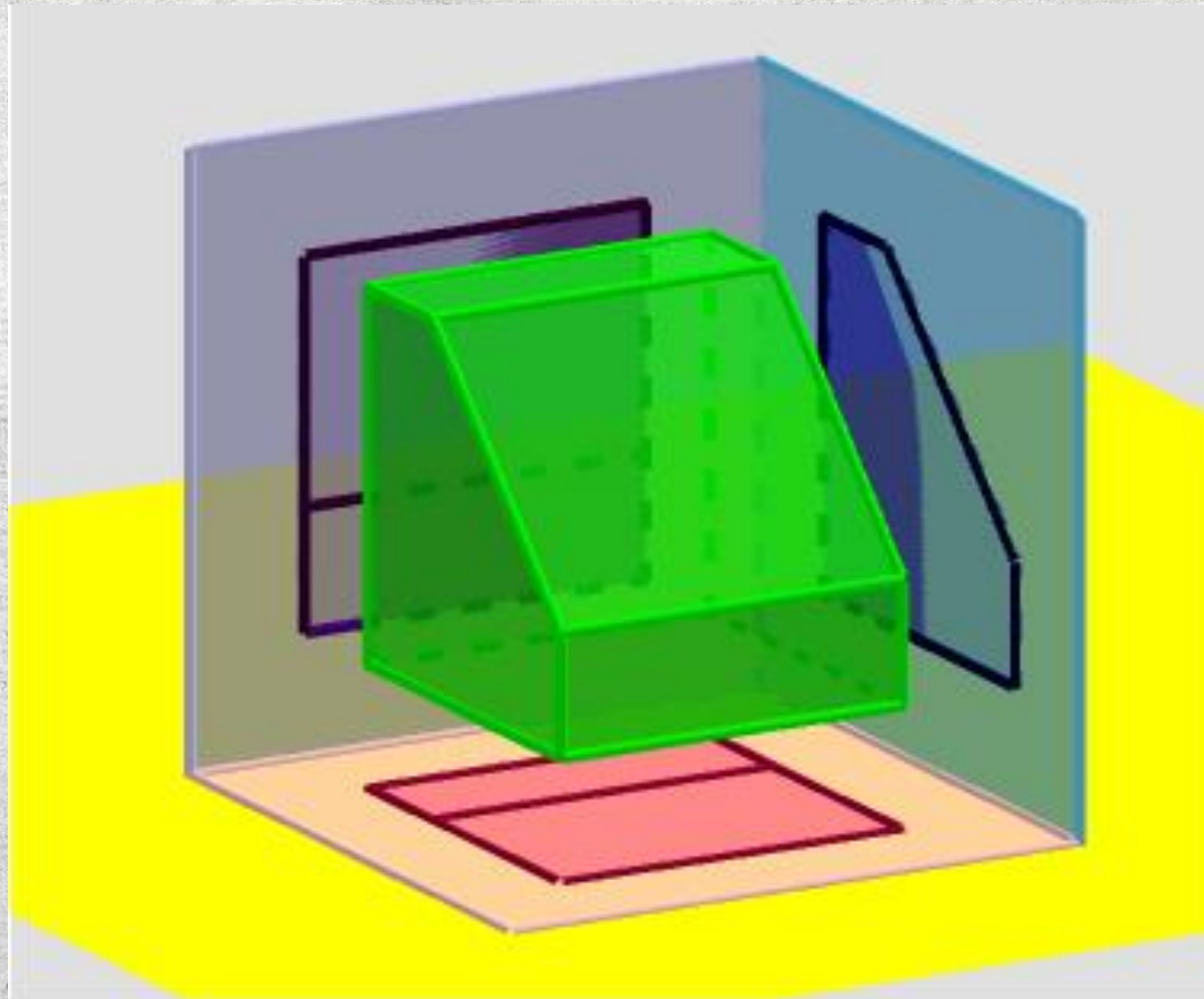


# Unified Modeling Language

Ася Тоскова



# Различни изгледи на софтуера



[http://chertane.valkova.info/9\\_i.php](http://chertane.valkova.info/9_i.php)



