UML

Unified Modeling Language — унифицированный язык моделирования

Определение и назначение

UML – графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех артефактов, создаваемых при разработке систем.

UML –

- (полу) формальное
- (иногда) удобное
- (почти) универсальное

средство для уменьшения расхождений в толковании спецификаций

Способы использования UML

- Рисование картинок
- Обмен информацией
- Спецификация систем
- Повторное использование архитектурных решений
- Генерация кода
- Имитационное моделирование
- Верификация моделей

Модель

Модель UML – это совокупность конечного множества конструкций языка, главные из которых – сущности и отношения.

Сущности (1)

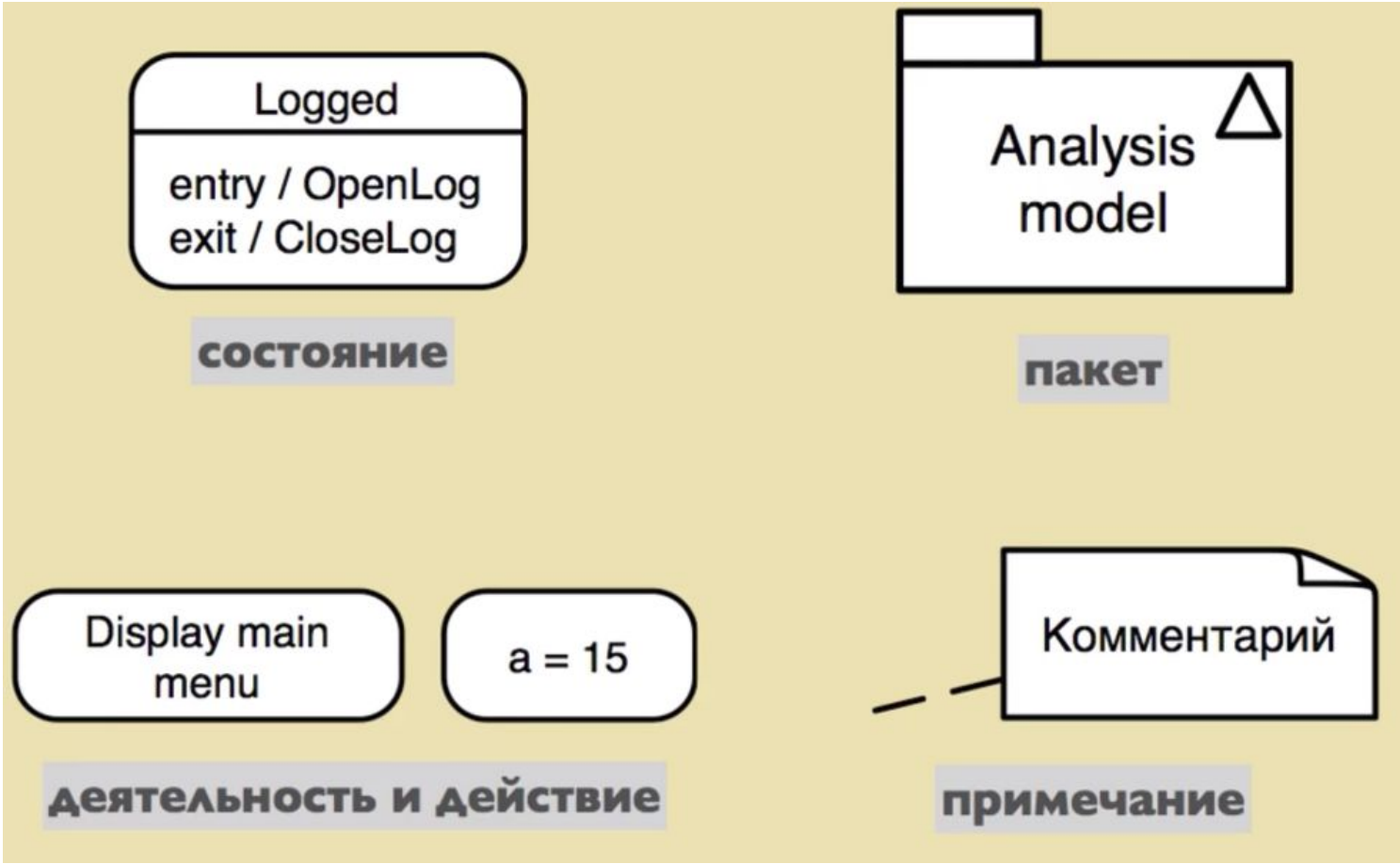
Тип сущности	Название	Перевод
Структурные	артефакт	artifact
	вариант использования	use case
	действующее лицо	actor
	интерфейс	interface
	класс	class
	компонент	component
	кооперация	collaboration
	объект	object
	узел	node

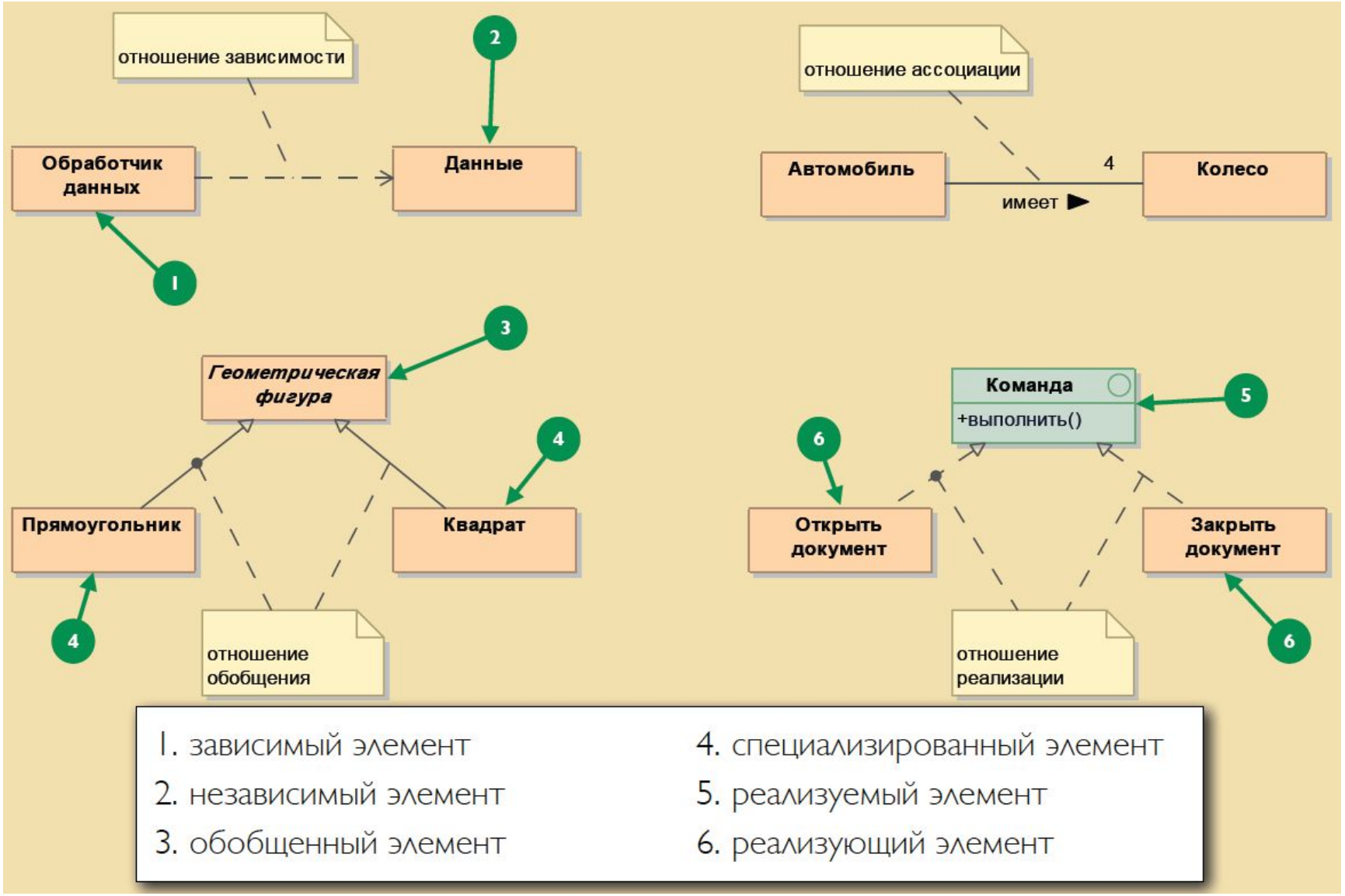
Нотации сущностей (1)

Сущности (2)

Тип сущности	Название	Перевод
Структурные	действие	action
	деятельность	activity
	состояние	state
Группирующие	пакет	package
Аннотационные	примечание	note

Нотация сущностей (2)





Примеры отношений в UML

Диаграммы

Диаграмма (1) – это графическое представление некоторой части графа.

