

**UNIFIED  
MODELING  
LANGUAGE™**

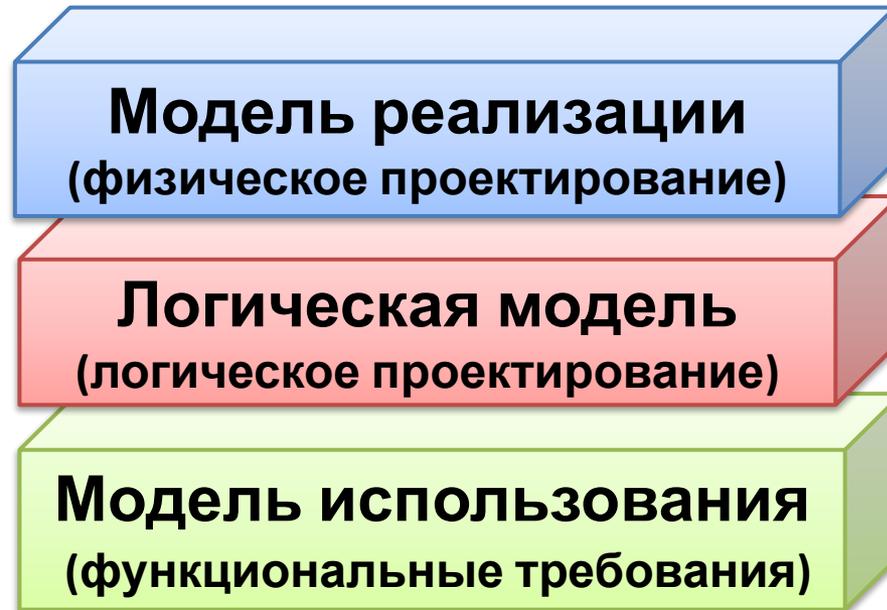


# ОБЪЕКТНО- ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ ИС (ФИЗИЧЕСКАЯ МОДЕЛЬ)

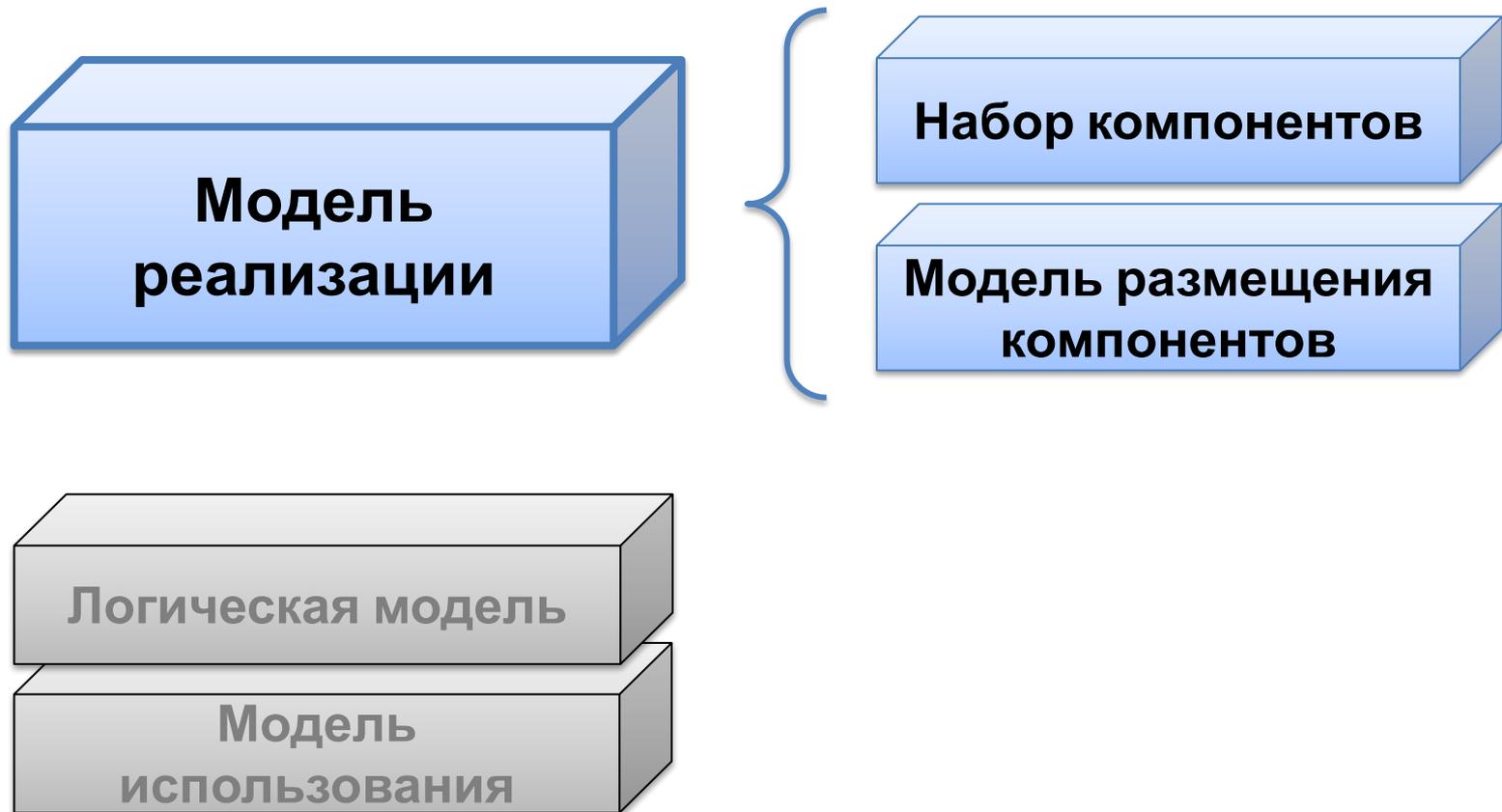
д.т.н. Емельянов Виталий Александрович

✉ : [v.yemelyanov@gmail.com](mailto:v.yemelyanov@gmail.com)

# Объектно-ориентированное проектирование ИС

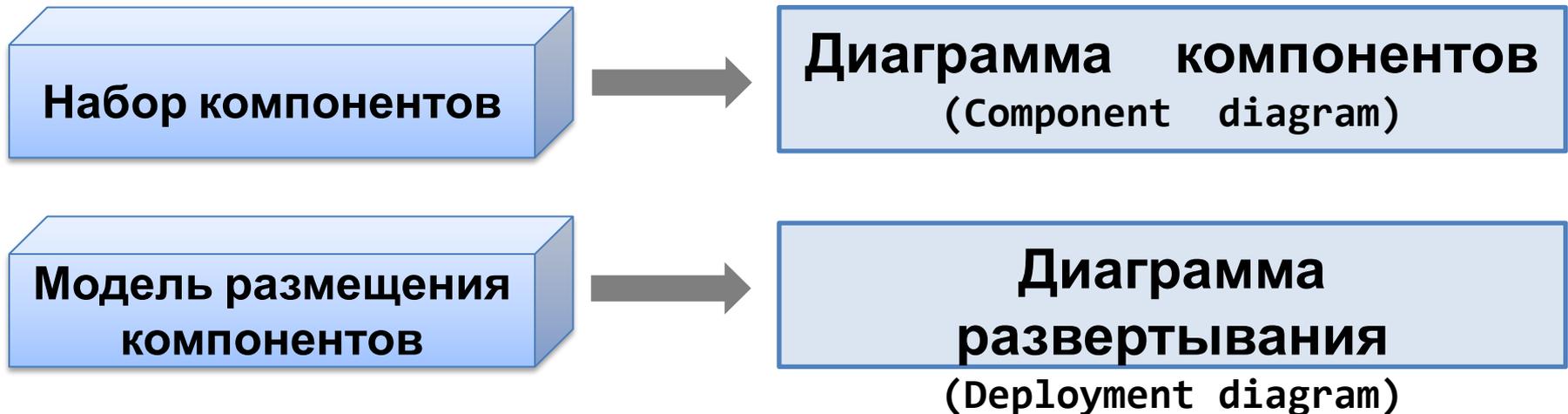


# Объектно-ориентированное проектирование ИС



# Физическое проектирование

**Физическое проектирование** предназначено для спецификации программных компонентов и узлов среды исполнения, на которых они размещаются (развертываются) и выполняются



# Диаграммы компонентов (Component diagram)

## Назначение:

- визуализация общей организации структуры исходного кода программы;
- спецификация исполнимого варианта программной системы;
- представление концептуальной и физической схем баз данных.

