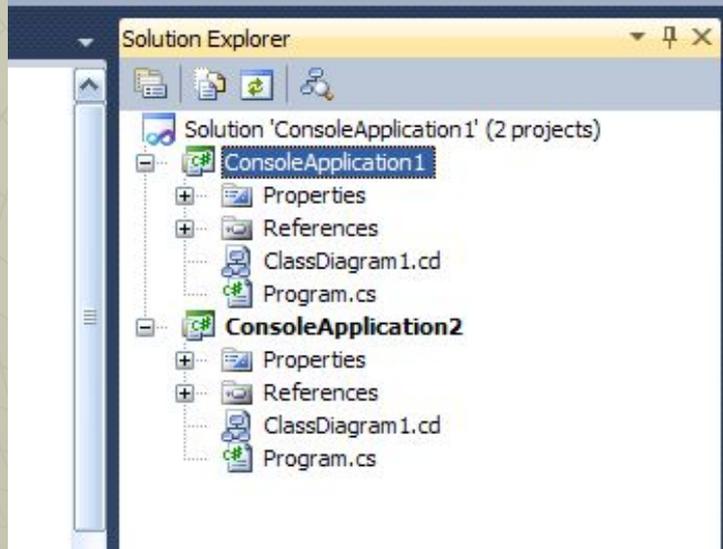




Объектно-ориентированное программирование

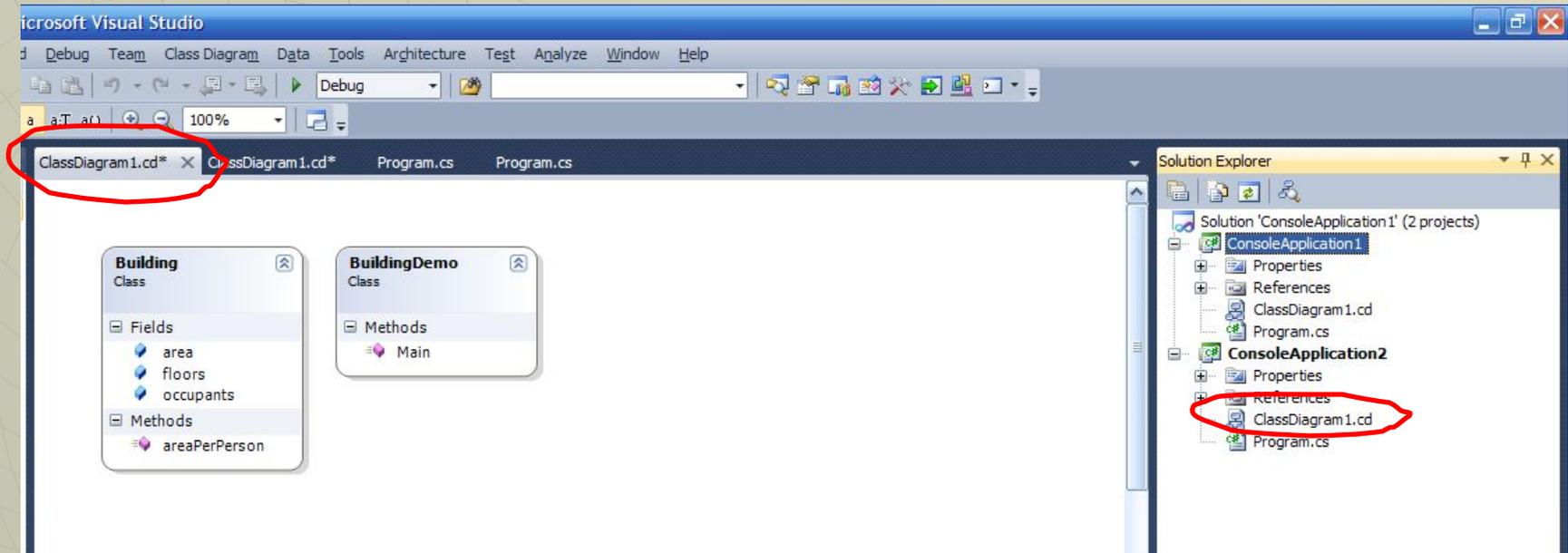
Лекция 5. Диаграммы классов

Class Diagram



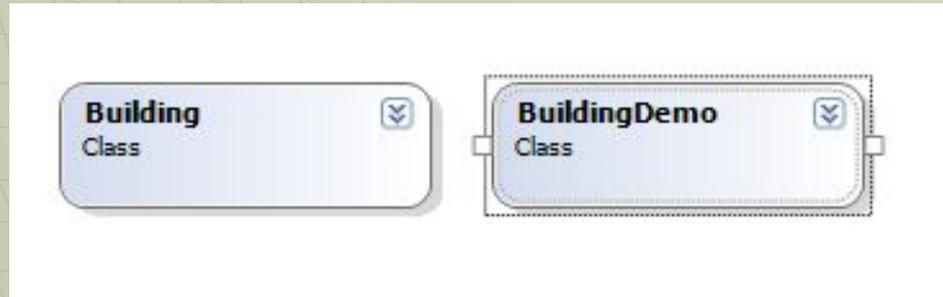