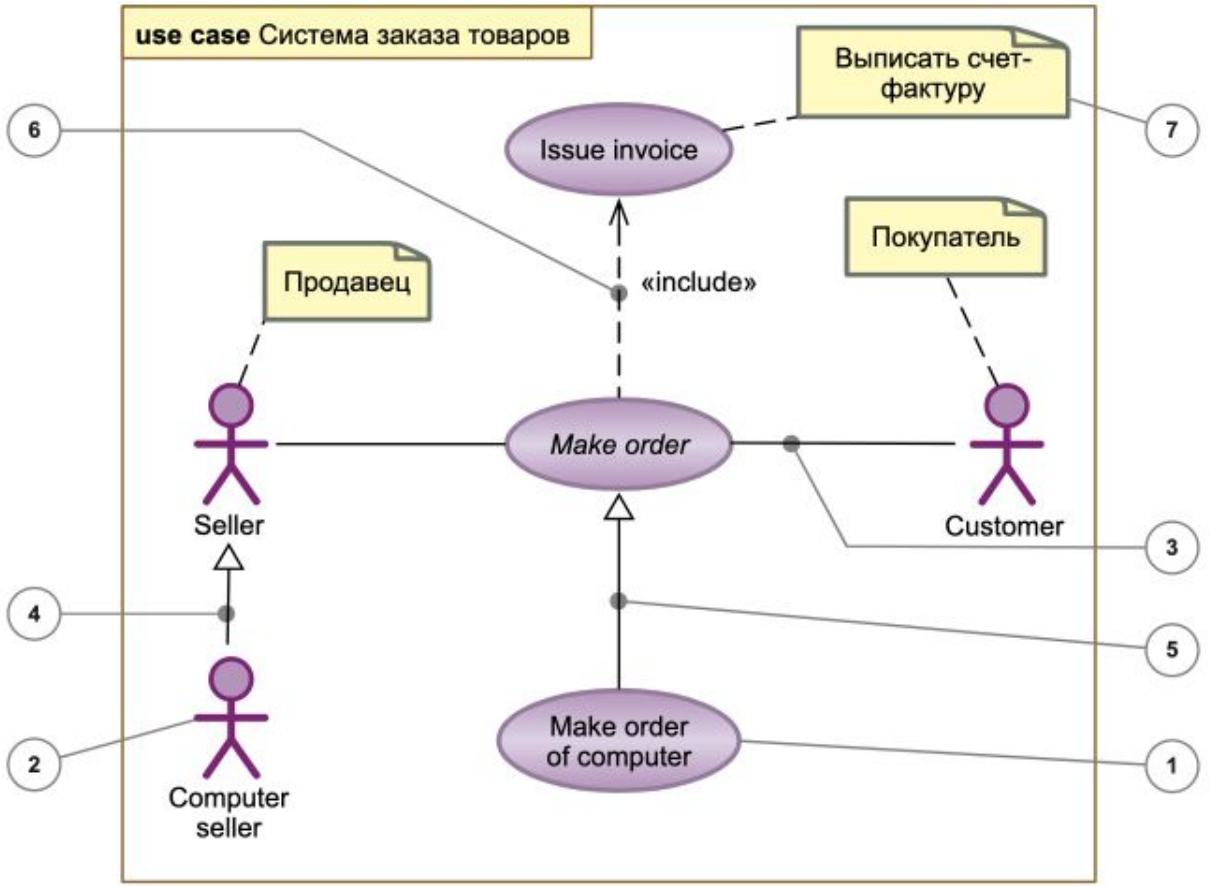
Диаграмма (2) – это накладываемая на модель **структура**, которая облегчает создание и использование модели.

Модель – объединение диаграмм.

Существует 13 видов диаграмм (Одна из диаграмм, например, может описывать взаимодействие пользователя с системой, другая - изменение состояний системы в процессе ее работы, третья - взаимодействие между собой элементов системы и т. д.)

Диаграмма использования (use case diagram)

Прецедент (use-case) - описание отдельного аспекта поведения системы с точки зрения пользователя.
Диаграмма использования – это наиболее общее представление функционального назначения системы.



Варианты использования – 1; Действующие лица – 2; Ассоциация между действующим лицом и вариантом использования – 3; Обобщение между действующими лицами – 4; Обобщение между вариантами использования – 5; Зависимости между вариантами использования – 6; Комментарии – 7.

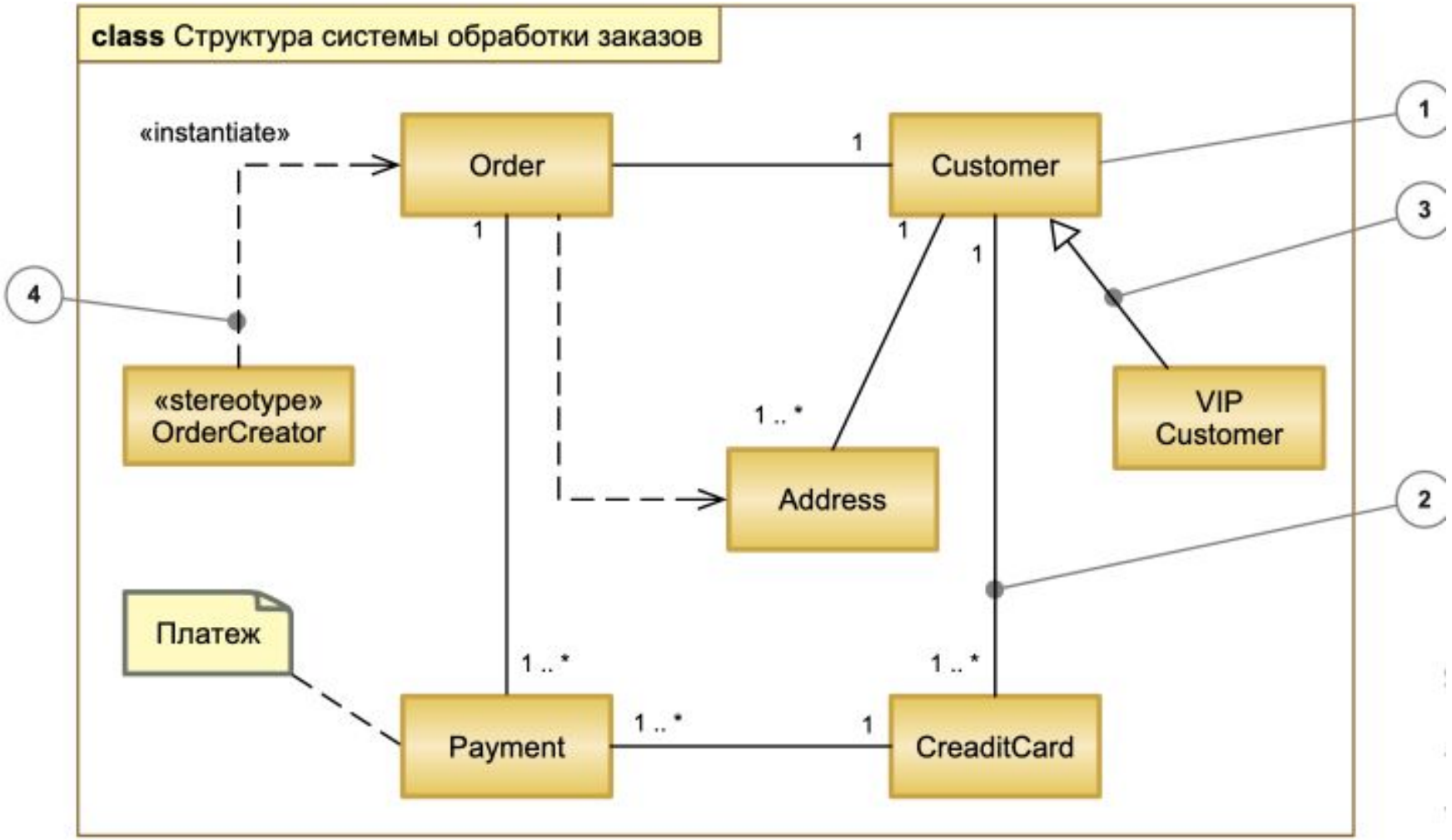
Диаграмма использования

Слева направо:
- Пример диаграммы использования
- Отношение включения на диаграмме использования
- Отношение расширения на диаграмме использования

UNIFIED MODELING LANGUAGE™

Диаграмма классов (Static Structure diagram)

Диаграмма классов (class diagram) – основной способ описания структуры системы.