## Элементы диаграммы:

- **Компоненты**
- **Интерфейсы**
- **Отношения зависимости, обобщения, ассоциации и реализации**

# Диаграммы компонентов (Component diagram)

## Сущность «Компонент»:

- Служит для обозначения элементов физического представления модели и может реализовывать некий набор интерфейсов

## Условное графическое обозначение:



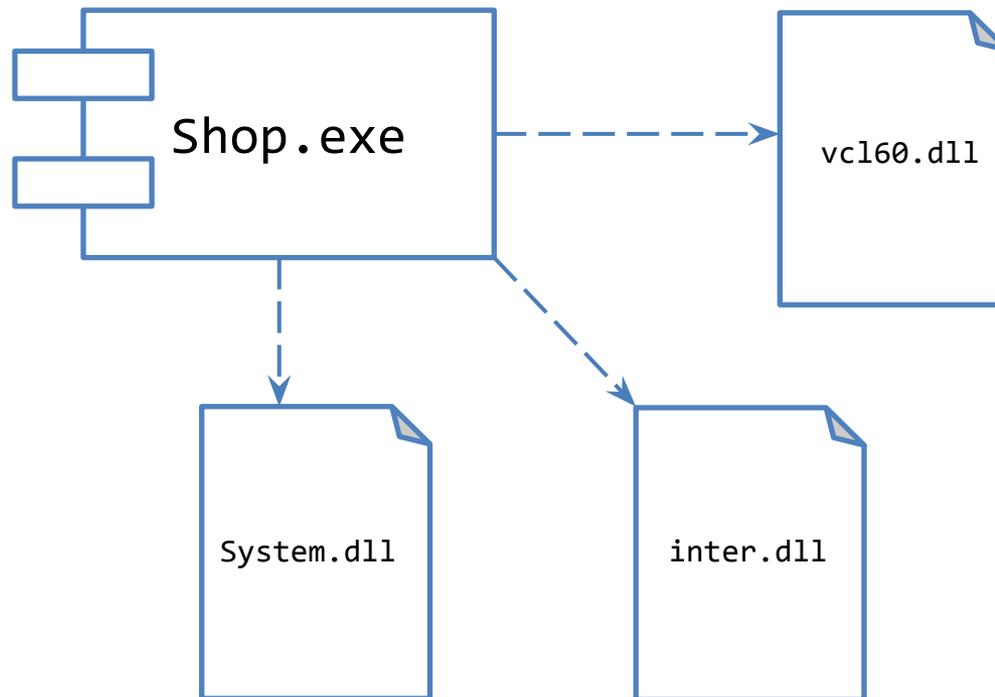
# Диаграммы компонентов (Component diagram)

## Виды компонентов:

- ▣ **компоненты развертывания (Deployment components)** – необходимы для построения исполняемой системы. К их числу относятся динамически подключаемые библиотеки (DLL) и исполняемые программы (EXE), динамические Web-страницы и т.д.
- ▣ **компоненты - рабочие продукты (Work product components)**. Представляют собой побочный результат процесса разработки. К этим компонентам относятся файлы с исходными текстами программ и данными, из которых создаются компоненты развертывания.
- ▣ **компоненты исполнения (Execution components)**. Они создаются как следствие работы системы.

