Диаграмма компонентов

- Диаграмма компонентов описывает особенности физического представления системы.
- Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код.
- Во многих средах разработки модуль или компонент соответствует файлу.
- Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ.
- Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

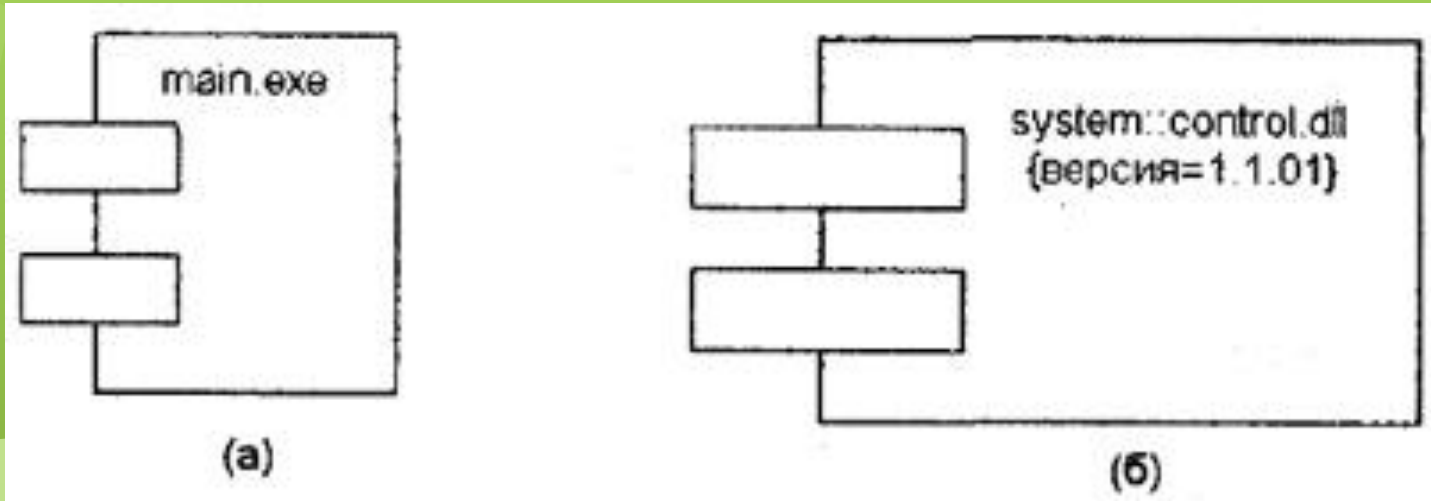
Диаграмма компонентов разрабатывается для следующих целей:

1. Визуализации общей структуры исходного кода программной системы.
2. Спецификации исполнимого варианта программной системы.
3. Обеспечения многократного использования отдельных фрагментов программного кода.
4. Представления концептуальной и физической схем баз данных.

Компоненты

- Для представления физических сущностей в языке UML применяется специальный термин — компонент.
- Для графического представления компонента может использоваться специальный символ — прямоугольник со вставленными слева двумя более мелкими прямоугольниками.
- Внутри объемлющего прямоугольника записывается имя компонента и, возможно, некоторая дополнительная информация.

