

# Class & Object diagrams



{упражнение}

Ася Тоскова



# Съдържание

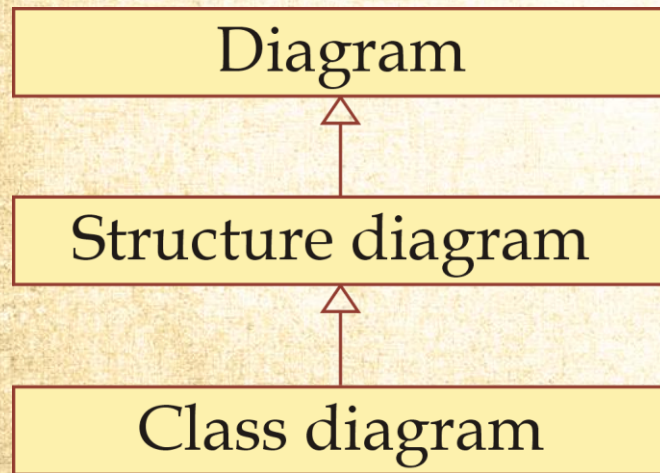
---

- Какво е Class диаграма?
- Елементи и връзки, които участват в Class диаграмата
  - Classes (Класове)
  - Interfaces (Интерфейси)
  - Packages (Пакети)
  - Връзки
- Какво е Object диаграма?



# Какво е Class диаграма?

---

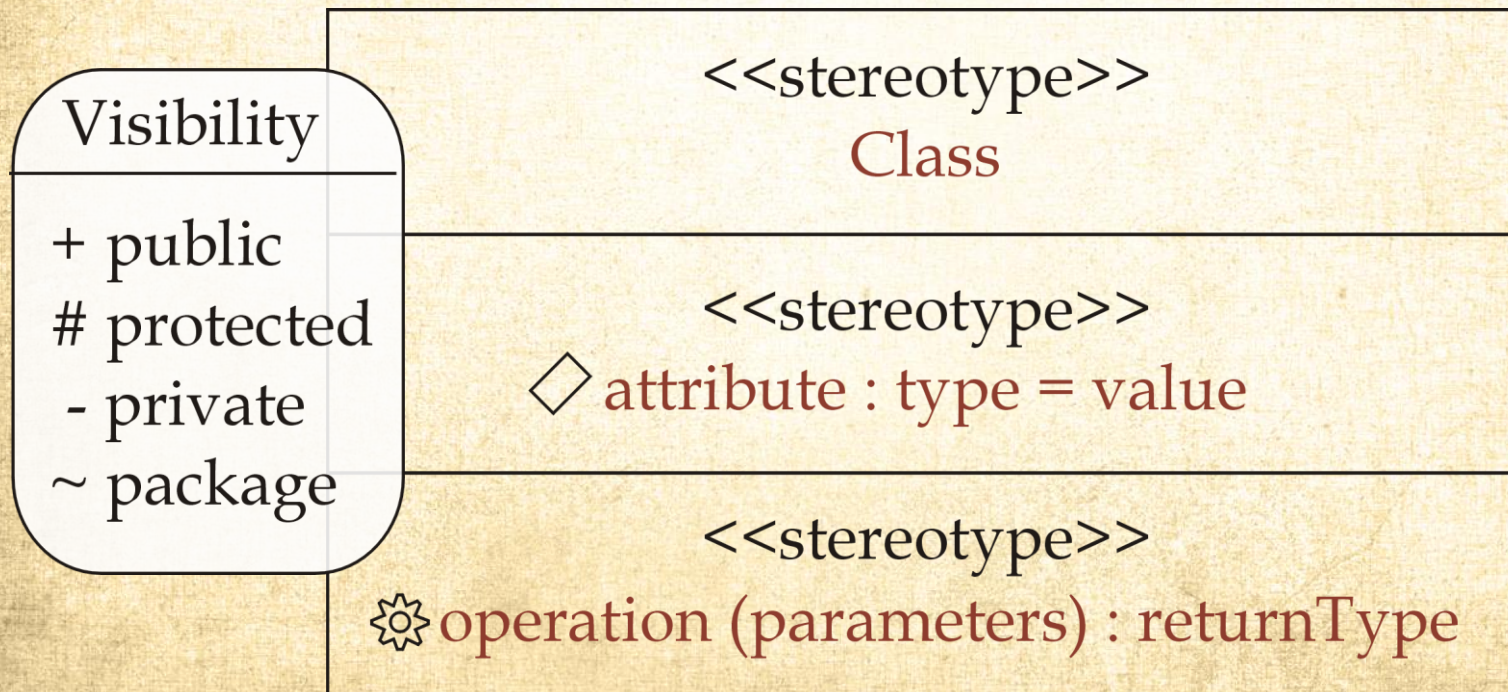


- Показва основната структура на модела.
- Визуализира тази структура с помощта на **classes, interfaces, packages** и връзки между тях.
- Задължително се изготвя при проектиране на приложение.



