левой кнопкой мыши на изображении перехода.

При спецификации свойств переходов следует помнить, что все переходы на диаграмме деятельности являются нетриггерными, т.е. не имеют имен событий. По этой причине поле ввода с именем *Name* (Имя) для всех переходов должно оставаться пустым. Но все переходы, выходящие из символов ветвления (решения), должны иметь сторожевые условия.

### Добавление дорожек на диаграмму деятельности

Для представления модели бизнес-процесса в форме диаграммы деятельности первоначально необходимо добавить на нее дорожки. Для добавления дорожки на диаграмму деятельности нужно с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы дорожки (*Swimlane*) на специальной панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы. Причем программа StarUML™ располагает возможностью добавить как вертикальную дорожку, так и горизонтальную, что иногда может быть полезно при построении диаграммы активности.

В результате этих действий на диаграмме в области диаграммы появится изображение дорожки с вертикальной линией и именем дорожки *ActivityPartition1* в верхней части, предложенное программой по умолчанию. Для задания имени дорожки следует дважды кликнуть левой кнопкой мыши на нее и вписать название в предложенную область (рис. 6.44).

Начиная практическую разработку модели бизнес-процесса разработки новой модели велосипеда, последовательно добавим на диаграмму деятельности дорожки с именами отдельных подразделений предприятия: *Управляющий отдел*, *Конструкторский отдел*, *Отдел тестирования*, *Отдел сертификации*, *Сборочный цех*. (рис. 6.45).



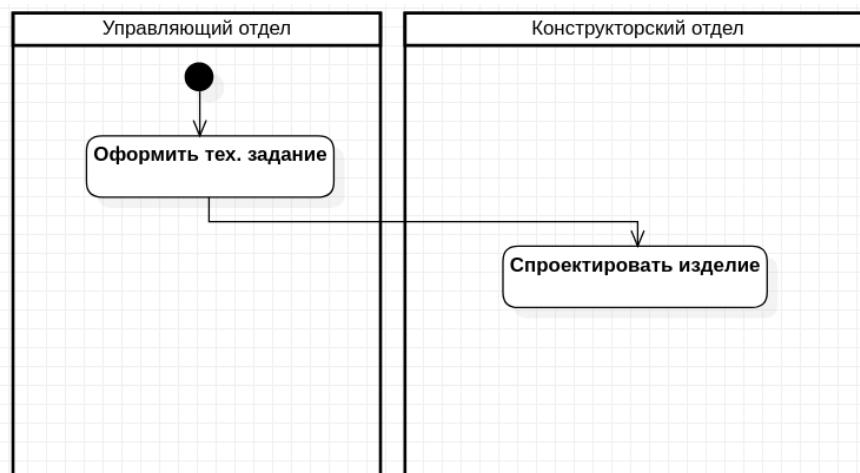


Рис. 6.46: Диаграмма активности. Добавление активности на Swimlane

дела тестирования, Отдела сертификации и Сборочного цеха (см рис 6.47.)