- ◆ Для графического представления членов класса служит окно Class Diagram. (Правая кнопка мыши на имени проекта в Solution Explorer – View Class Diagram)

Class Diagram



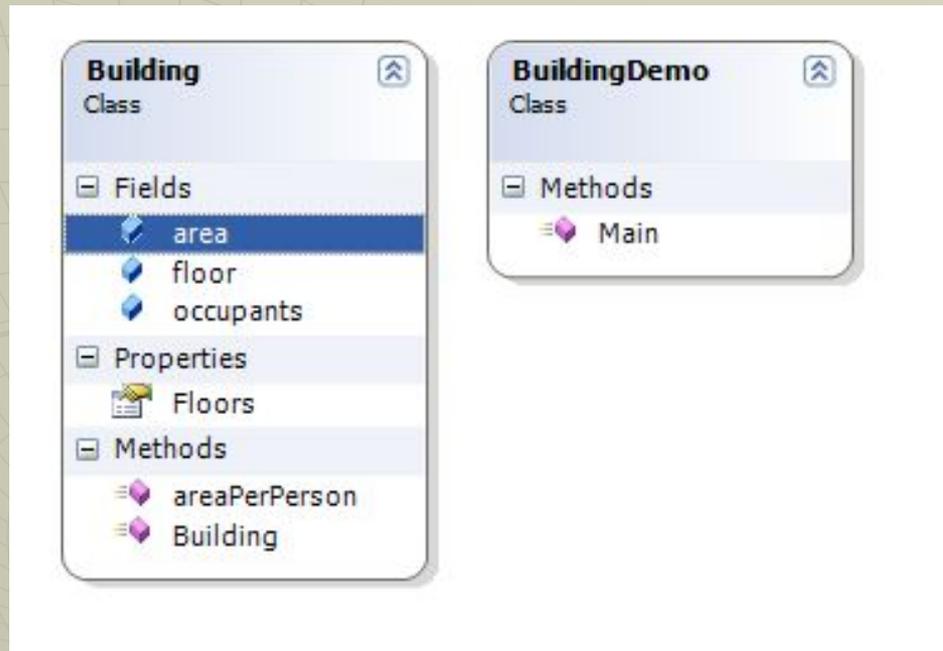
- ◆ ClassDiagram1.cd – добавляется окно и новый объект в Solution Explorer.