# Същност на UML

- Унифициран графичен език за моделиране на софтуер.
- Използва се за **специфициране**, **визуализиране**, **конструирание** и **документиране** на елементите на една софтуерна система.
- Стандартизиран 1997 г. от Object Management Group (OMG) на базата на:
  - OO анализ и дизайн (OOAD) на Grady Booch
  - техники за обектно моделиране (OMT) на James Rumbaugh
  - OO софтуерен инженеринг (OOSE) на Ivar Jacobson.
- Поддържа обектно-ориентирани концепции.
- Настоящата версия е UML 2.5.1



# Концептуален модел на UML

## 1. Градивни блокове

- елементи
- връзки
- диаграми

## 2. Правила за семантична съвместимост на градивните блокове

- име
- обхват
- видимост
- интегритет
- действие

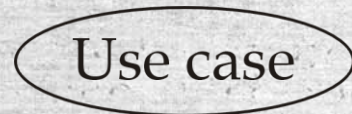
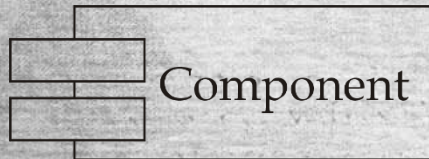
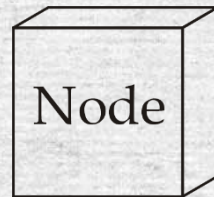
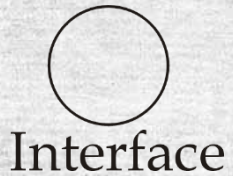
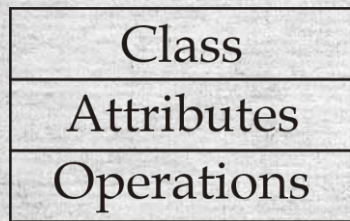
## 3. Специфични механизми



# Елементи

Статични (за структура)

Динамични (за поведение)