Классы – 1; Ассоциация между классами – 2; Обобщение между классами – 3; Зависимости между классами – 4.

```
1.  public class Rectangle2D {
2.      private int _x;
3.      private int _y;
4.      private int _width;
5.      private int _height;
6.
7.      public Rectangle2D( int x, int y, int width, int height ) {
8.          _x = x;
9.          _y = y;
10.         _width = width;
11.         _height = height;
12.     }
13.
14.     public int GetSquare() {
15.         return _width * _height;
16.     }
17. }
```

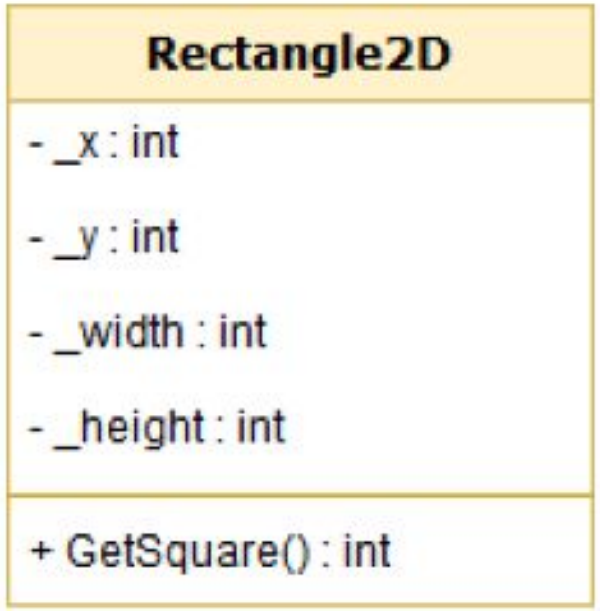


Диаграмма классов

Слева направо:
- Код примера представления класса на диаграмме
- Диаграмма классов кода

В этом коде определен тестовый класс **Rectangle2D**. Он содержит 4 закрытых поля – *_x, _y, _width, _height*, и один открытый метод – **GetSquare()**.

UNIFIED MODELING LANGUAGE™

Полезные материалы

На диаграммах классов UML могут быть представлены следующие отношения между классами: зависимость, агрегация, ассоциация, реализация и наследование.

Хорошая статья по связям между классами:

https://habr.com/ru/post/150041/

Инструкция по формированию uml диаграммы по коду в Visual Studio:

https://www.visual-paradigm.com/tutorials/visual-studio-uml-to-csharp-tutorial.jsp ;
https://www.c-sharpcorner.com/UploadFile/deveshomar/class-diagram-in-C-Sharp/

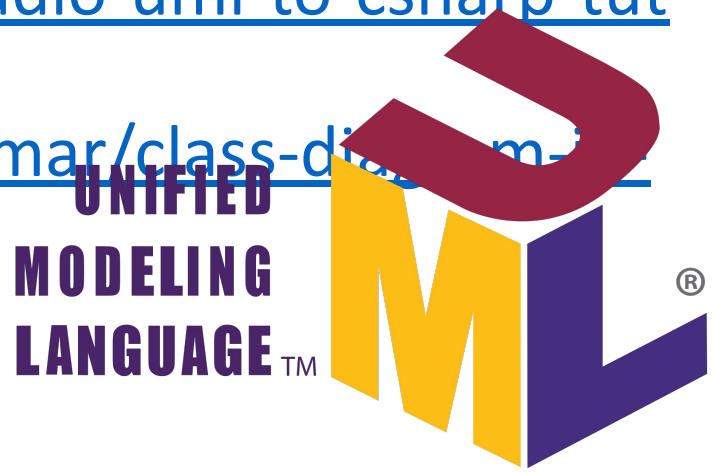
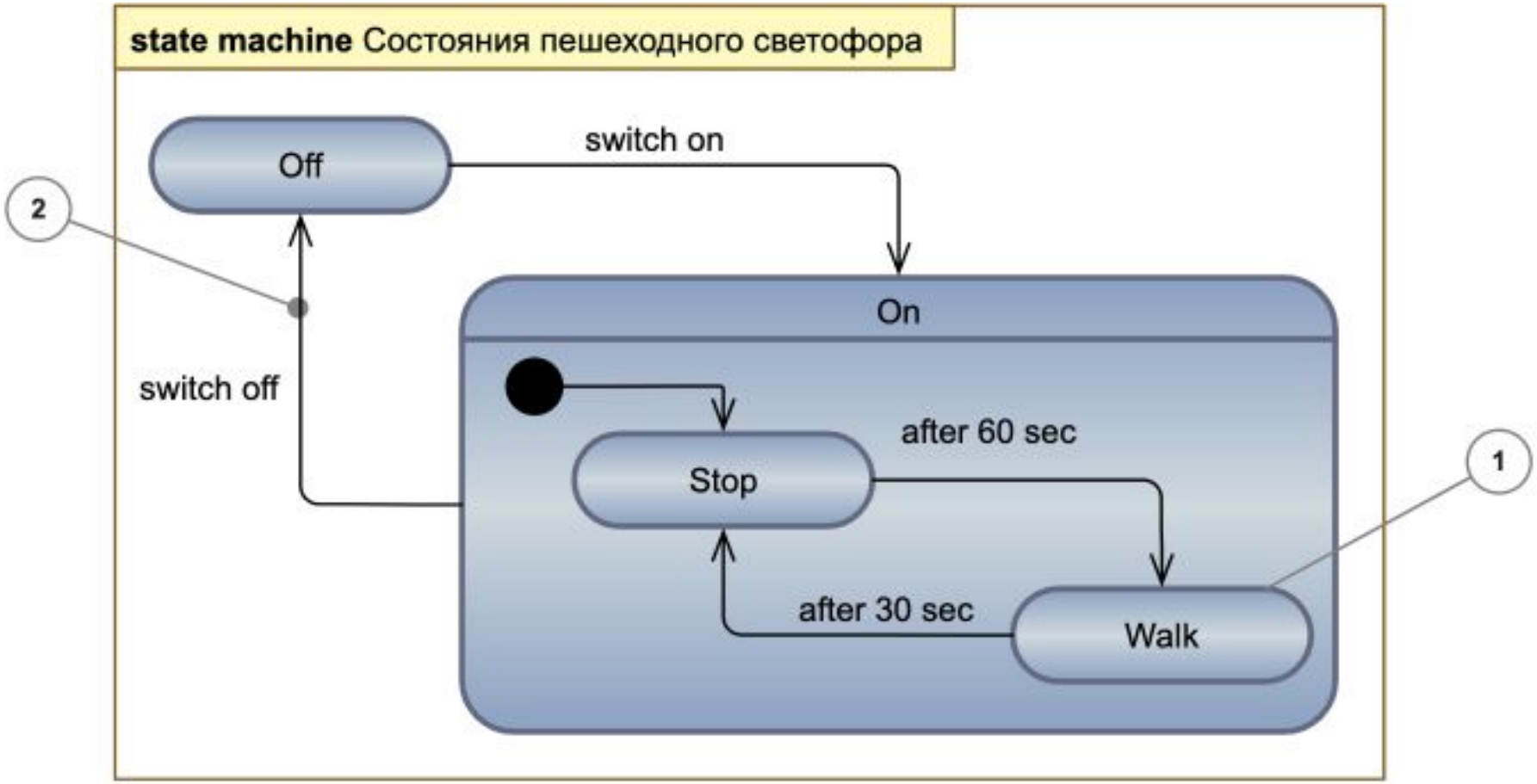


Диаграмма состояний (диаграмма автомата)