Class Diagram



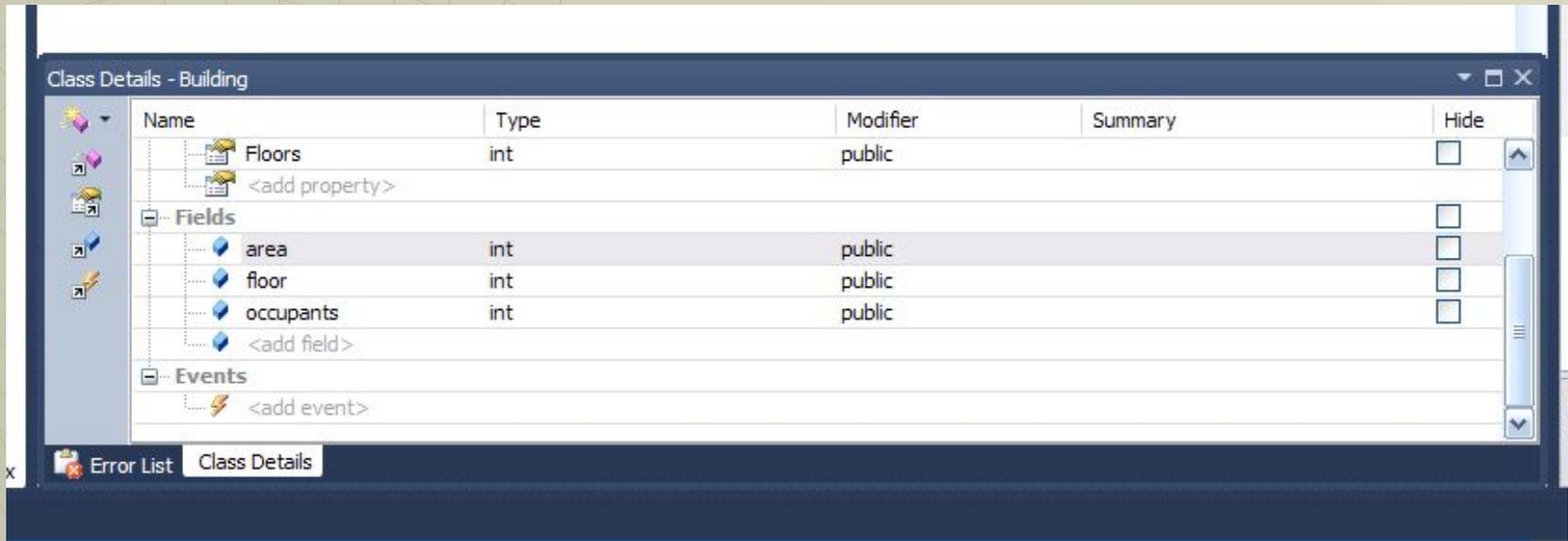
- ◆ Здесь видны все классы текущего проекта

Class Diagram



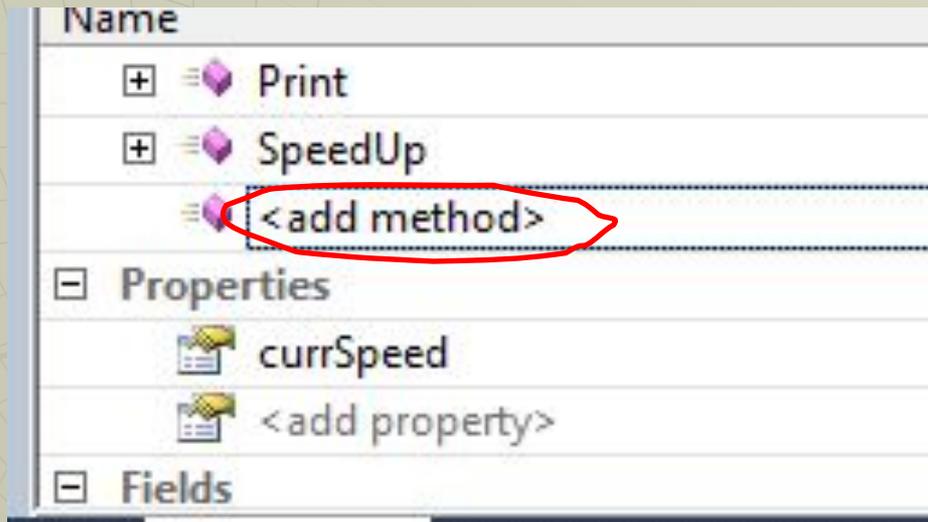
- ◆ Щелчок по двойной стрелке в правом верхнем углу раскрывает класс:

Class Diagram



- ◆ Если какой-либо класс активен:
внизу окно Class Details

Class Diagram



- ◆ Можно добавлять поля, свойства и методы прямо здесь

Class Diagram

- ◆ Если добавить свойство Area, будет сформирована заготовка:

```
public int Area
{
    get
    {
        throw new System.NotImplementedException();
    }
    set
    {
    }
}
```

- ◆ При добавлении нового поля или метода будут вносить изменения в код. Причем поля по умолчанию **int** и **private**, методы **void** и **public**, свойства - **int** и **public**.

Диаграммы классов

- ◆ При проектировании нового класса используют UML-диаграммы.
- ◆ Существует специальный язык программного проектирования UML для построения схем моделирования.

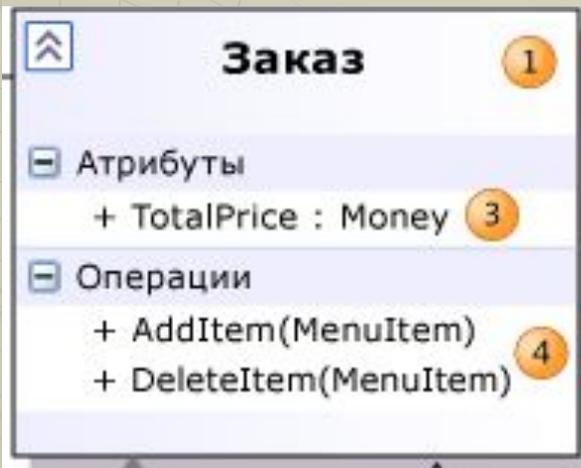
Диаграммы классов

- ◆ Схемы моделирования помогают понять, прояснить и сообщить сведения о коде и пользовательских требованиях, которые должны удовлетворяться программной системой. Например, для описания пользовательских требований и передачи информации о них можно использовать UML-схемы вариантов использования, действий, классов и последовательностей. Для описания функциональных возможностей системы и передачи информации о них можно использовать UML-схемы компонентов, классов, действий и последовательностей.

Варианты схем:

Схема	Показывает
UML-схемы деятельности:	Рабочий процесс между действиями и участниками бизнес-процесса.
Схемы компонентов UML:	Компоненты системы, их интерфейсы, порты и отношения.
UML-схемы классов:	Типы, используемые для хранения данных и обмена данными в системе, и их отношения.
UML-схемы последовательностей:	Последовательности взаимодействий между объектами, компонентами, системами или субъектами.
UML-схемы вариантов использования:	Поддерживаемые системой цели и задачи пользователя.