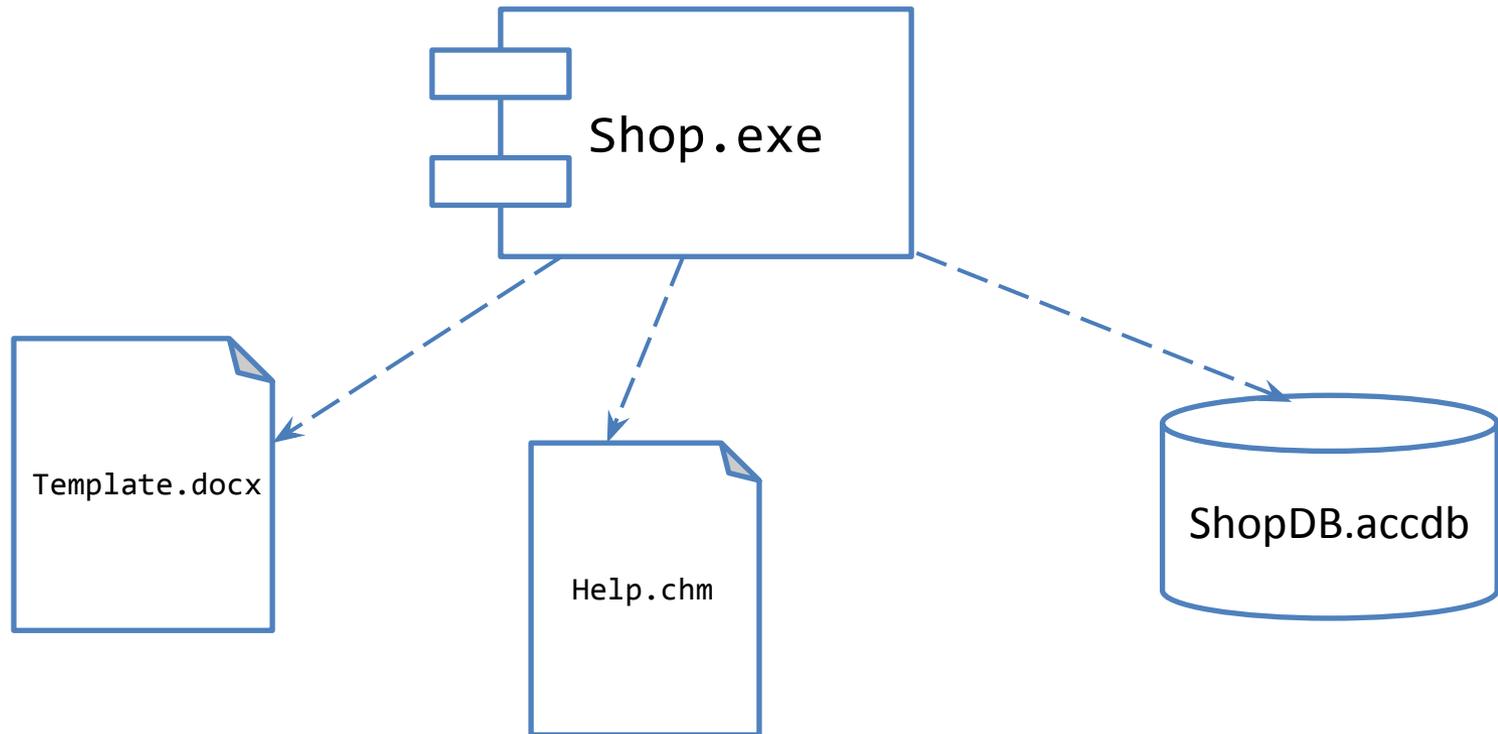
# Диаграммы компонентов (Component diagram)

## Моделирование исполняемых программ и библиотек:



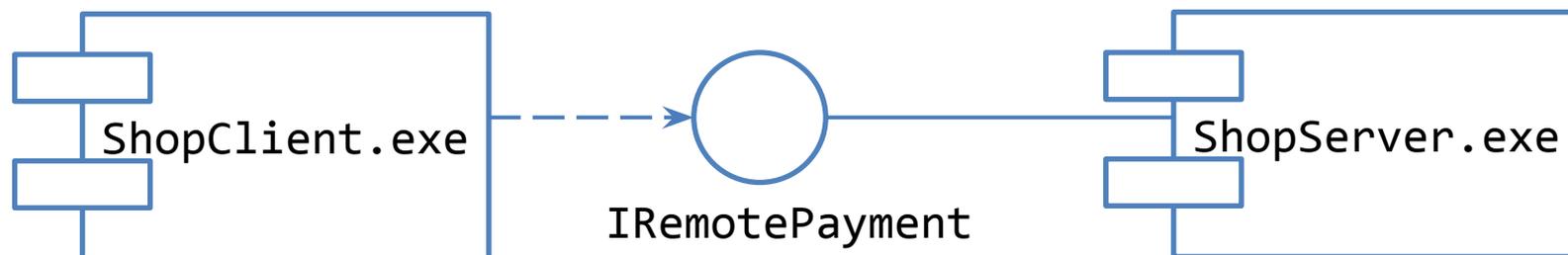
# Диаграммы компонентов (Component diagram)

## Моделирование БД, файлов и документов:



# Диаграммы компонентов (Component diagram)

## Моделирование интерфейсов:



# Диаграммы развертывания (Deployment diagram)

## Цели построения диаграммы:

- указать размещение исполнимых компонентов программной системы по отдельным физическим узлам;
- показать физические связи между всеми узлами реализации системы на этапе ее исполнения;
- выявить узкие места системы и реконфигурировать ее топологию для достижения наилучшей производительности.