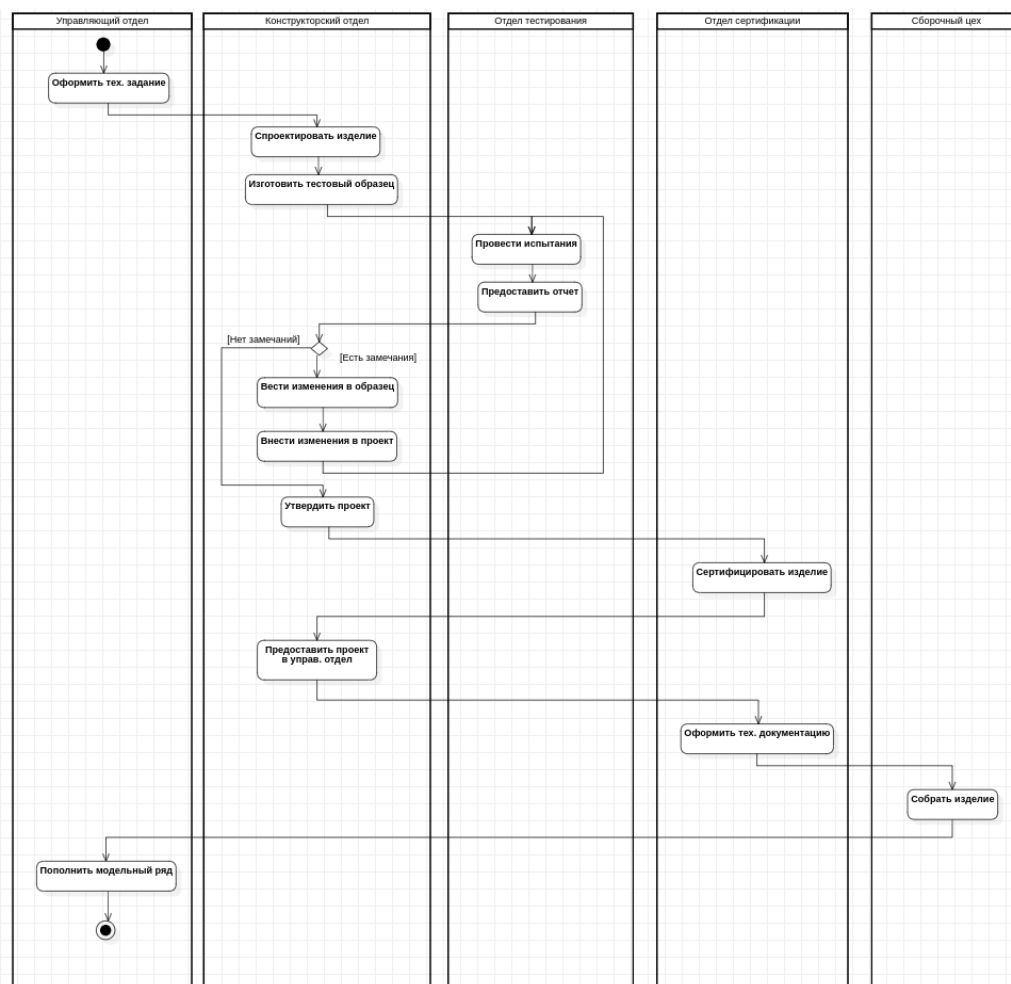


Рис. 6.47: Диаграмма активности с дорожками для модели бизнес-процесса

## Построение диаграммы деятельности с дорожками и потоком объектов

Для построения диаграммы деятельности с дорожками и потоком объектов для рассматриваемой модели бизнес-процесса следует добавить на диаграмму объекты и стрелки потоков объектов.

Для графического представления объектов используются прямоугольник класса, с тем отличием, что имя объекта подчеркивается. Далее после имени может указываться характеристика состояния объекта в прямых скобках. Такие прямоугольники объектов присоединяются к состояниям действия отношением зависимости пунктирной линией со стрелкой. Соответствующая зависимость определяет состояние конкретного объекта после выполнения предшествующего действия.

На диаграмме деятельности с дорожками расположение объекта может иметь некоторый дополнительный смысл. А именно, если объект расположен на границе двух дорожек, то это может означать, что переход к следующему состоянию действия в соседней дорожке ассоциирован с готовностью некоторого документа (объект в некотором состоянии). Если же объект целиком расположен внутри дорожки, то и состояние этого объекта целиком определяется действиями данной дорожки.

## 6.7 Особенности разработки диаграммы компонентов

Диаграмма компонентов служит частью физического представления модели, играет важную роль в процессе объектно-ориентированного анализа и проектирования *ООАП*.

Диаграммы компонентов используются для визуализации организации компонентов системы и зависимостей между ними. Они позволяют получить высокоуровневое представление о компонентах системы.

Компонентами могут быть программные компоненты, такие как база данных или пользовательский интерфейс; или аппаратные компоненты, такие как схема, микросхема или устройство; или бизнес-подразделение, такое как поставщик, платежная ведомость или доставка.

### Компонент

*Компонент* — это физическая часть системы. Компоненты, представляющие собой файлы с исходным кодом классов, библиотеки, исполняемые модули и т.п., которые должны обладать согласованным набором интерфейсов. Для их графического представления используются следующие графические символы.



Рис. 6.48: Диаграмма компонентов. Примеры компонентов

Внутри прямоугольника записывается имя компонента и, возможно, некоторая дополнительная информация в виде помеченного значения.

Компоненты могут иметь следующие стандартные стереотипы:

- «file» — любой файл, кроме таблицы:
  - «executable» — программа (исполняемый файл);
  - «library» — статическая или динамическая библиотека;
  - «source» — файл с исходным текстом программы;
  - «document» — остальные файлы (например, файл справки);
- «table» — таблица базы данных.