Основные правила UML- диаграмм классов



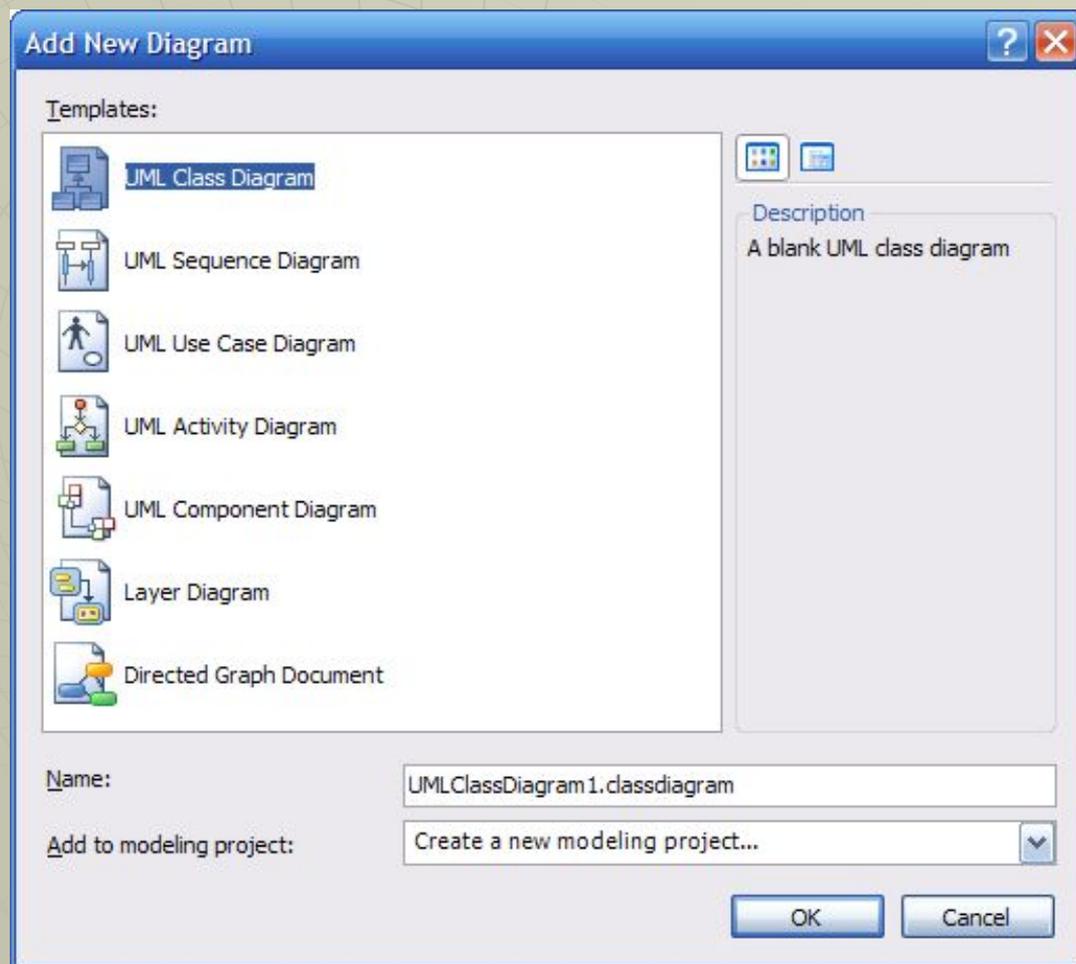
- ◆ Любой класс состоит из атрибутов и операций.
- ◆ Каждый атрибут (поле) имеет какой-либо тип
- ◆ Каждая операция (метод) имеет набор параметров