# Class

**Class** – колекция от атрибути и операции, които описват свойствата и поведението на група идентични обекти.





# Секция - Class

visibility stereotype name

**visibility** – public по подразбиране.

Някои **stereotypes** (категории) за клас:

- **boundary** – клас, който граничи с външната среда.
- **entity** – клас, който съдържа информация, която ще се записва в база данни.
- **control** – клас, който съдържа бизнес-логиката.
- **utility** – клас със статични атрибути и операции.

**name** - име на клас – следва конвенцията Pascal case - едно или няколко съществителни имена, всяко с главна буква, без интервали между тях.



# Секция - Attributes <sup>1/2</sup>

---

visibility stereotype name : type [cardinality] = value {property}

**visibility** – public по подразбиране.

**stereotype** (tagged values) - категория за атрибут.

**name** – следва конвенцията Camel case -  
едно или няколко съществителни имена, първото – с  
малка буква, всяко следващо - с главна буква, без  
интервали между тях.

**type** - примитивен или референтен тип.



# Секция - Attributes <sup>2/2</sup>

---

**cardinality** (multiplicity) – ограничение на броя на заеманите различни стойности. По подразбране  $\text{cardinality} = 1$ .

Някои **properties** на атрибут:

- По подразбиране - стойността може да се променя
- **static** – статичен атрибут, общ за всички инстанции
- **derived** – стойностите му се изчисляват на базата на други атрибути
- **leaf** – не може да се предефинира
- **unique** - стойностите му не може да се повтарят



# Секция - Operation <sup>1/2</sup>

visibility stereotype name (parameter\_list) : returnType {property}

**visibility** – public по подразбиране.

**stereotype**:

- **implementer** – изпълнява бизнес-логика
- **manager** – управлява създаване/унищожаване на обект
- **access** – позволява на други класове да променят атрибути
- **helper** – private/protected помощни операции

**name** – Camel case - комбинация от глагол (с малка буква) и съществително име (с главна буква), без интервал между тях.



# Секция - Operation <sup>2/2</sup>

---

**parameter\_list** - може да има 0 или повече входящи параметри.

(parameter1 : type, parameter2 : type, ...)

**returnType**: тип на върнатия от операцията резултат.

Някои **property** на операция:

- **abstract** – няма тяло, трябва да се предефинира (override)
- **unique** – не може да има повтарящи се стойности
- **leaf** – не може да се предефинира.
- **concurrency** – описва паралелен процес
- **static** – статична операция, обща за всички инстанции



# Interfaces

---

В Class-диаграмата може да има интерфейси.



Interface

**Interface** – колекция от атрибути и операции, която определя абстрактно единен набор от поведение.

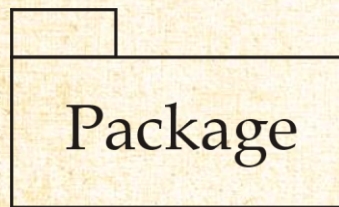
Интерфейсите се **реализират** от класове или компоненти, които прилагат в конкретна ситуация атрибутите и операциите на интерфейса. Един клас/компонент може да реализира множество интерфейси.



# Packages

---

В Class-диаграмата може да има пакети.



**Package** – колекция от елементи, които са семантично свързани.

В пакета може да се постави всеки един вид елемент. Може да се използва expand/collapse контрол, за да се показва или скрива съдържанието на пакета.