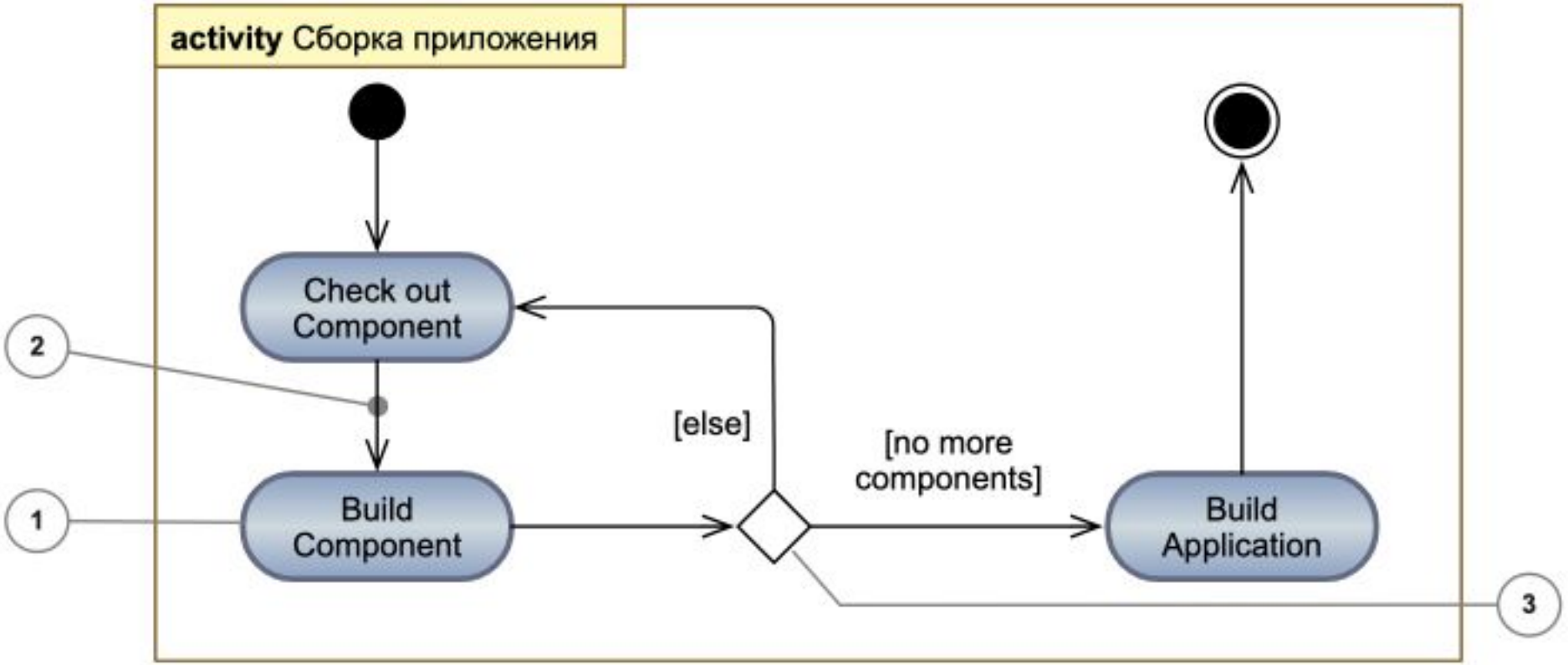
Диаграмма состояний – это один из способов детального описания поведения в UML на основе явного выделения состояний и описания переходов между ними.



Состояния – 1; Переходы – 2.

Диаграмма деятельности

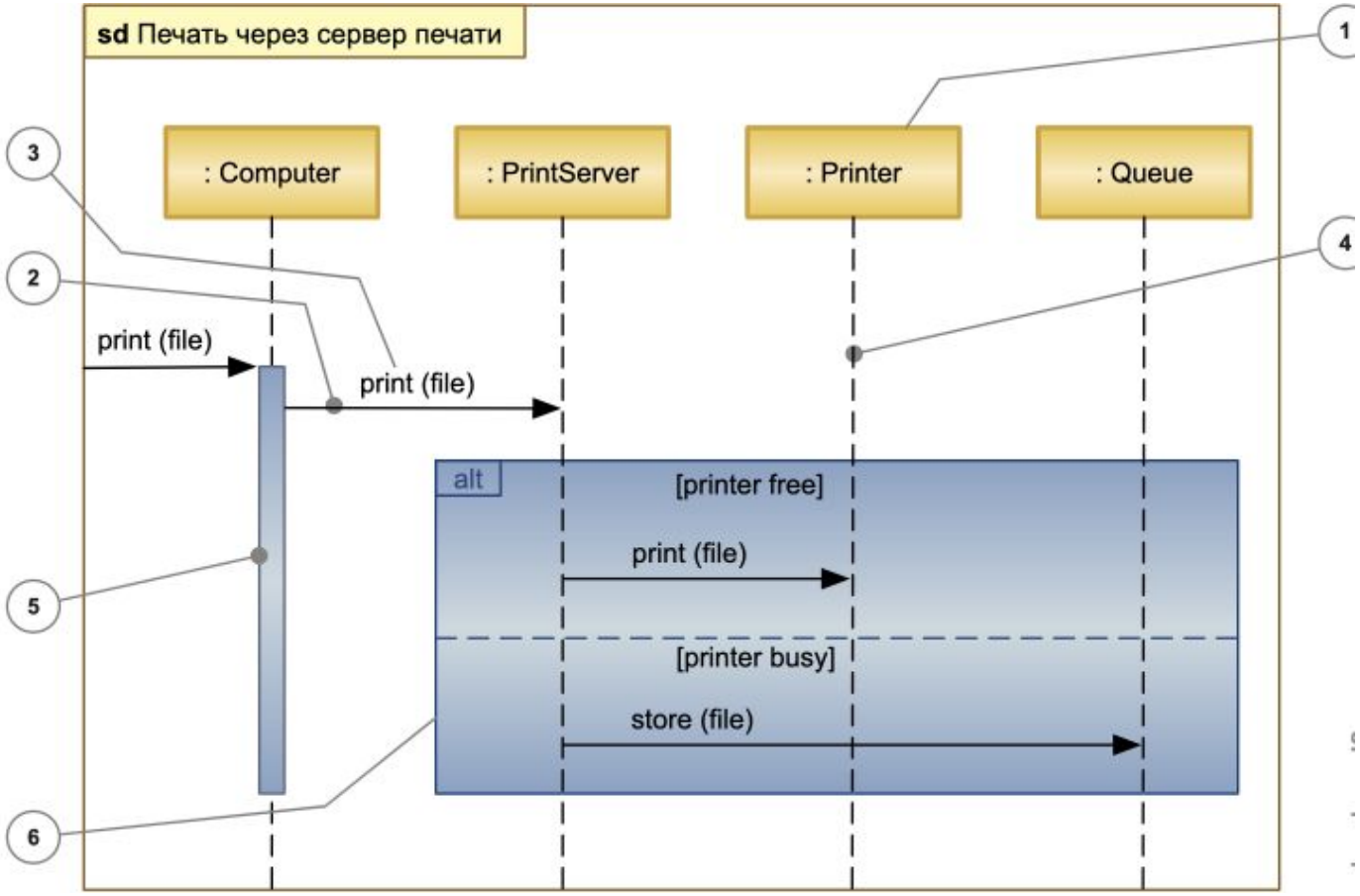
Диаграмма деятельности (activity diagram) – способ описания поведения на основе указания потоков управления и потоков данных.



Действие – 1; Переходы – 2; Развилки, слияния, соединения, ветвления – 3.

Диаграмма последовательности

Диаграмма последовательности (sequence diagram) – это способ описания поведения системы на основе указания последовательности передаваемых сообщений.



Экземпляры классификаторов – 1; Тип отношения – 2; Обмен сообщениями – 3; Линия жизни – 4; Место активации объеквзаимодействующих та – 5; Алгоритмические аспекты протокола взаимодействия – 6.