Диаграммы классов

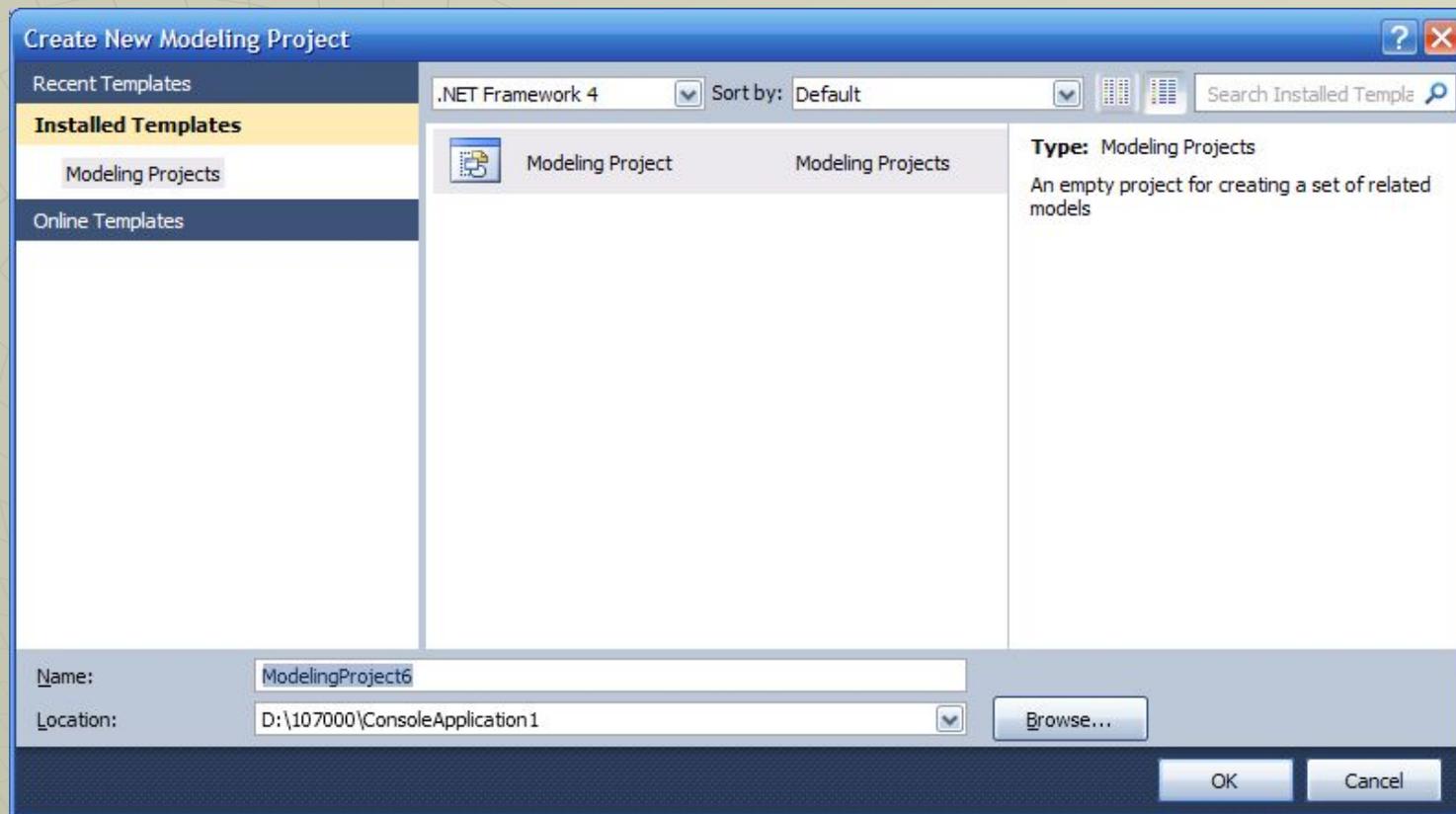
- ◆ Среда MS Visual Studio предоставляет возможность строить UML-схемы.
- ◆ Architecture – New Diagram

