



*Язык UML*

# *Диаграммы КОМПОНЕНТОВ*







# Преимущества UML

- ✓ UML объектно-ориентированный, в результате чего методы описания результатов анализа и проектирования семантически близки к методам программирования на современных объектно-ориентированных языках;
- ✓ UML позволяет описать систему практически со всех возможных точек зрения и разные аспекты поведения системы;
- ✓ Диаграммы UML сравнительно просты для чтения после достаточно быстрого ознакомления с его синтаксисом;
- ✓ UML расширяет и позволяет вводить собственные текстовые и графические стереотипы, что способствует его применению не только в сфере программной инженерии;
- ✓ UML получил широкое распространение и динамично развивается.



# Компонентные диаграммы

Компонентная диаграмма – первая из двух разновидностей диаграмм реализации, моделирующих физические аспекты объектно-ориентированных систем. Компонентная диаграмма показывает организацию набора компонентов и зависимости между компонентами.

Элементами диаграмм являются компоненты и интерфейсы, а также отношения зависимости и реализации. Как и другие диаграммы, компонентные диаграммы могут включать примечания и ограничения. Кроме того, компонентные диаграммы могут содержать пакеты или подсистемы, используемые для группировки элементов модели в крупные фрагменты.

# Компонентные диаграммы

Компонент – физическая и заменяемая часть системы, которая соответствует набору интерфейсов и обеспечивает реализацию этого набора интерфейсов.

Интерфейс – список операций, которые определяют услуги класса или компонента.

Графически компонент изображается как прямоугольник с вкладками, обычно включающий имя.

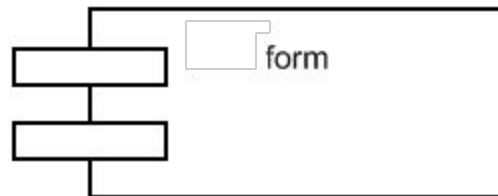


Рис. 1. Обозначение компонента

# Компонентные диаграммы

Компонентные диаграммы используют для моделирования статического представления реализации системы. Это представление поддерживает управление конфигурацией системы, составляемой из компонентов.

Подразумевается, что для получения работающей системы существуют различные способы сборки компонентов.

Компонентные диаграммы показывают отношения:

- ✓ периода компиляции (среди текстовых компонентов);
- ✓ периода сборки, линковки (среди объектных двоичных компонентов);
- ✓ периода выполнения (среди машинных компонентов).