Графическое изображение компонента в языке UML



Имя компонента

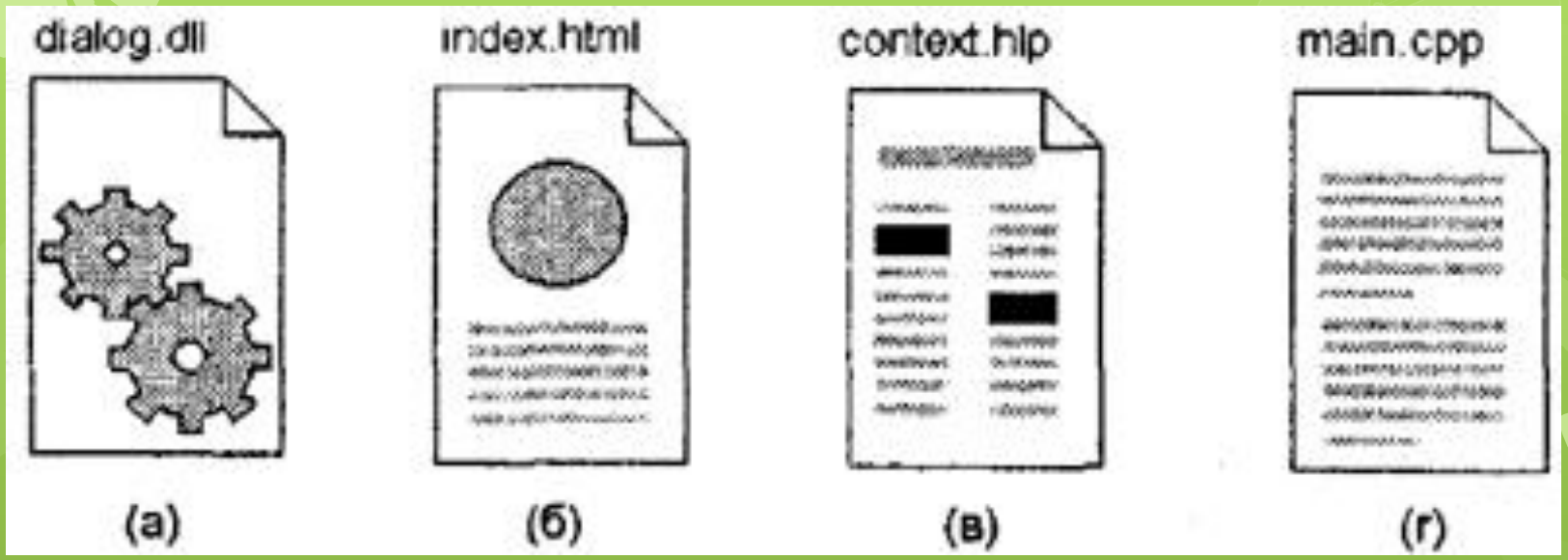
- Имя компонента подчиняется общим правилам именования элементов модели в языке UML и может состоять из любого числа букв, цифр и некоторых знаков препинания.
- Отдельный компонент может быть представлен на уровне типа или на уровне экземпляра.
- Хотя его графическое изображение в обоих случаях одинаковое, правила записи имени компонента несколько отличаются.
- Если компонент представляется на уровне типа, то в качестве его имени записывается только имя типа с заглавной буквы.
- Если же компонент представляется на уровне экземпляра, то в качестве его имени записывается <имя компонента ':' имя типа X.
- При этом вся строка имени подчеркивается.
- В качестве простых имен принято использовать имена исполняемых файлов, имена динамических библиотек, имена текстовых файлов или файлов справки, имена файлов баз данных или имена файлов с исходными.

- В отдельных случаях к простому имени компонента может быть добавлена информация об имени объемлющего пакета и о конкретной версии реализации данного компонента.
- В этом случае номер версии записывается как помеченное значение в фигурных скобках.
- В других случаях символ компонента может быть разделен на секции, чтобы явно указать имена реализованных в нем интерфейсов.
- Такое обозначение компонента называется расширенным.

Виды компонентов

1. Компоненты развертывания, которые обеспечивают непосредственное выполнение системой своих функций. Такими компонентами могут быть динамически подключаемые библиотеки с расширением dll, Web-страницы на языке разметки гипертекста с расширением html.
2. Компоненты-рабочие продукты. Как правило — это файлы с исходными текстами программ, например, с расширениями h или cpp для языка C++.
3. Компоненты исполнения, представляющие исполнимые модули — файлы с расширением exe. Они обозначаются обычным образом.