## Интерфейс

*Интерфейс* — это внешне видимый, именованный набор операций, который класс, компонент или подсистема может предоставить другому классу, компоненту или подсистеме, для выполнения им своих функций. В некоторых языках программирования, в частности в Java, интерфейс представляет собой отдельный класс, включаемый и реализуемый (конкретизируемый) в части программного кода операций в составе других классов. На диаграмме компонентов интерфейс отображается так же, как и на диаграмме классов (слева от компонента необходимые для работы интерфейсы, справа — предоставляемые).

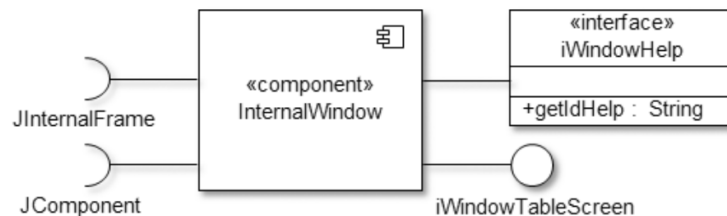


Рис. 6.49: Диаграмма компонентов. Пример интерфейсов

Начать построение диаграммы компонентов для выбранного элемента модели или моделируемой системы в целом можно одним из следующих способов:

- выбрать соответствующий вариант, перейдя по следующим пунктам меню:  
*Model* → *AddDiagram* → *ComponentDiagram*;
- нажать правую кнопку мыши на рабочей области и выбрать аналогичные пункты контекстного меню: *AddDiagram* → *ComponentDiagram*;

В результате выполнения этих действий появляется новое окно с чистым рабочим листом диаграммы компонентов и специальная панель инструментов, содержащая кнопки с изображением графических элементов, необходимых для разработки диаграммы. Назначение отдельных кнопок панели можно узнать из всплывающих подсказок. Более полный список предоставляемых инструментов можно видеть на рисунке 6.40.



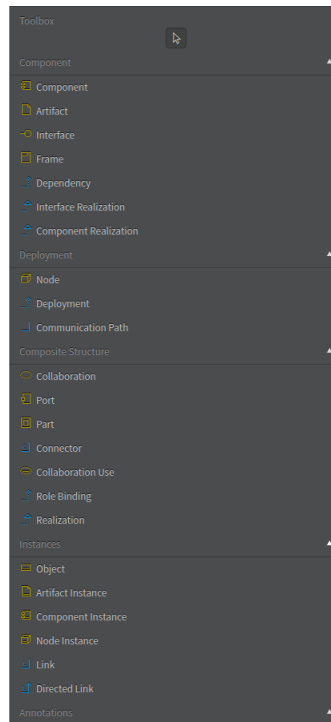


Рис. 6.50: Диаграмма компонентов. Панель инструментов

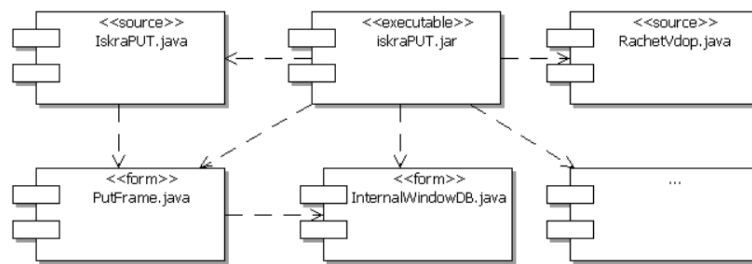


Рис. 6.51: Диаграмма компонентов. Пример ассоциации компонентов

## Отношение ассоциации

*Отношение ассоциации* отображается между компонентами и их интерфейсами.

*Отношение зависимости* означает зависимость реализации одних компонентов от реализации других. Такое возможно в следующих случаях:

- в методах классов одного компонента (зависимого) осуществляется вызов методов или обращение к атрибутам классов другого компонента (независимого);
- компонент состоит из других компонентов (например, при сборке исполняемого файла из файлов с исходными кодами);
- компонент осуществляет чтение или запись данных в другой компонент;
- связь между таблицами БД;

## Предмет моделирования

Объектом моделирования выберем систему контроля производственного процесса по созданию новой модели велосипеда на производстве.