Диаграммы классов

- ◆ Выбираем UML Class Diagram

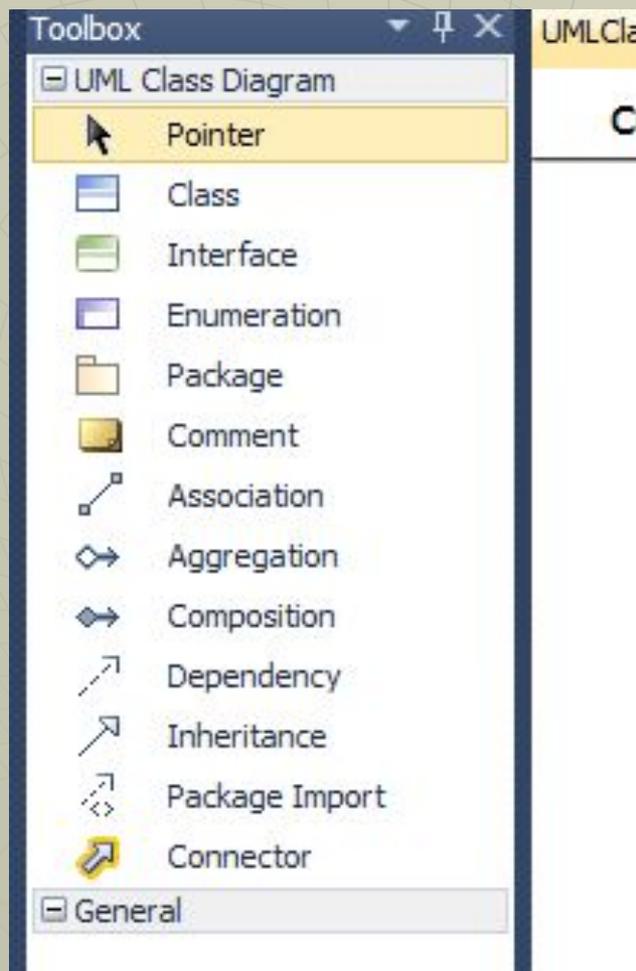


Диаграммы классов



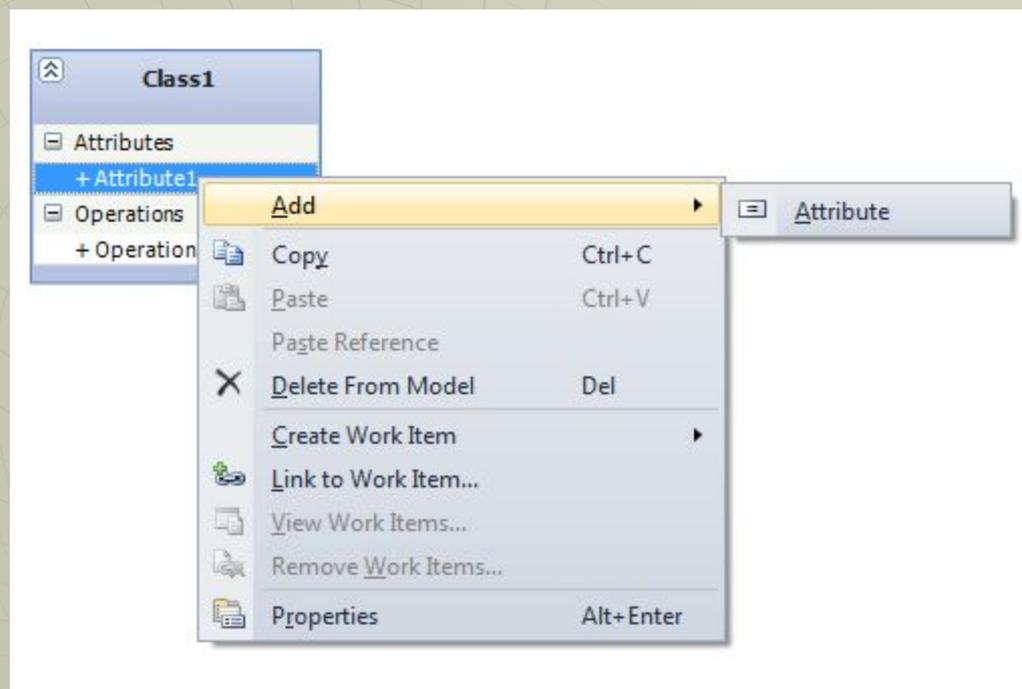
- ◆ Обратите внимание, что эта схема – как отдельный проект в Solution.

Диаграммы классов



- ◆ Слева – панель инструментов
- ◆ Добавляем класс, к классу – атрибуты и операции.

Диаграммы классов



- ◆ На этапе проектирования мы не выделяем – поля это или свойства, обычные методы или перегруженные операции

Диаграммы классов

- ◆ То же самое можно сделать в приложении MS Visio (поставляется отдельно от офиса).