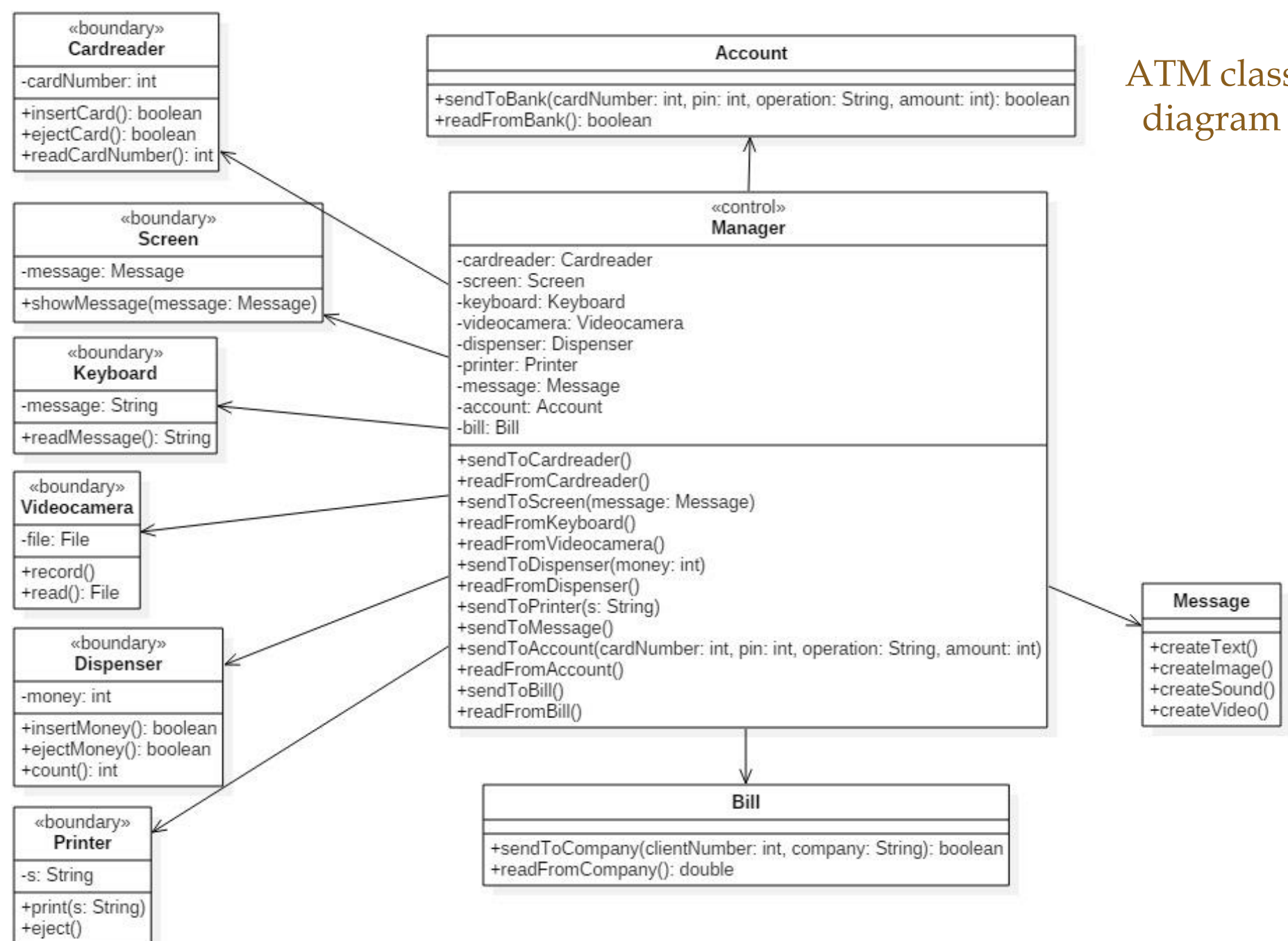
# Връзки в Class diagram

---

- Association – директна, двупосочна, агрегация, композиция
- Dependency
- Generalization
- Realization



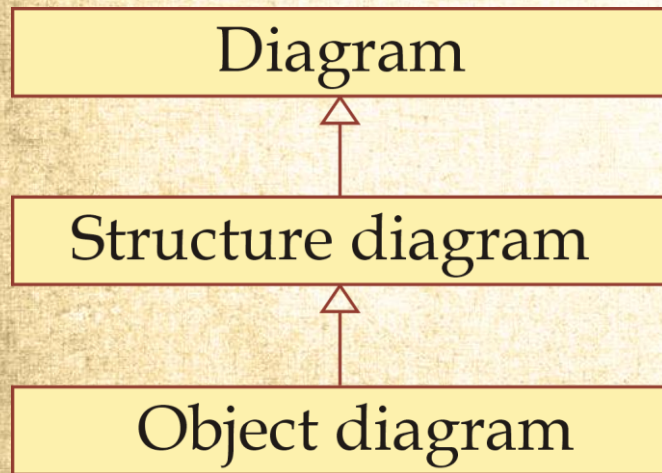
# ATM class diagram





# Object diagram

---



- Структурна диаграма.
- Показва множество обекти, тяхното състояние и отношенията между тях в един определен момент от време.
- Моментна „снимка“ на клас-диаграмата.
- В едно състояние, атрибутите на обект заемат точно определени стойности.



# Елементи на Object diagram

---

Object диаграмата е изградена от:

- Обекти в определено състояние
- Връзки между тях

Може да се каже, че Object диаграмата е „инстанция“ на Class диаграма.

Връзките между обектите в Object диаграмата са еднакви с връзките между класовете в Class диаграмата.