Опишем данный программный комплекс. Приложение *Производство.exe* является главным и позволяет взаимодействовать с системой контроля. Данное приложение предоставляет следующий функционал:

- Добавление технической и проектной документации, связанной с проектом
- Регистрация результатов производственных испытаний
- Автоматическая генерация отчетной документации
- Хранение истории выполненных работ по каждому проекту

## Добавление компонента на диаграмму компонентов и редактирование его свойств

Для добавления компонента на диаграмму компонентов нужно с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы компонента на специальной панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы.

В результате этих действий на диаграмме появится изображение компонента с маркерами изменения его геометрических размеров и предложенным средой именем по умолчанию, которое разработчику следует изменить.

Построим для нее диаграмму компонентов модели системы контроля за производственным процессом. С этой целью изменим имя диаграммы, предложенное по умолчанию, на *ComponentDiagramCycle*, а для первого добавленного компонента зададим имя *Производство.exe* (рис. 6.52).



Рис. 6.52: Диаграмма компонентов. Добавление компонента

## Добавление отношения зависимости и редактирование его свойств

Добавление отношения зависимости на диаграмму компонентов аналогично добавлению соответствующего отношения на диаграмму вариантов использования. Продолжая разработку

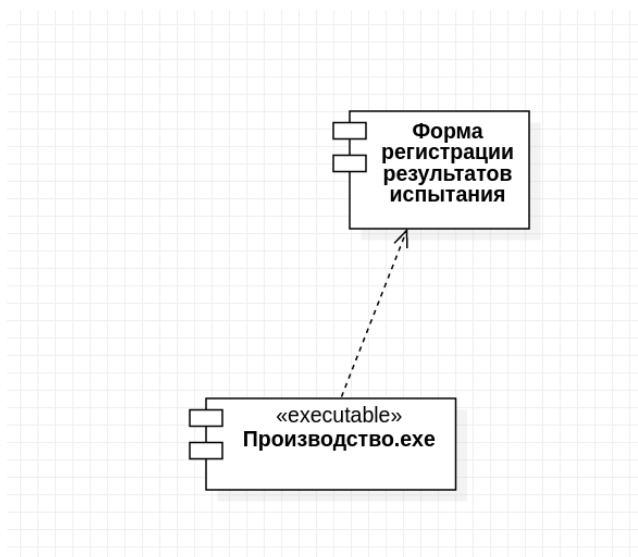


Рис. 6.53: Диаграмма компонентов. Добавление отношения зависимости

модели системы, на диаграмму компонентов предварительно следует добавить второй компонент с именем *Форма регистрации результатов испытания*.

Для добавления зависимости между двумя компонентами нужно с помощью левой кнопки мыши нажать кнопку с изображением зависимости на специальной панели инструментов, отпустить левую кнопку мыши, щелкнуть левой кнопкой мыши на изображении исходного компонента на диаграмме и отпустить ее на изображении целевого компонента. В результате этих действий на диаграмме появится изображение отношения зависимости в форме пунктирной линии со стрелкой, соединяющей два выбранных компонента.

Применительно к диаграмме компонентов модели системы рассмотренным способом следует добавить отношение зависимости от компонента с именем *Производство.exe* к компоненту с именем *Форма регистрации результатов испытания* (см. рис 6.53).

## Окончательное построение диаграммы компонентов

Для завершения построения диаграммы компонентов рассматриваемого примера следует описанным выше способом добавить оставшиеся компоненты и зависимости. С этой целью следует выполнить следующие действия:

1. Добавить компонент *Система журналирования*, добавить ассоциацию с компонентом *Производство.exe*
2. Добавить компонент *Брокер сообщений*, добавить ассоциацию с компонентом *Система журналирования*
3. Добавить компонент *Историческое хранилище*, добавить ассоциацию с компонентом *Система журналирования*
4. Добавить компонент *Компонент генерации отчетной документации*, добавить ассоциацию с компонентом *Производство.exe*

5. Добавить компонент *Документация*, добавить ассоциацию с компонентом *Форма проектной документации*
6. Добавить компонент *Форма добавления заявок на внесение изменений*, добавить ассоциацию с компонентом *Производство.exe*
7. Добавить компонент *Форма регистрации результатов испытания*, добавить ассоциацию с компонентом *Производство.exe*
8. Добавить компонент *Технологический процесс*, добавить ассоциации с компонентами *форма добавления заявок на внесение изменений* и *Форма регистрации результатов испытания*
9. Добавить компонент *Интерфейс приложения*, добавить ассоциацию с компонентом *Производство.exe*

Построенная таким образом диаграмма компонентов будет иметь следующий вид (рис.6.54).