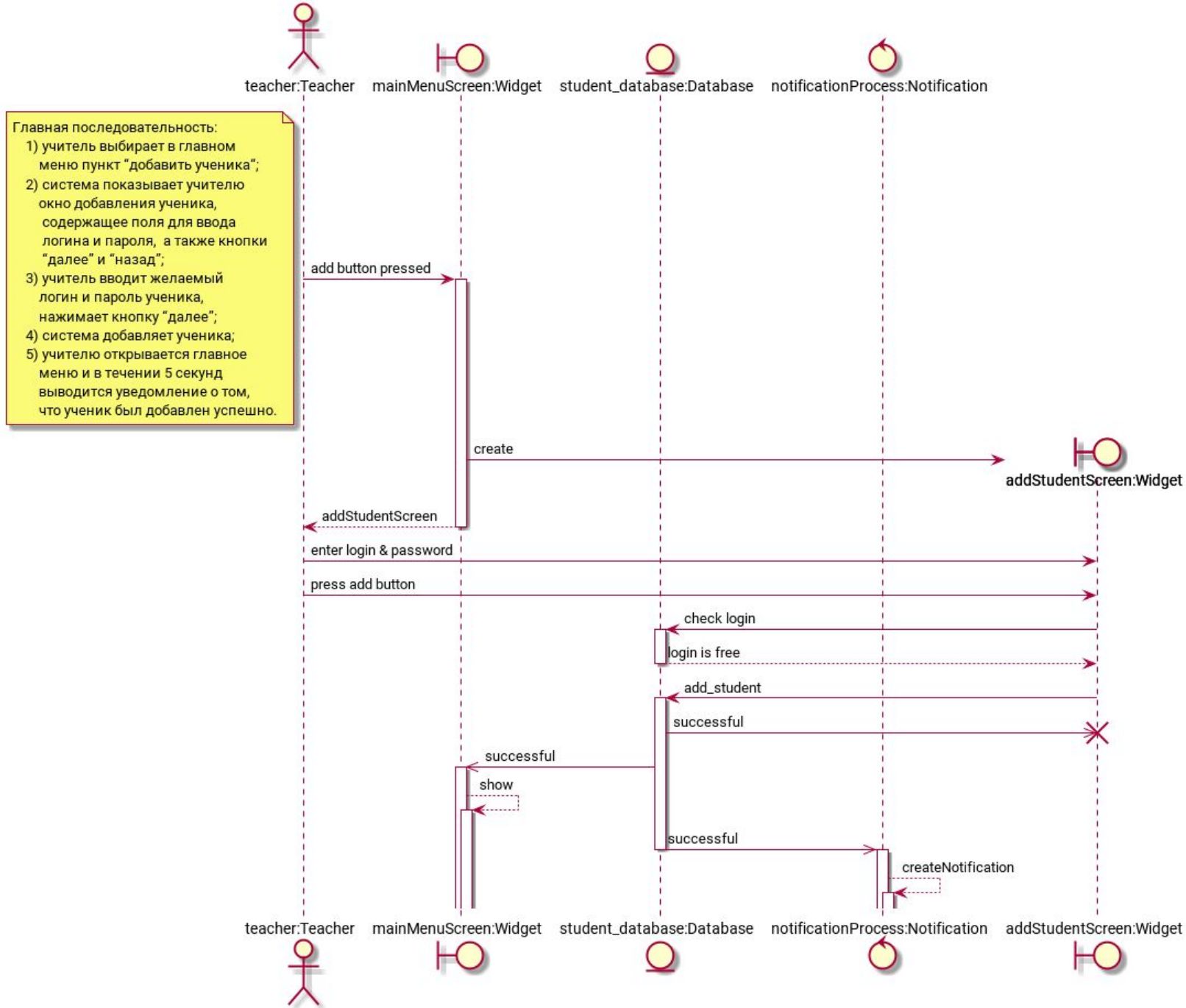
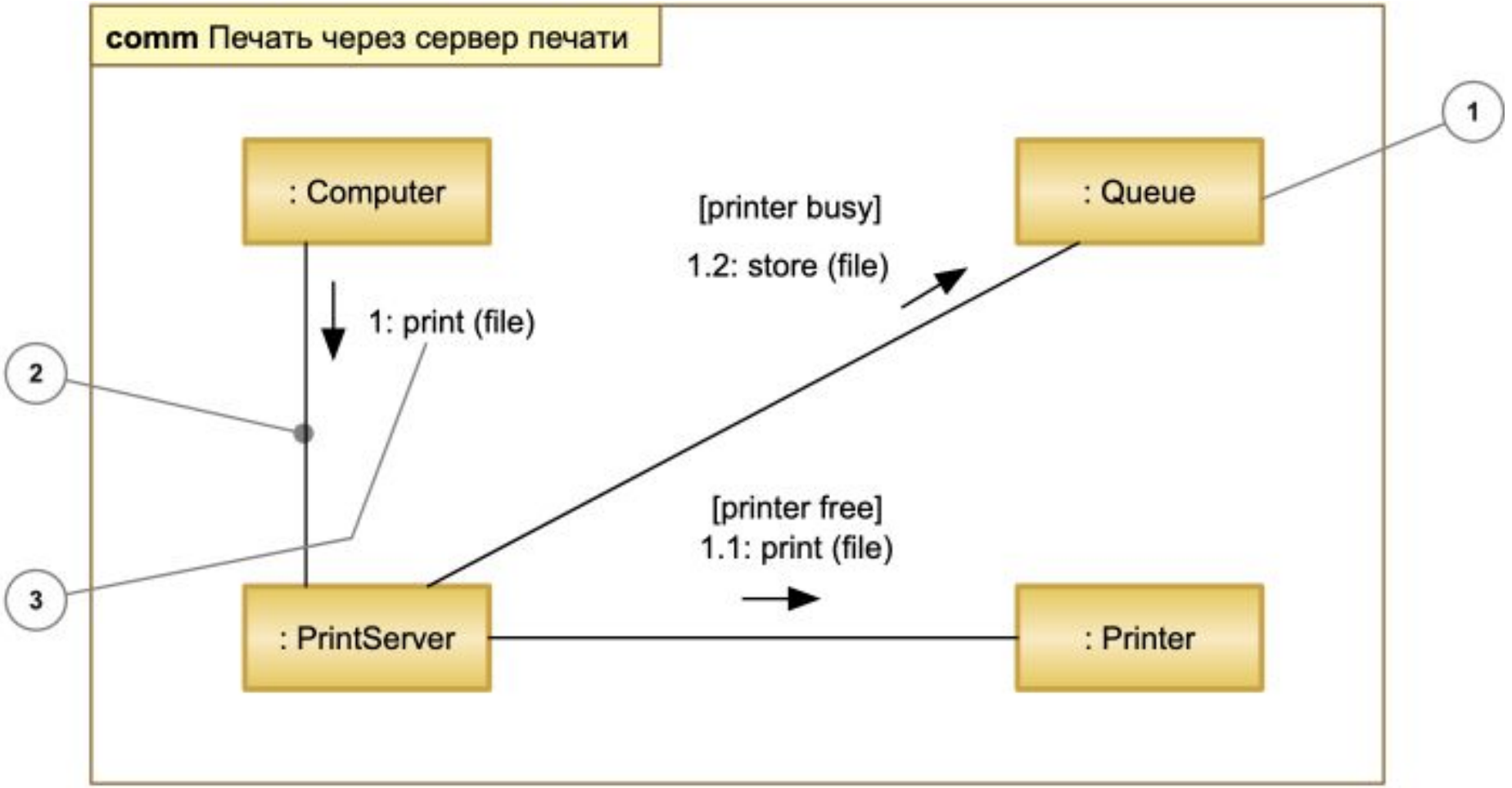

Главная последовательность:
1) учитель выбирает в главном меню пункт “добавить ученика”;
2) система показывает учителю окно добавления ученика, содержащее поля для ввода логина и пароля, а также кнопки “далее” и “назад”;
3) учитель вводит желаемый логин и пароль ученика, нажимает кнопку “далее”;
4) система добавляет ученика;
5) учителю открывается главное меню и в течении 5 секунд выводится уведомление о том, что ученик был добавлен успешно.
teacher:Teacher
mainMenuScreen:Widget
student_database:Database
notificationProcess:Notification
addStudentScreen:Widget
add button pressed
create
addStudentScreen
enter login & password
press add button
check login
login is free
add_student
successful
successful
show
successful
createNotification
teacher:Teacher
mainMenuScreen:Widget
student_database:Database
notificationProcess:Notification
addStudentScreen:Widget