Групиращи



Анотиращи





# Връзки

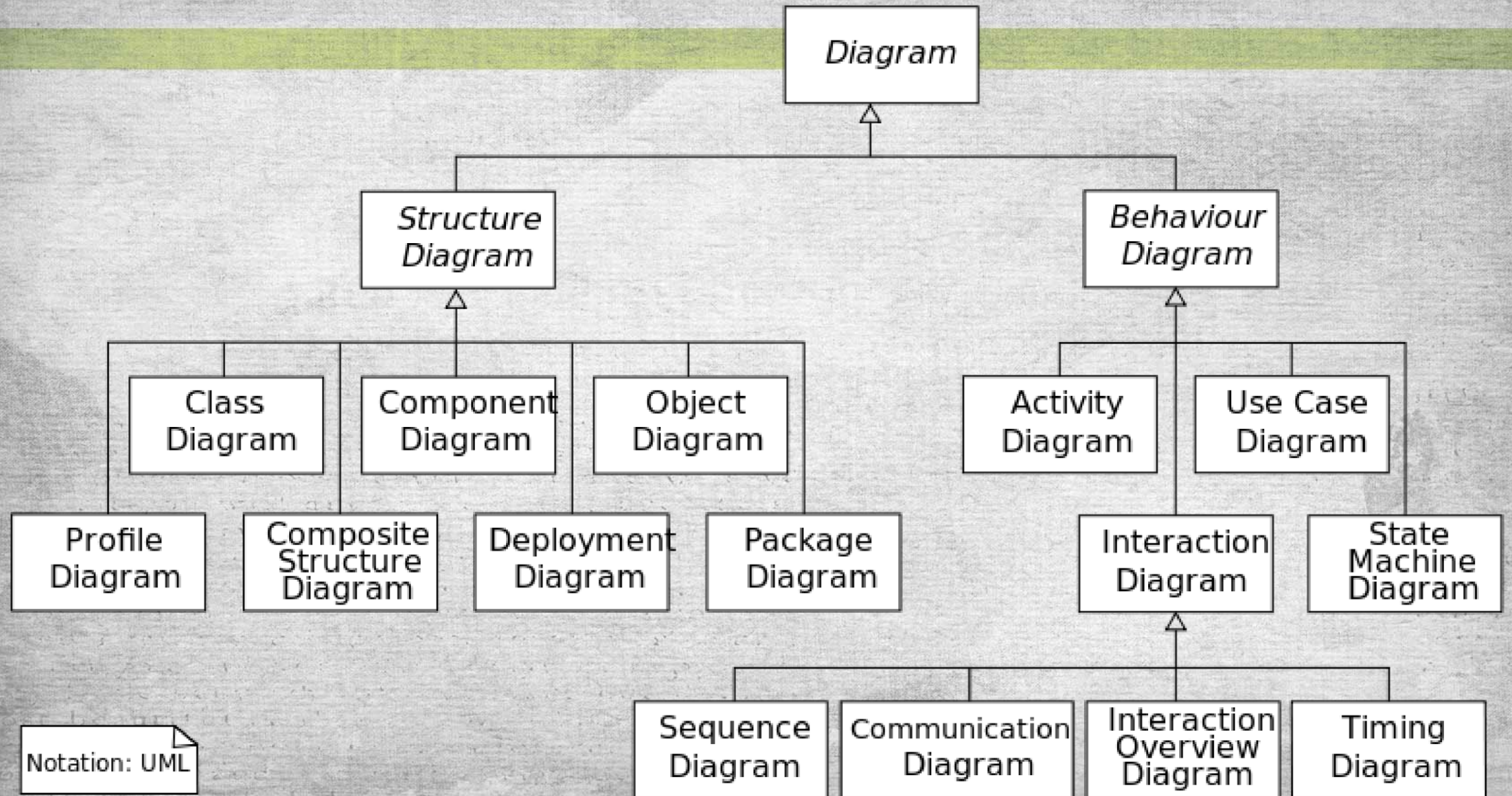
Връзките свързват два елемента или един – сам със себе си (рефлексивна връзка).

## Основни видове връзки

- • Association (асоциация) – обща структура или общо поведение
- > • Dependency (зависимост) – единият елемент (клиент) изисква информация, а другият елемент (доставчик) доставя информация
- ▷ • Generalization (обобщение) – наследяване на структура и поведение
- ▷ • Realization (реализация) - единият елемент определя договор, който другите елементи гарантират, че ще спазват



# Диаграми





# Специфични механизми в UML

1. **Спецификация** – дефиниция на синтаксиса и семантиката на градивните блокове
2. **Декорация** – графични символи за градивните блокове (нотации)
3. **Разделяне на класификатори и инстанции** – напр. клас и обект, интерфейс и реализация, асоциация и линк и др.
4. **Разширяемост** – използване на специфични метаданни, които позволяват разширяване на UML, с цел да се отговори на конкретни изисквания:
  - *Stereotype* (стереотип) – разширява речника на UML – с негова помощ се създават нови градивни блокове от съществуващите
  - *Tagged value* (таг) – разширява свойствата на градивните блокове (напр. име, атрибути, операции, кардиналност, роля и др.)
  - *Constraint* (ограничение) – разширява семантиката на градивните блокове, като задава обхват, условия или правила



# Преобразуване в UML

**Право преобразуване** (forward engineering) - трансформиране на модела в код на избран програмен език. Води до загуба на информация, тъй като моделите, създадени в UML, са семантично по-богати от всеки текущ ОО език за програмиране.

**Обратно преобразуване** (reverse engineering) - трансформиране на код в модел. То е непълно, защото кодът не дава достатъчно детайлна информация за създаването на полезен модел.



# UML инструменти

Инструменти за моделиране на софтуерни приложения на базата на езика UML:

- IBM Rational Rose
- StarUML - <https://staruml.io>
- MagicDraw
- ArgoUML
- Eclipse
- Astah
- ...