## Элементы диаграммы:

- **Узлы**
- **Отношения зависимости и ассоциации**

# Диаграммы развертывания (Deployment diagram)

## Сущность «Узел»:

- представляет собой некоторый физически существующий элемент системы, обладающий некоторым вычислительным ресурсом.
- Диаграмма размещения показывает наличие физических соединений – маршрутов передачи информации между аппаратными устройствами, задействованными в реализации системы.

## Условное графическое обозначение:



# Диаграммы развертывания (Deployment diagram)

## Стереотипы узлов:

**<<processor>>** – активный узел. Узел, способный выполнять размещенные на нем компоненты (сервер, рабочая станция, локальный компьютер)

**<<device>>** – пассивный узел. Узел, на котором могут быть размещены компоненты (принтер, сканер, табло коллективного пользования)

# Диаграммы развертывания (Deployment diagram)

## Стереотипы узлов:

**<<net>>** – сеть (компьютерная сеть, промышленная и т.д.)

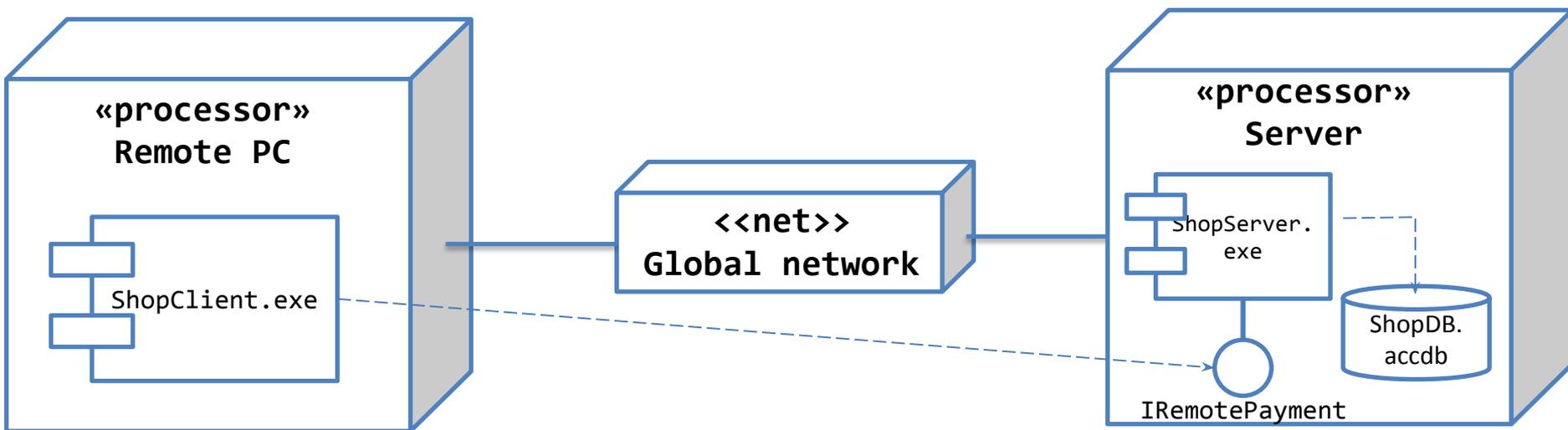
**<<sensor>>** – датчик

**<<mobile device>>** – мобильное устройство

**<<server application>>** – сервер приложений

**<<client workstation>>** – клиентская рабочая станция

# Диаграммы развертывания (Deployment diagram)



# CASE-средства для проектирования

## Встроенные

Часть интегрированной среды разработки  
(**Средства разработки диаграмм в составе Microsoft Visual Studio**)

## Встраиваемые

После инсталляции внедряются в интегрированную среду разработки и становятся частью этой среды.

**Rational XDE (внедрение в Microsoft Visual Studio)**

## Автономные

После инсталляции существуют как самостоятельные приложения

- **MS Visio**: пакет деловой графики общего назначения
- **StarUML**: специализированный инструмент ООП
- **RSA (Rational Software Architect)**: специализированный инструмент ООП, ориентированный на коллективную разработку

# Сравнение возможностей CASE-средств

	Visio	XDE	StarUML	RSA
Контроль согласованности диаграмм	-	+	+	+
Прямая трансформация (модель->код)	-	+	+	+
Обратная трансформация (код->модель)	-	+	+	+
Диаграммы для трансформации	-	ДК	ДК	ДК; ДП
Разделение модели между разработчиками	-	-	-	+
Объединение моделей	-	-	-	+
Импорт требований из других нотаций в UML	-	-	-	+
Экспорт требований из UML в другие нотации	-	-	-	+