Диаграмма коммуникации

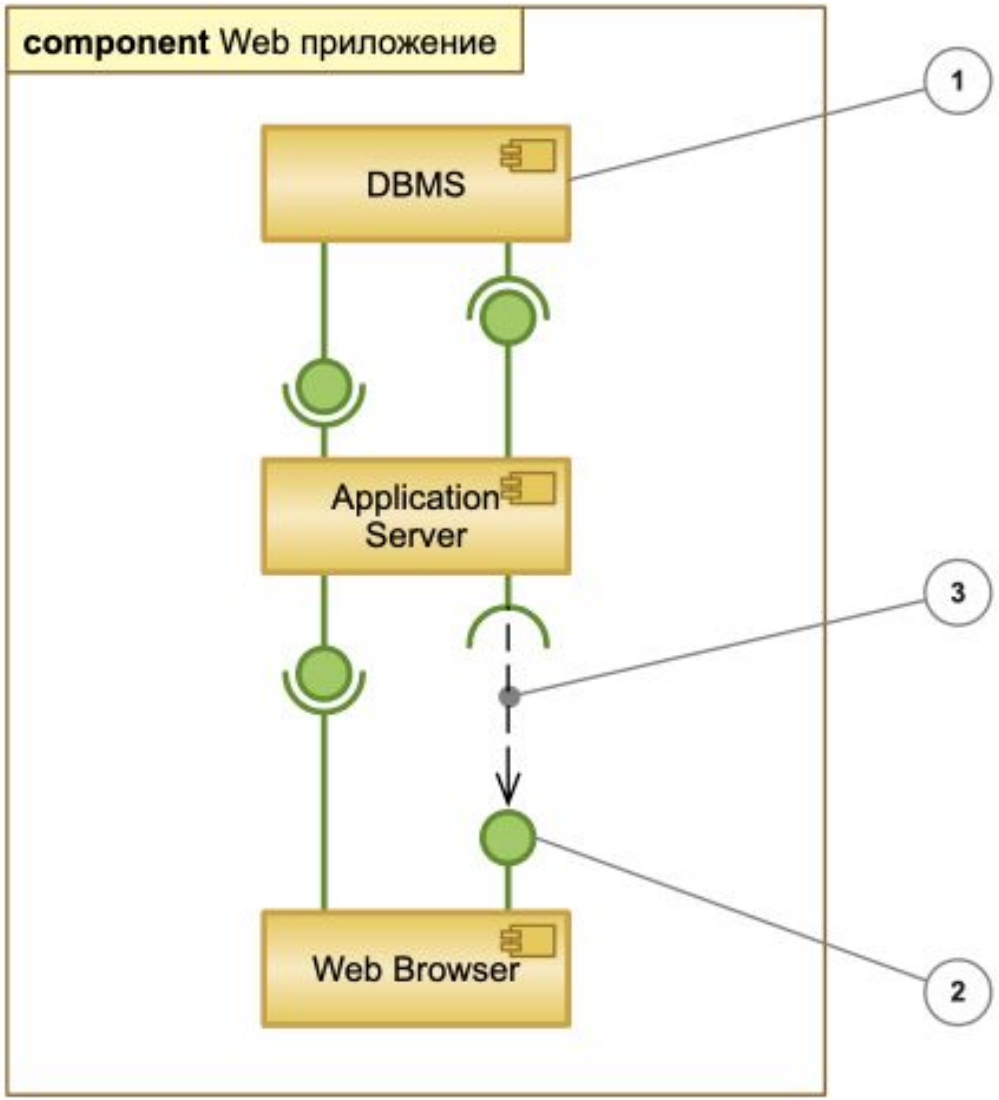
Диаграмма коммуникации (communication diagram) – способ описания поведения, семантически эквивалентный диаграмме последовательности. Главное отличие: делается акцент не на времени, а на структуре связей между кокретными экземплярами.



Экземпляры классификаторов – 1; Тип отношения – 2; Обмен сообщениями – 3.

Диаграмма компонентов

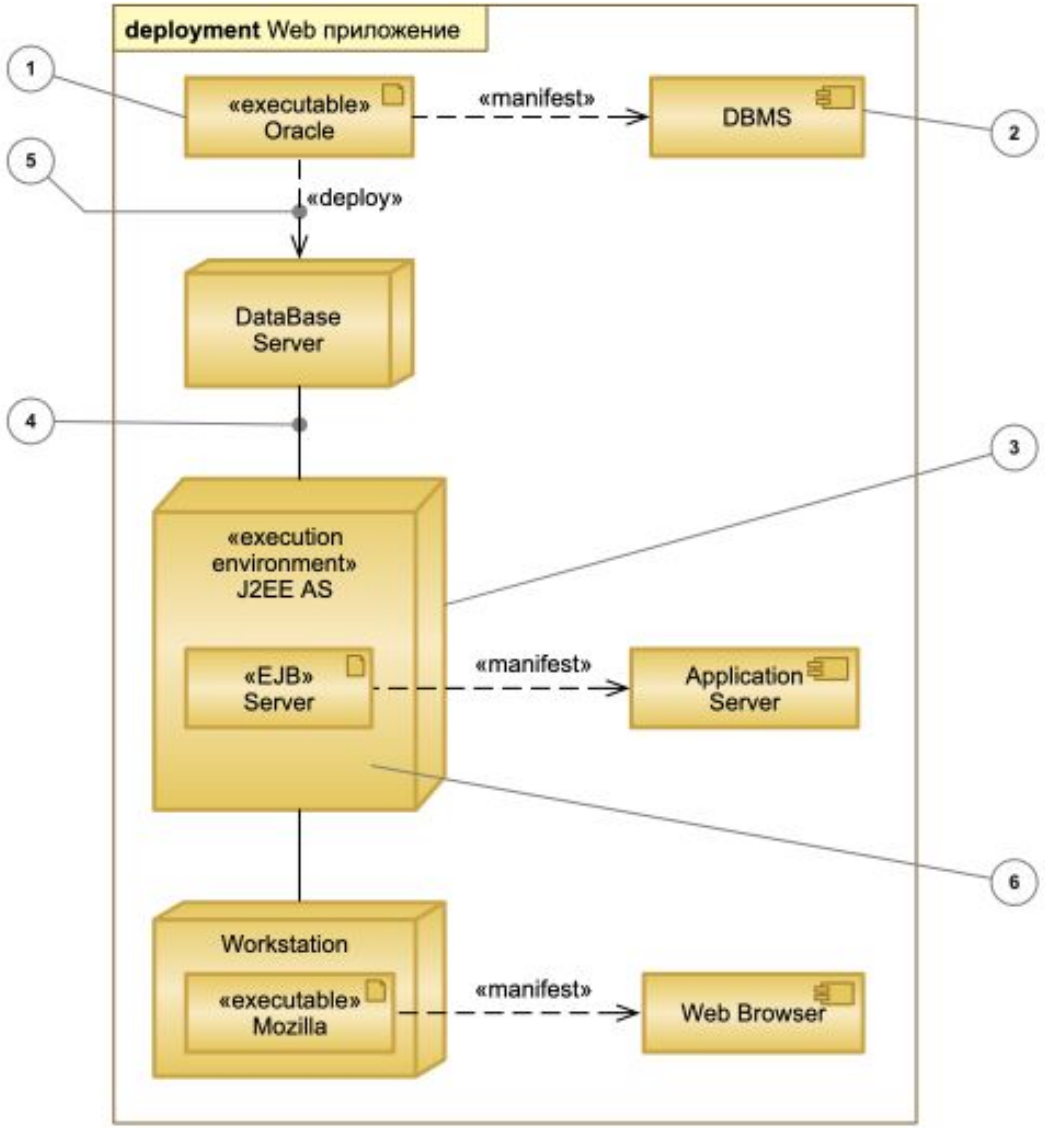
Диаграмма компонентов (component diagram) – показывает взаимосвязи между модулями (логическими или физическими), из которых состоит моделируемая система.



Компоненты – 1;
Интерфейсы – 2;
Зависимости между компонентами – 3.

