<http://programador.ru/uml-class-diagram-relation/>

## [UML. Диаграммы классов. Отношения](http://programador.ru/uml-class-diagram-relation/)

Мини-справочник по обозначениям которые приняты в UML диаграммах классов для отношений между различными классами.

Обозначаются стрелками.

## Отношения на уровне классов.

### Generalization. Наследование.

[http://programador.ru/wp-content/uploads/2010/03/drawing_generalization.png](http://programador.ru/wp-content/uploads/2010/03/drawing_generalization.png)  
Самое обычное наследование:**class A extends B {    }**

### Implementation. Имплементация.

[http://programador.ru/wp-content/uploads/2010/03/drawing_implementation.png](http://programador.ru/wp-content/uploads/2010/03/drawing_implementation.png)  
Реализация интерфейса: **class A implements I {   }**

## Отношения на уровне объектов

### Association. Ассоциация.

[http://programador.ru/wp-content/uploads/2010/03/class_association.png](http://programador.ru/wp-content/uploads/2010/03/class_association.png)  
Семейство отношения между объектами классов.  "Студент" - "Преподаватель", "Покупатель" - "Продавец" и т.д.  Может обозначаться вообще без стрелки.  
Агрегация и Композиция  -  подтипы ассоциации.

### Aggregation. Агрегация.

[http://programador.ru/wp-content/uploads/2010/03/drawing_aggregation.png](http://programador.ru/wp-content/uploads/2010/03/drawing_aggregation.png)  
Подтип ассоциации. Например один класс содержит (агрегируют) объекты другого класса.

### Composition. Композиция.

[http://programador.ru/wp-content/uploads/2010/03/class_composition.png](http://programador.ru/wp-content/uploads/2010/03/class_composition.png)  
Похоже на агрегацию только более сильная связь. Поэтому закрашенный ромб. Например: если уничтожается композитор, то его объекты классов на которые он ссылается также перестают существовать.

## Просто отношение

### Dependency. Зависимость.

[диаграмма классов зависимость](http://programador.ru/wp-content/uploads/2010/03/class_dependency.png)

Классы "каким либо образом" зависят друг от друга. Например, если у одного класса меняются методы, конструкторы или поля, и поэтому приходится переписывать другой класс, то значит они зависимы.  Одна из самых слабеньких связей.  Например объекты одного класса передаются как параметр в методы другого класса и т.д.

## Полезные ссылки

1. Cтатья на  [wikipedia.](http://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%BE%D0%B2" \t "_blank) Рекомендую прочитать также эту же статью на английской википедии.
2. [Официальный сайт](http://www.uml.org/) UML. Спецификация и т.д.
3. [uml2.ru](http://www.uml2.ru/) - в целом по UML