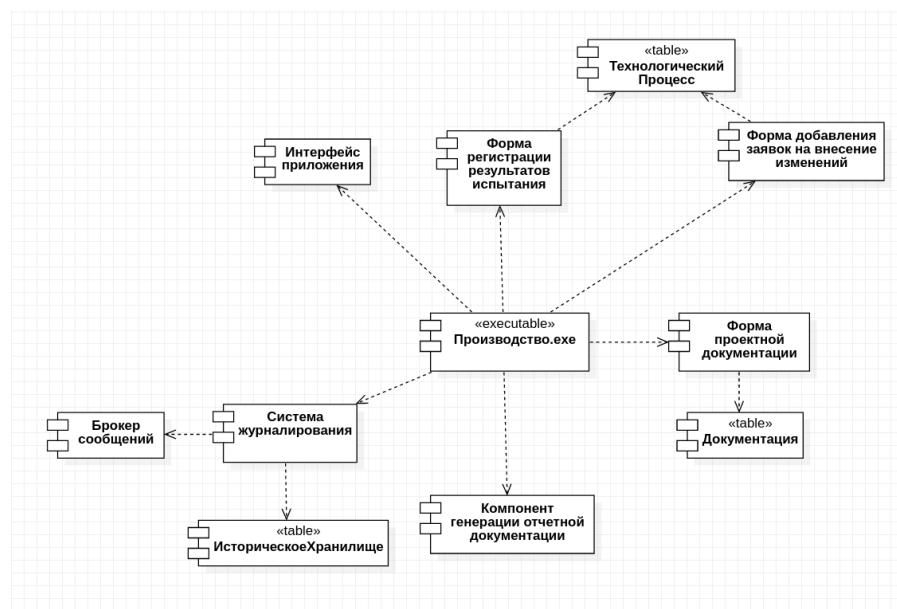


Рис. 6.54: Диаграмма компонентов

## 6.8 Особенности разработки диаграммы развертывания

Диаграмма развертывания является второй составной частью физического представления модели и разрабатывается, как правило, для территориально распределенных систем.

Начать построение диаграммы развертывания для выбранного элемента модели или моделируемой системы в целом можно одним из следующих способов:

- выбрать соответствующий вариант, перейдя по следующим пунктам меню:  
*Model* → *AddDiagram* → *DeploymentDiagram*;
- нажать правую кнопку мыши на рабочей области и выбрать аналогичные пункты контекстного меню: *AddDiagram* → *DeploymentDiagram*;

В результате выполнения этих действий появляется новое окно с чистым рабочим листом диаграммы развертывания и специальная панель инструментов, содержащая кнопки с изображением графических примитивов, необходимых для разработки диаграммы развертывания (см рис. 6.55).

Работа с диаграммой развертывания состоит в создании процессоров и устройств, их спецификации, установлении связей между ними, а также добавлении и спецификации процессов.

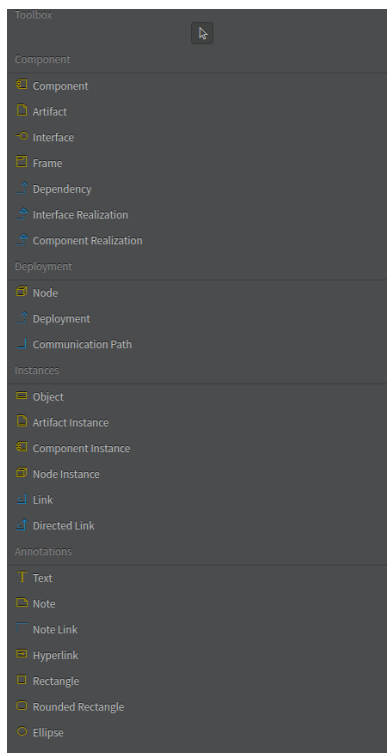


Рис. 6.55: Диаграмма развертывания. Панель инструментов

## Элементы диаграммы

### Узел

*Узел (node)* представляет собой физически существующий элемент системы, который может обладать вычислительным ресурсом или являться техническим устройством.

В качестве вычислительного ресурса узла может рассматриваться один или несколько процессоров, а также объем электронной или магнитооптической памяти. Однако в языке UML понятие узла включает в себя не только вычислительные устройства (процессоры), но и другие механические или электронные устройства, такие как датчики, принтеры, модемы, цифровые камеры, сканеры и манипуляторы.