Диаграмма размещения

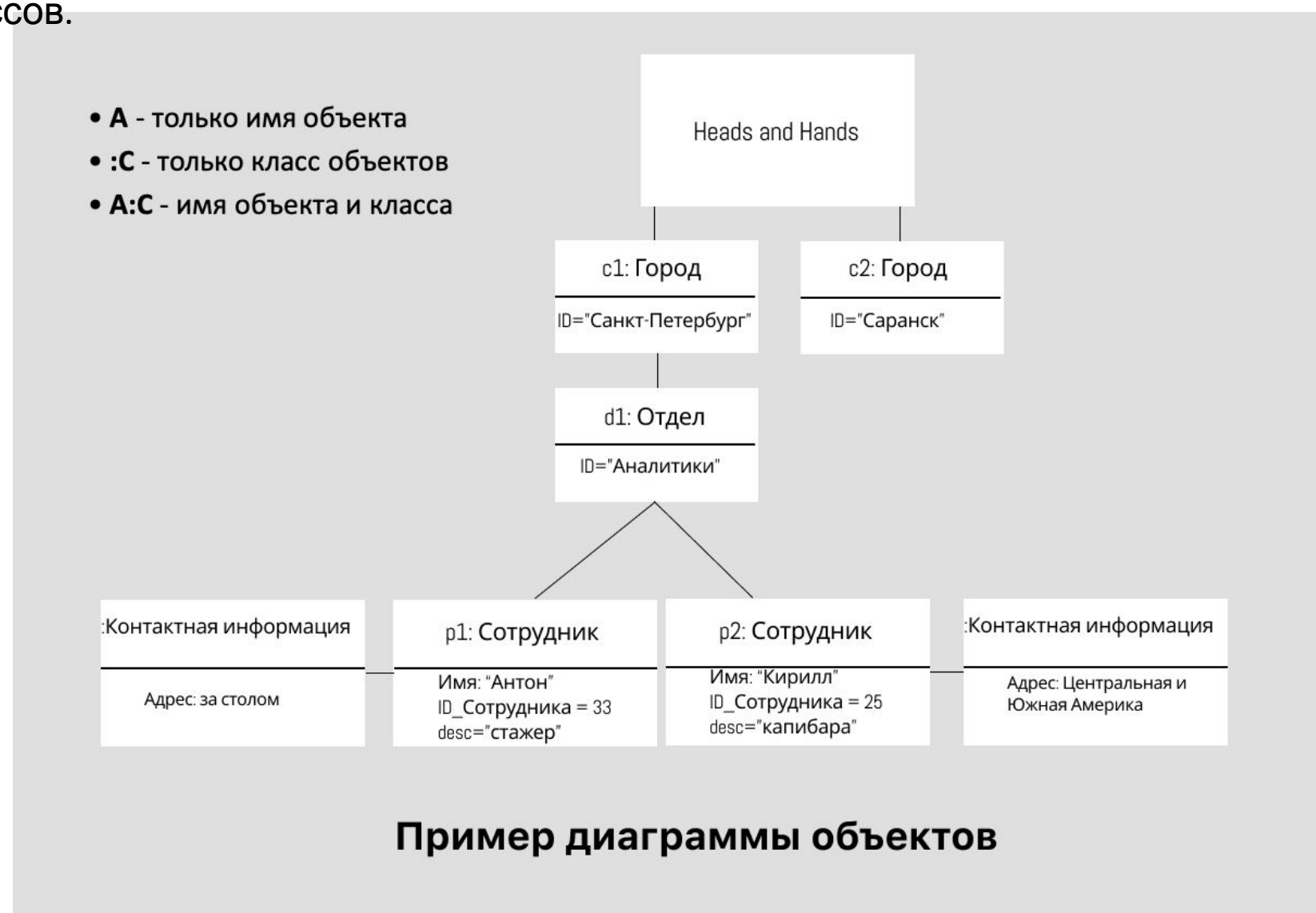
Диаграмма размещения (deployment diagram) наряду с отображением состава и связей элементов системы показывает, как они физически размещены на вычислительных ресурсах во время выг



Артефакт – 1;
Компонент – 2;
Узел – 3;
Отношение ассоциации между узлами – 4;
Отношение зависимости – 5;
Сущность в сущности – 6.

Диаграмма объектов (object diagram)

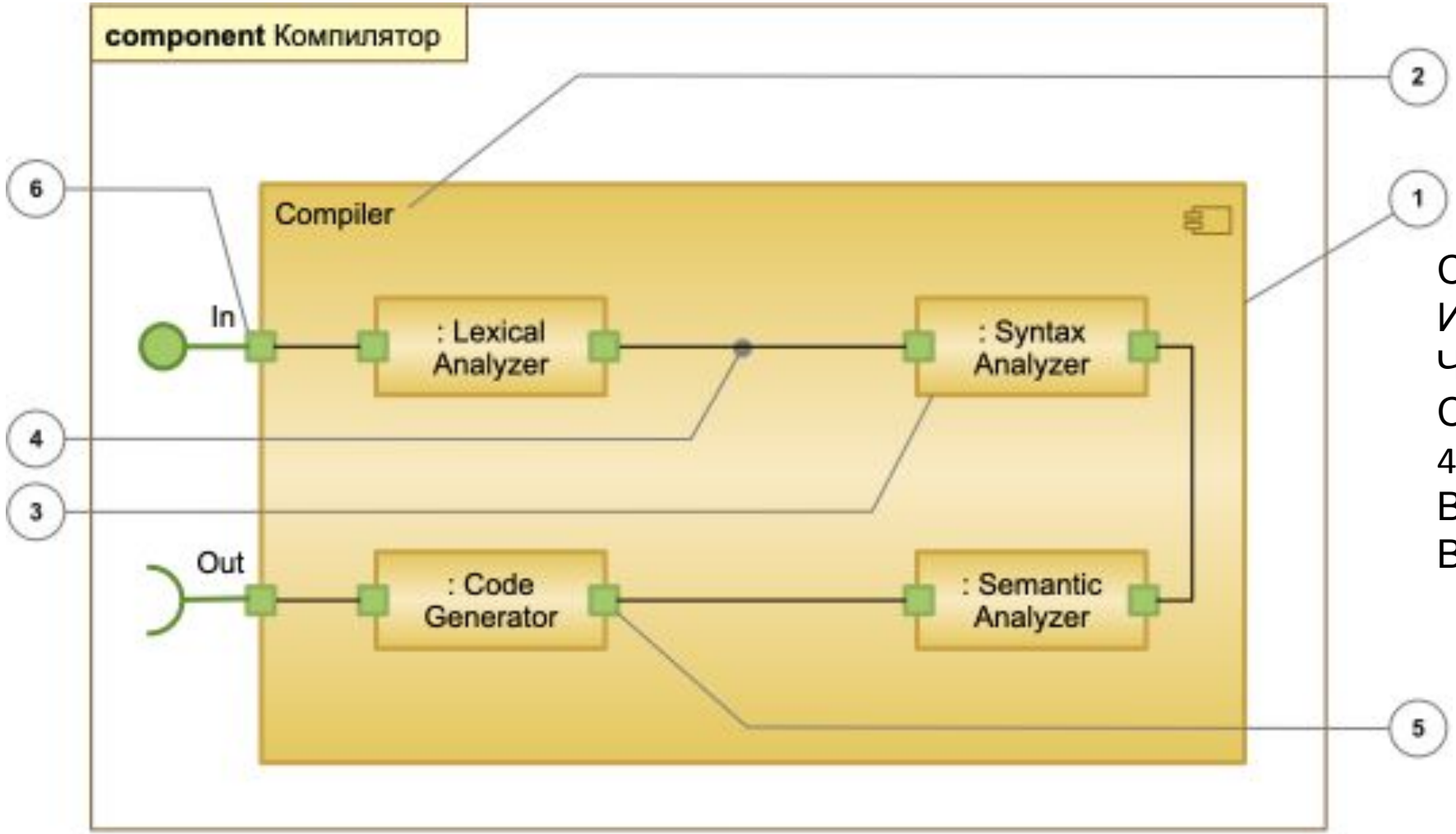
Диаграмма объектов (object diagram) – является экземпляром диаграммы классов.



Пример диаграммы объектов

Диаграмма внутренней структуры

Диаграмма внутренней структуры (composite structure diagram) используется для более подробного представления структурных классификаторов, прежде всего классов и компонентов.



Структурный классификатор – 1;
Имя классификатора – 2;
Части – 3;
Соединители различных типов – 4;
Внутренние порты – 5;
Внешние порты – 6.