Варианты графического изображения компонентов на диаграмме компонентов



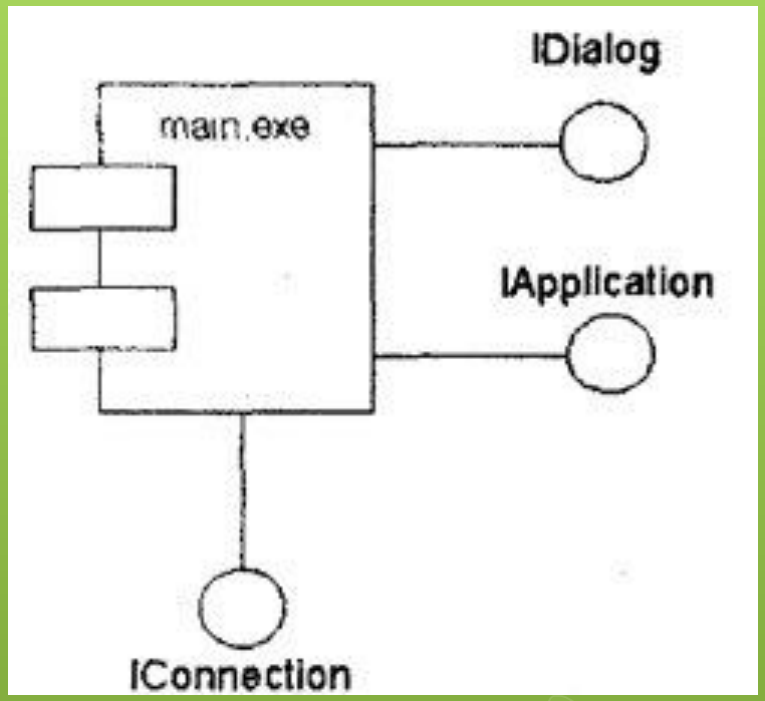
- В языке UML для компонентов определены следующие стереотипы:

1. Библиотека (library) — определяет первую разновидность компонента, который представляется в форме динамической или статической библиотеки.
2. Таблица (table) — также определяет первую разновидность компонента, который представляется в форме таблицы базы данных.
3. Файл (file) — определяет вторую разновидность компонента, который представляется в виде файлов с исходными текстами программ.
4. Документ (document) — определяет вторую разновидность компонента, который представляется в форме документа.
5. Исполнимый (executable) — определяет третий вид компонента, который может исполняться в узле.

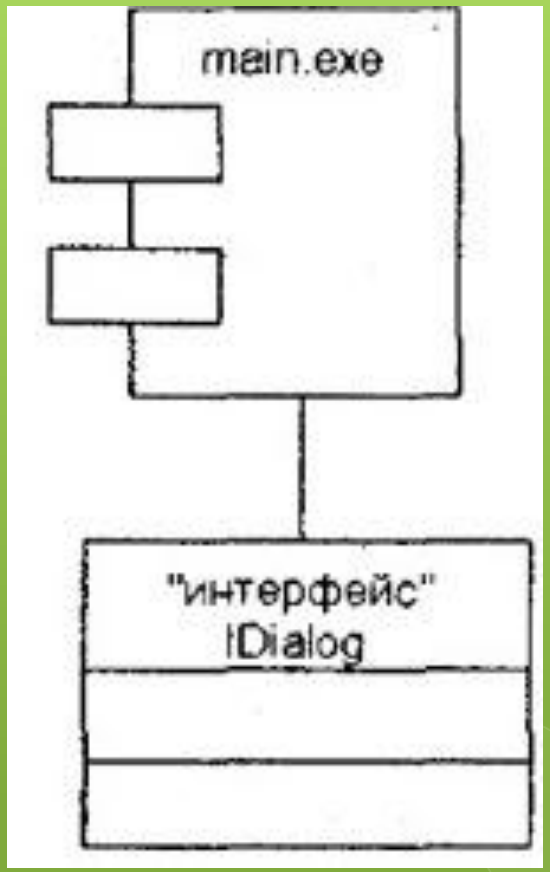
Интерфейсы

- Следующим элементом диаграммы компонентов являются интерфейсы.
- Интерфейс графически изображается окружностью, которая соединяется с компонентом отрезком линии без стрелок.
- При этом имя интерфейса, которое обязательно должно начинаться с заглавной буквы "I", записывается рядом с окружностью.
- Семантически линия означает реализацию интерфейса, а наличие интерфейсов у компонента означает, что данный компонент реализует соответствующий набор интерфейсов.