Наиболее известны два специальных графических стереотипа для обозначения разновидностей узлов. Первый обозначает ресурсоемкий узел (*processor*), под которым понимается узел с процессором и памятью, необходимыми для выполнения исполняемых компонентов. Он изображается в форме куба с боковыми гранями, окрашенными в серый цвет (рис. 6.56, а).

Второй стереотип в форме обычного куба обозначает устройство (*device*), под которым понимается узел без процессора и памяти (рис. 6.56, б). На этом типе узлов не могут размещаться исполняемые компоненты программной системы.



Рис. 6.56: Диаграмма развертывания. Виды узлов

### Соединения и зависимости на диаграмме развертывания

На диаграмме развертывания кроме изображения узлов указываются отношения между ними. В качестве отношений выступают физические соединения между узлами, а также зависимости между узлами и компонентами, которые допускается изображать на диаграммах развертывания.

Соединения являются разновидностью ассоциации и изображаются отрезками линий без стрелок. Наличие такой линии указывает на необходимость организации физического канала для обмена информацией между соответствующими узлами. Характер соединения может быть дополнительно специфицирован примечанием, стереотипом, помеченным значением или ограничением. Так, на представленном ниже фрагменте диаграммы развертывания (рис. 6.57) явно определены рекомендации по технологии физической реализации соединений в форме примечания.

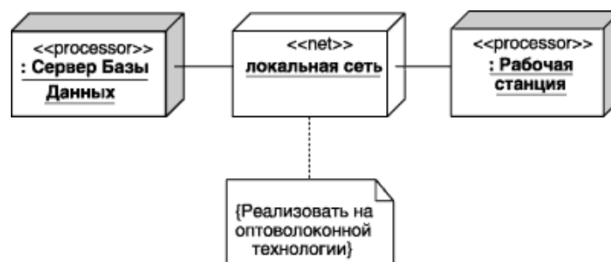


Рис. 6.57: Диаграмма развертывания. Соединения между узлами

Кроме соединений на диаграмме развертывания могут присутствовать отношения зависимости между узлом и размещаемыми на нем компонентами. Подобный способ представляет собой альтернативу вложенному изображению компонентов внутри символа узла, что не всегда удобно, поскольку делает этот символ излишне объемным. При большом количестве развернутых на узле компонентов соответствующую информацию можно представить в форме отношения зависимости (рис. 6.58).



Рис. 6.58: Диаграмма развертывания. Соединение компонентов

## Добавление узла на диаграмму развертывания и редактирование его свойств

Для добавления узла на диаграмму развертывания нужно с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы требуемого узла (процессора или устройства) на специальной панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы.

В результате этих действий на диаграмме развертывания появится изображение узла требуемого типа с маркерами изменения его геометрических размеров и предложенным средой именем по умолчанию. Изменить стереотип узла можно в секции *Editors*, введя название стереотипа в поле *stereotype*.

Добавим узел *СерверБазыДанных* на диаграмму развертывания, указав для него стереотип *processor*. Данный узел представляет сервер, на котором размещаются файлы, используемые производством в ходе технологического процесса. Это могут быть как файлы, обрабатываемые СУБД, так и файлы прикладных программ и утилит.



Рис. 6.59: Диаграмма развертывания. Узел «СерверБазыДанных»

## Добавление соединения и редактирование его свойств

Для добавления соединения между двумя узлами нужно с помощью левой кнопки мыши нажать кнопку с изображением соединения на специальной панели инструментов, отпустить левую кнопку мыши, щелкнуть левой кнопкой мыши на изображении одного из узлов на диаграмме и отпустить ее на изображении другого узла.

В результате этих действий на диаграмме появится изображение соединения в форме линии без стрелок, соединяющей два выбранных узла. Применительно к диаграмме развертывания модели системы по контролю за производственным процессом одним из рассмотренных способов следует добавить соединение для узлов с именами *СерверБазыДанных* и *ЗакрывающаяСеть* (см. рис. 6.60).