Обзорная диаграмма взаимодействия

Обзорная диаграмма взаимодействия (interaction overview diagram) является разновидностью диаграммы деятельности с расширенным синтаксисом: в качестве элементов обзорной диаграммы взаимодействия могут выступать ссылки на взаимодействия (interaction use) - 1 (см. ниже), определяемые диаграммами последовательности

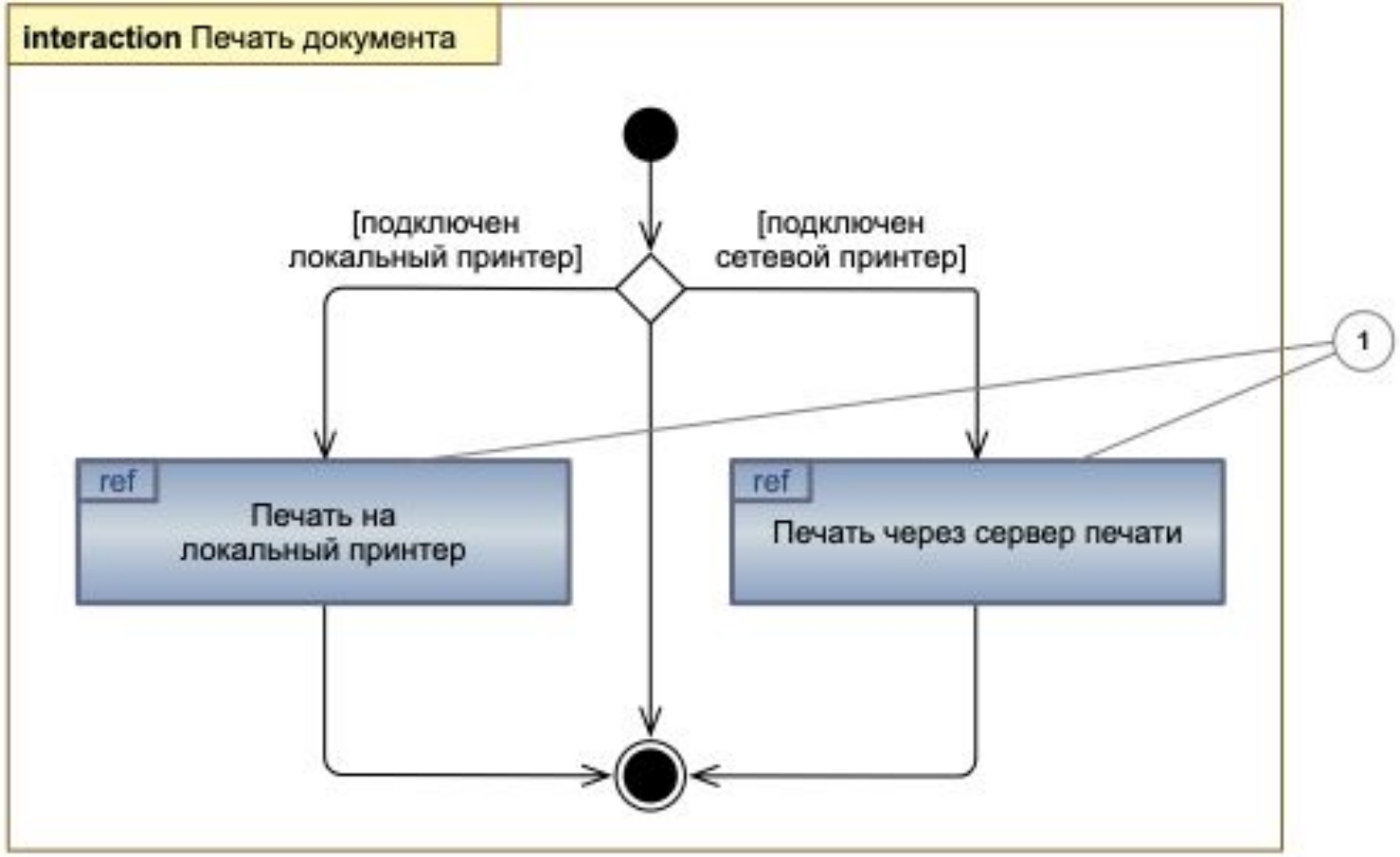
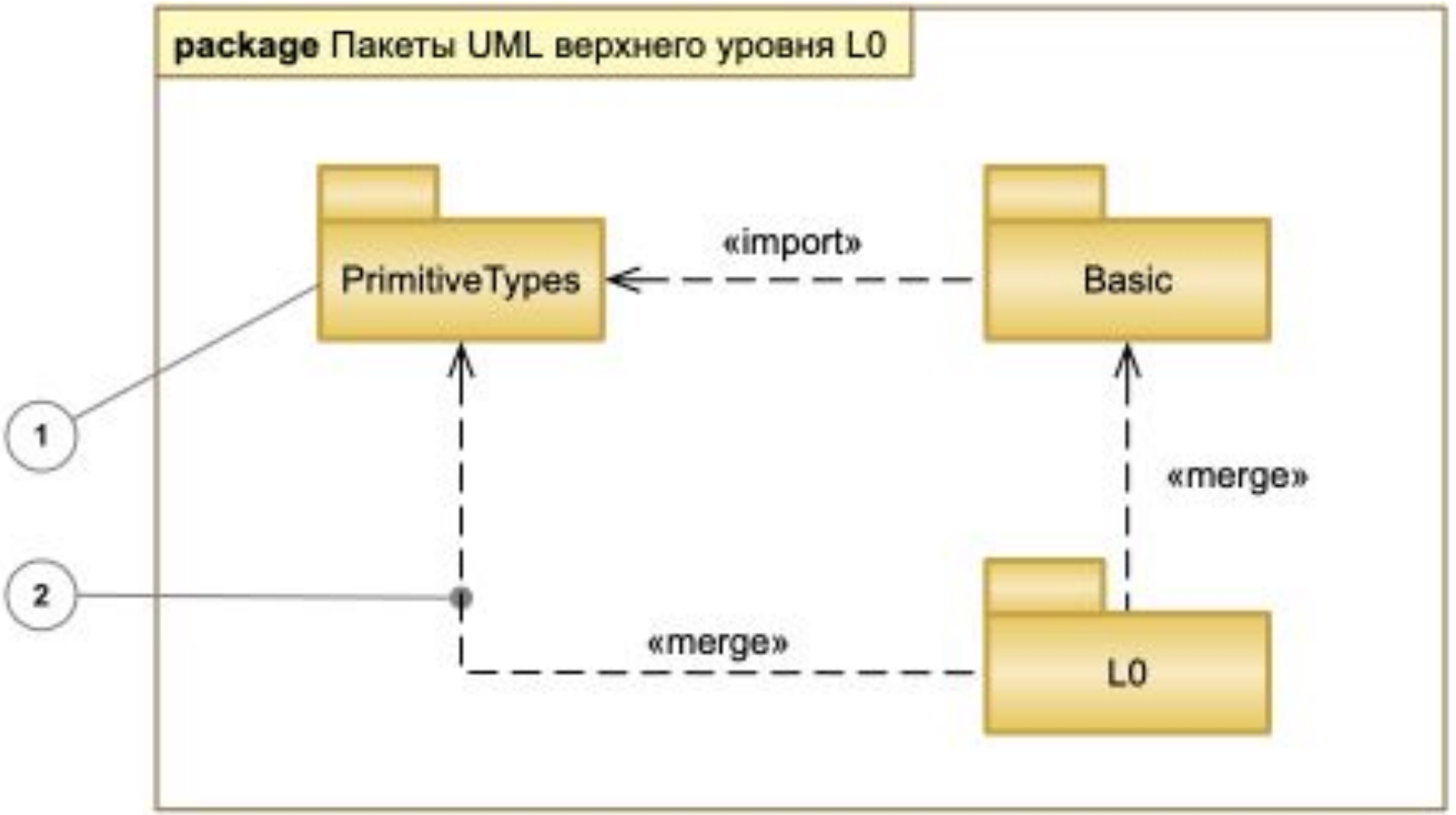


Диаграмма синхронизации

Диаграмма синхронизации (timing diagram) представляет собой особую форму диаграммы последовательности, на которой особое внимание уделяется изменению состояний - 1 (см. ниже) различных экземпляров классификаторов и их временной синхронизации 2 (см. ниже).

Диаграмма пакетов

Диаграмма пакетов (package diagram) – средство группирования элементов модели.



Пакеты – 1;
Стереотипы (связи) – 2.

Выводы

- Таким образом, UML – это графический язык моделирования общего назначения, имеющий нотацию, семантику и прагматику, регулируемую международными стандартами.
- UML позволяет строить описательные модели систем, в том числе программных систем, любой сложности.
- Модель системы может быть визуализирована в форме графических диаграмм, показывающих сущности моделируемой системы и связи между ними.
- В случае необходимости элементы UML могут быть расширены и переопределены средствами самого языка.

BPMN и UML в паре?

BPMN можно сравнить с паровозом, который тянет за собой вагоны UML, а это подтверждает тезис о том, что совместное использование **BPMN и UML** наиболее рационально.

- Анализ бизнес-процессов на этапе предпроектного обследования и их графическое представление целесообразно выполнять в графической нотации BPMN.
- Определять как и кем использовать будущую информационную систему и ее функциональность лучше с помощью элементов языка моделирования UML, в частности диаграммы вариантов использования.

Бизнес анализ и моделирование

- Бизнес – систематическая деятельность
- Организация – ведет бизнес
- Бизнес-процесс – последовательность действий, в результате которой происходит выполнение некоторой функции
- Бизнес-модель – конструктивное описание бизнес-процессов
- Бизнес-анализ – процесс построения бизнес-модели