Графическое изображение интерфейсов на диаграмме компонентов



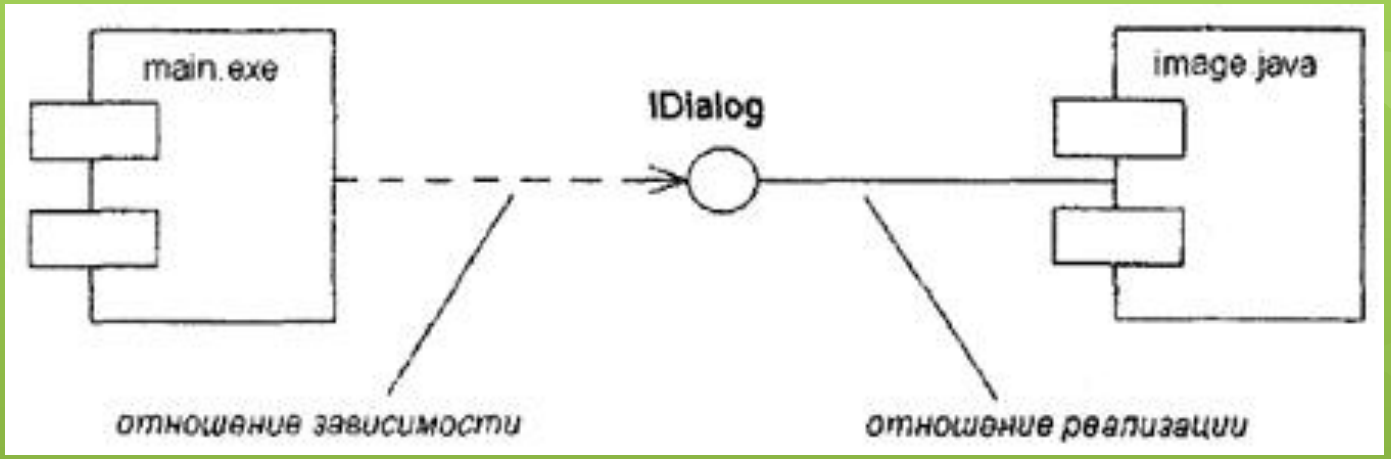
Другим способом представления интерфейса на диаграмме компонентов является его изображение в виде прямоугольника класса со стереотипом "интерфейс" и возможными секциями атрибутов и операций.