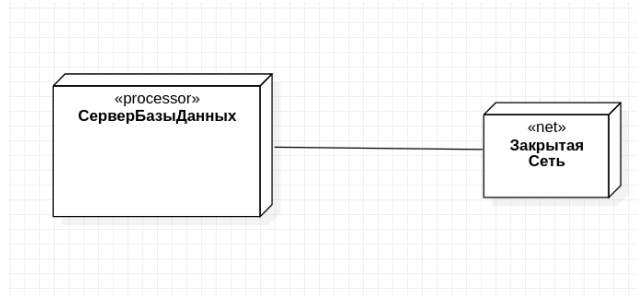


Рис. 6.60: Диаграмма развертывания. Добавление соединения

## Окончательное построение диаграммы развертывания

Для завершения построения диаграммы развертывания рассматриваемого примера следует описанным выше способом добавить оставшиеся узлы и соединения. С этой целью следует выполнить следующие действия:

1. Добавить узел **Рабочее место УО**, добавить связь с узлом **Закрывающая сеть**
2. Добавить узел **Рабочее место КО**, добавить связь с узлом **Закрывающая сеть**
3. Добавить узел **Рабочее место ТО**, добавить связь с узлом **Закрывающая сеть**
4. Добавить узел **Рабочее место СО**, добавить связь с узлом **Закрывающая сеть**
5. Добавить узел **Рабочее место СЦ**, добавить связь с узлом **Закрывающая сеть**
6. Добавить компонент **Производство.exe**. Соединить все узлы рабочих мест связью ассоциации с этим компонентом

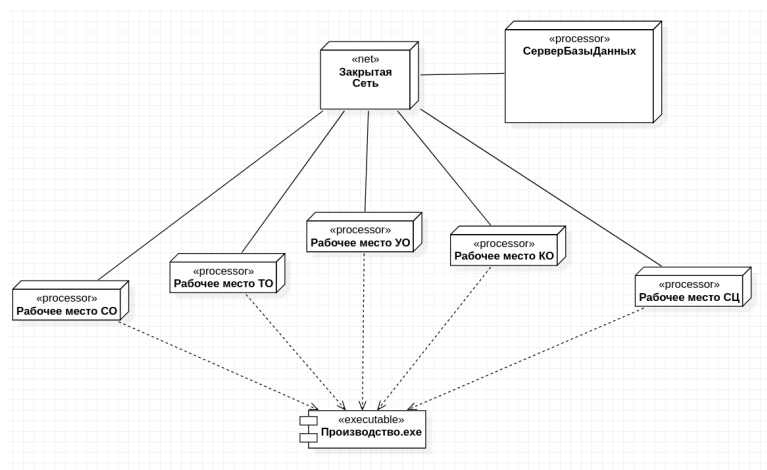


Рис. 6.61: Диграмма развертывания



Построенная таким образом диаграмма развертывания будет иметь следующий вид причем для данной диаграммы показаны выполняемые на процессорах процессы и не показаны процедуры их планирования. Это сделано по той причине, что при наличии единственного процесса планирование ресурсов процессора теряет свое значение.

# Список иллюстративного материала

Рисунок 2.1	Модель, представление, диаграмма . . . . .	4
Рисунок 3.1	Сохранение проекта . . . . .	9
Рисунок 3.2	Закрытие проекта . . . . .	9
Рисунок 5.1	Главное окно . . . . .	18
Рисунок 5.2	Меню File . . . . .	19
Рисунок 5.3	Меню Edit . . . . .	20
Рисунок 5.4	Меню Format . . . . .	21
Рисунок 5.5	Меню View . . . . .	22
Рисунок 5.6	Общие инструменты . . . . .	25
Рисунок 5.7	Особые инструменты. Ч1 . . . . .	25
Рисунок 5.8	Особые инструменты. Ч2 . . . . .	26
Рисунок 5.9	Особые инструменты. Ч3 . . . . .	27
Рисунок 5.10	Особые инструменты. Ч4 . . . . .	28
Рисунок 5.11	Особые инструменты. Ч5 . . . . .	29
Рисунок 6.1	Use Case: Панель инструментов . . . . .	31
Рисунок 6.2	Use Case:Actor . . . . .	31
Рисунок 6.3	Спецификация актера . . . . .	33
Рисунок 6.4	Измененная спецификация актера . . . . .	34
Рисунок 6.5	Вариант использования . . . . .	34
Рисунок 6.6	Дополненная спецификация варианта использования . . . . .	35
Рисунок 6.7	Ассоциация актера с вариантом использования . . . . .	35
Рисунок 6.8	Свойство Navigable . . . . .	36
Рисунок 6.9	Отношение зависимости между вариантами использования . . . . .	37
Рисунок 6.10	Диаграмма вариантов использования . . . . .	38
Рисунок 6.11	Class Diagram:Панель инструментов . . . . .	39
Рисунок 6.12	Класс «Конструкторский отдел» . . . . .	40
Рисунок 6.13	Секция Editors . . . . .	41
Рисунок 6.14	Ассоциация . . . . .	43
Рисунок 6.15	Кратность ассоциации . . . . .	43
Рисунок 6.16	Абстрактный класс . . . . .	44
Рисунок 6.17	Диаграмма Классов . . . . .	45
Рисунок 6.18	Communication Diagram: Панель инструментов . . . . .	46

Рисунок 6.19	Объекты диаграммы кооперации . . . . .	47
Рисунок 6.20	Контекстное меню . . . . .	48
Рисунок 6.21	Изображение связи . . . . .	48
Рисунок 6.22	Выбор сообщения (метода) . . . . .	48
Рисунок 6.23	Рефлексивное сообщение . . . . .	49
Рисунок 6.24	Диаграмма кооперации . . . . .	51
Рисунок 6.25	Объекты диаграммы последовательности . . . . .	52
Рисунок 6.26	Рефлексивная связь . . . . .	53
Рисунок 6.27	Отображение связи между объектами . . . . .	53
Рисунок 6.28	Свойства связи . . . . .	54
Рисунок 6.29	Меню обозревателя модели . . . . .	54
Рисунок 6.30	Пример альтернативного фрагмента . . . . .	55
Рисунок 6.31	Пример справочного и вариантного фрагмента . . . . .	56
Рисунок 6.32	Настройки фрагмента . . . . .	57
Рисунок 6.33	Диаграмма последовательности . . . . .	57
Рисунок 6.34	Диаграмма состояний: Панель инструментов . . . . .	59
Рисунок 6.35	Добавление состояния на диаграмму . . . . .	60
Рисунок 6.36	Добавление перехода на диаграмму . . . . .	60
Рисунок 6.37	Составное состояние . . . . .	61
Рисунок 6.38	параллельные состояния . . . . .	62
Рисунок 6.39	Диаграмма состояний . . . . .	63
Рисунок 6.40	Диаграмма Активности: Панель инструментов . . . . .	65
Рисунок 6.41	Диаграмма активности. Добавление активности . . . . .	66
Рисунок 6.42	Диаграмма активности. Меню <i>Editors</i> . . . . .	66
Рисунок 6.43	Диаграмма активности. Добавление перехода . . . . .	67
Рисунок 6.44	Диаграмма активности. Добавление Swimlane . . . . .	68
Рисунок 6.45	Диаграмма активности. Добавление дорожек отделов . . . . .	68
Рисунок 6.46	Диаграмма активности. Добавление активности на Swimlane . . . . .	69
Рисунок 6.47	Диаграмма активности с дорожками для модели бизнес-процесса . . . . .	69
Рисунок 6.48	Диаграмма компонентов. Примеры компонентов . . . . .	70
Рисунок 6.49	Диаграмма компонентов. Пример интерфейсов . . . . .	71
Рисунок 6.50	Диаграмма компонентов. Панель инструментов . . . . .	72
Рисунок 6.51	Диаграмма компонентов. Пример ассоциации компонентов . . . . .	72
Рисунок 6.52	Диаграмма компонентов. Добавление компонента . . . . .	73
Рисунок 6.53	Диаграмма компонентов. Добавление отношения зависимости . . . . .	74
Рисунок 6.54	Диаграмма компонентов . . . . .	75
Рисунок 6.55	Диаграмма развертывания. Панель инструментов . . . . .	76
Рисунок 6.56	Диаграмма развертывания. Виды узлов . . . . .	77
Рисунок 6.57	Диаграмма развертывания. Соединения между узлами . . . . .	77
Рисунок 6.58	Диаграмма развертывания. Соединение компонентов . . . . .	78

Рисунок 6.59	Диаграмма развертывания. Узел «СерверБазыДанных» . . . . .	78
Рисунок 6.60	Диаграмма развертывания. Добавление соединения . . . . .	79
Рисунок 6.61	Диаграмма развертывания . . . . .	79

## Список таблиц

6.1	Use Case: Назначение кнопок . . . . .	32
6.2	Class Diagram: Назначение пунктов меню . . . . .	40
6.3	Свойства сообщений . . . . .	49
6.4	Диаграмма последовательности: Панель инструментов . . . . .	52
6.5	Диаграмма состояний: Назначения инструментов . . . . .	58
6.6	Диаграмма активности: Назначение основных инструментов . . . . .	64