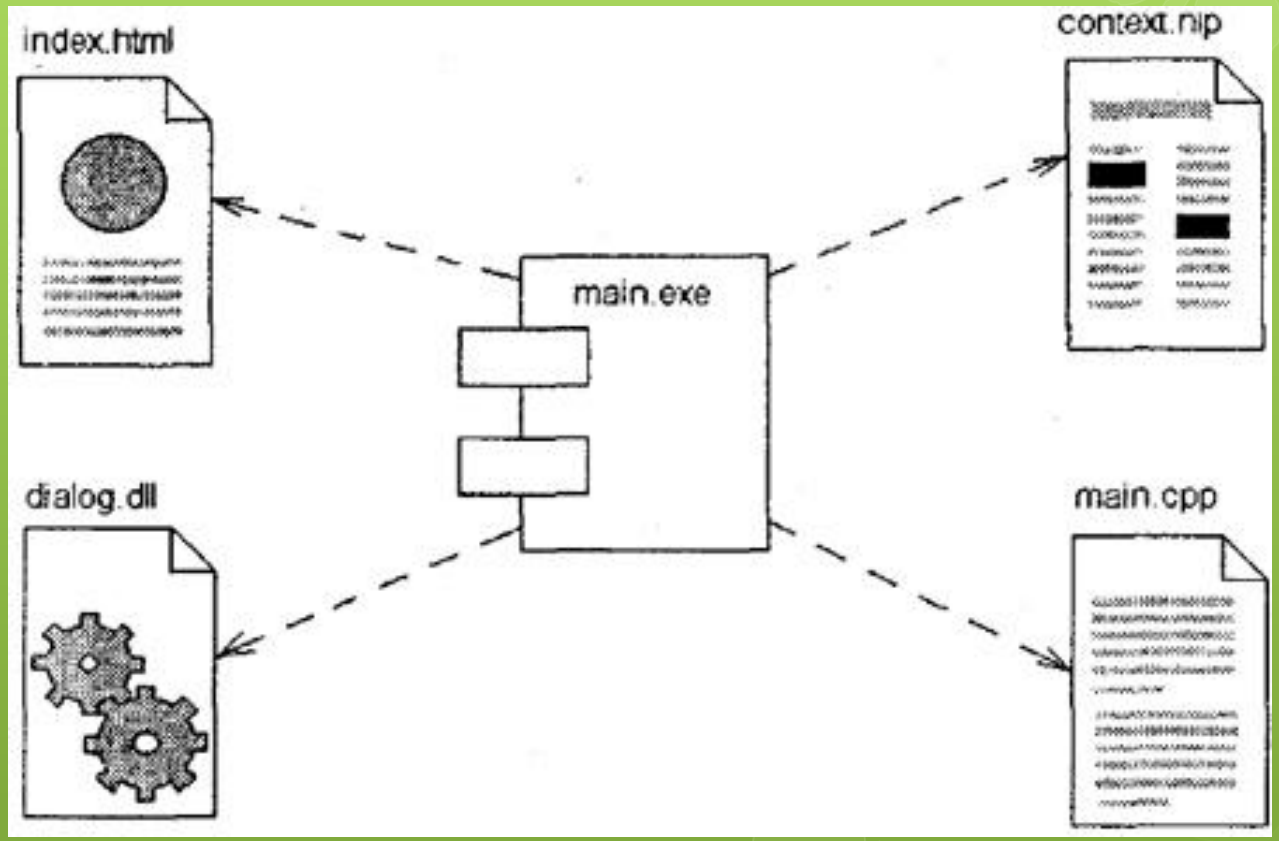
Зависимости

- Отношение зависимости на диаграмме компонентов изображается пунктирной линией со стрелкой, направленной от клиента (зависимого элемента) к источнику (независимому элементу).
- Применительно к диаграмме компонентов зависимости могут связывать компоненты и импортируемые этим компонентом интерфейсы, а также различные виды компонентов между собой.

Фрагмент диаграммы компонентов с отношением зависимости

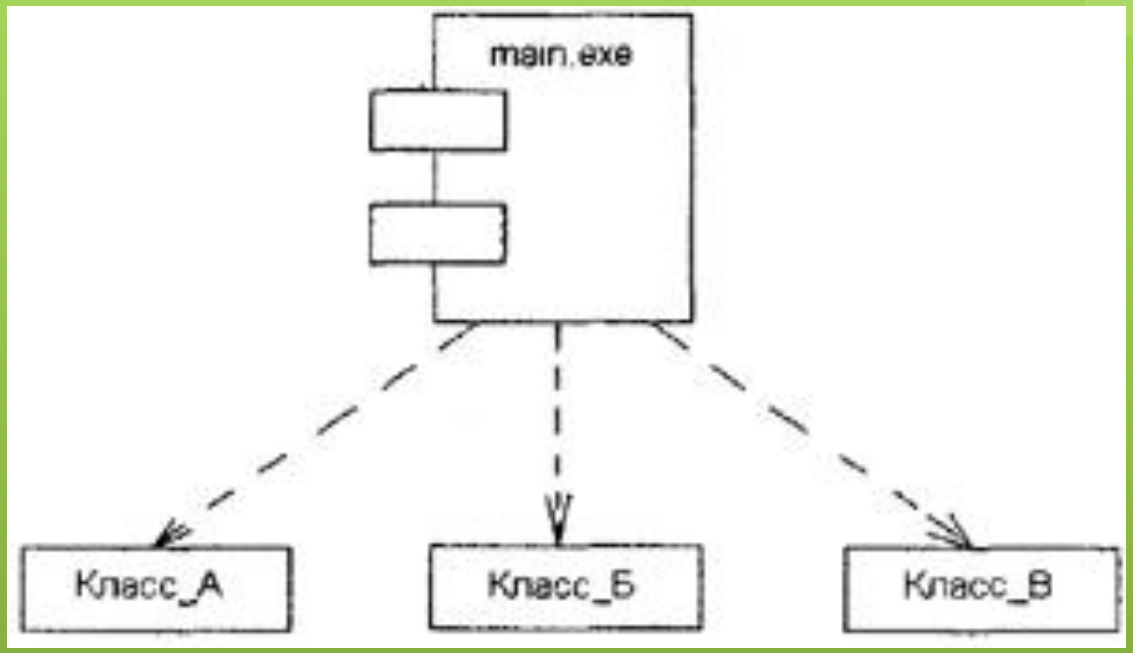


Другим случаем отношения зависимости на диаграмме компонентов является отношение между различными видами компонентов.



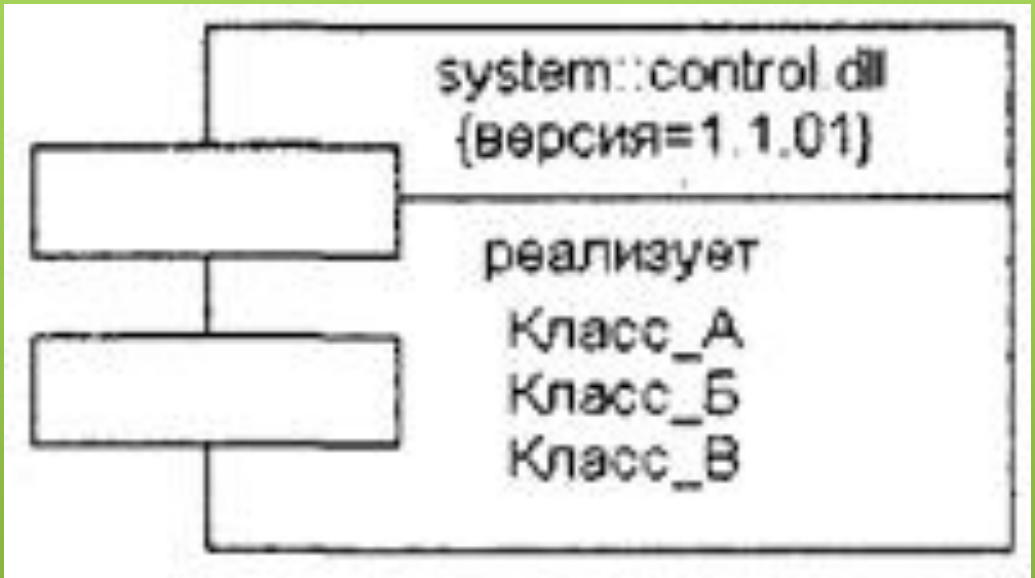
- На диаграмме компонентов могут быть представлены отношения зависимости между компонентами и реализованными в них классами.
- Эта информация имеет важное значение для обеспечения согласования логического и физического представлений модели системы.
- Изменения в структуре описаний классов могут привести к изменению компонента.

Графическое изображение зависимости между компонентом и классами



- Если требуется подчеркнуть, что некоторый компонент реализует отдельные классы, то для обозначения компонента используется расширенный символ прямоугольника.
- При этом прямоугольник компонента делится на две секции горизонтальной линией.
- Верхняя секция служит для записи имени компонента, а нижняя секция — для указания дополнительной информации.

Графическое изображение компонента с дополнительной информацией о реализуемых им классах



- Внутри символа компонента могут изображаться другие элементы графической нотации, такие как классы (компонент уровня типа) или объекты (компонент уровня экземпляра).
- В этом случае символ компонента изображается таким образом, чтобы вместить эти дополнительные символы.
- Изображенный ниже компонент является экземпляром и реализует три отдельных объекта.

Графическое изображение компонента уровня экземпляра, реализующего отдельные объекты

Рекомендации по построению диаграммы компонентов

- Разработка диаграммы компонентов предполагает использование информации как о логическом представлении модели системы, так и об особенностях ее физической реализации.
- До начала разработки необходимо принять решения о выборе вычислительных платформ и операционных систем, на которых предполагается реализовывать систему, а также о выборе конкретных баз данных и языков программирования.
- После этого можно приступать к общей структуризации диаграммы компонентов.
- Необходимо дополнить модель интерфейсами и схемами базы данных.
- Завершающий этап построения диаграммы компонентов связан с установлением и нанесением на диаграмму взаимосвязей между компонентами, а также отношений реализации.
- Эти отношения должны иллюстрировать все важнейшие аспекты физической реализации системы, начиная с особенностей компиляции исходных текстов программ и заканчивая исполнением отдельных частей программы на этапе ее выполнения.
- Следует обратить внимание, что диаграмма компонентов, как правило, разрабатывается совместно с диаграммой развертывания, на которой представляется информация о физическом размещении компонентов программной системы по ее отдельным узлам.