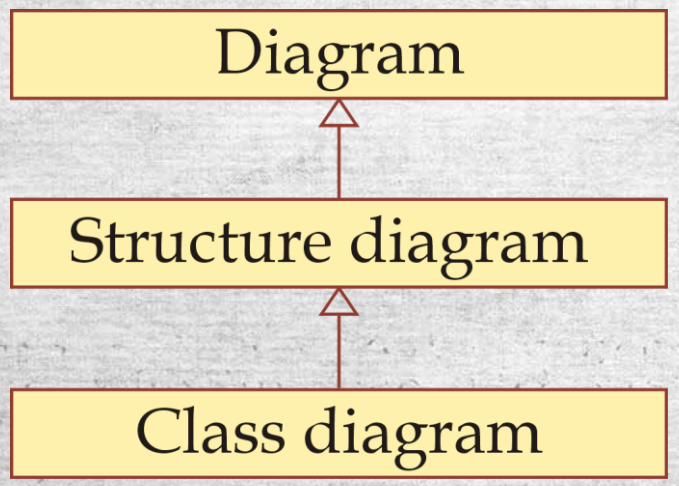
Class & Object diagrams

2020

Ася Тоскова

Class diagram



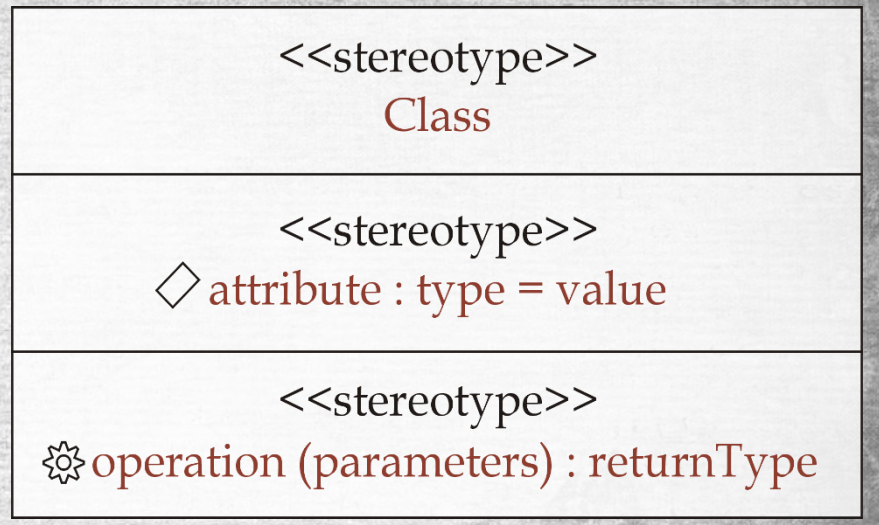
- Структурна (статична) диаграма - показва структурата на софтуера.
- Съдържа класове, интерфейси, пакети и връзки между тях.
- Задължително се изготвя при моделиране на софтуер.

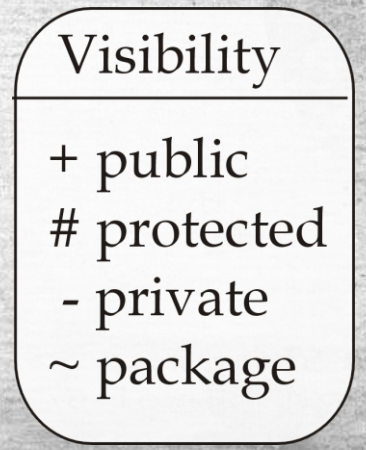
Елементи в class diagram

Клас - колекция от атрибути и операции, която описва абстрактно свойствата и поведението на предмет от реалния свят.

Интерфейс - колекция от атрибути (константи) и операции (сигнатури), която определя единен набор от поведение за реализиращите го класове.

Пакет – колекция от класове, интерфейси и пакети, 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.





Секция “Class”

visibility <<stereotype>> name

visibility – видимост на класа, по подразбиране visibility = public

stereotype – категория за клас

Някои възможни stereotypes:

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

name – име на клас, следва конвенцията Pascal case

Секция “attribute”

visibility <<stereotype>> name: type [multiplicity] = value {property}

visibility – видимост, по подразбиране visibility = public
stereotype – категория за атрибут
name – име, следва конвенцията Camel case
type - тип на атрибута, примитивен или референтен
multiplicity – кардиналност, по подразбране multiplicity = 1
property – свойство на атрибут, по подразбиране = стойността може да се променя
Някои възможни properties:

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

Секция “operation”

visibility <<stereotype>> name (parameter_list): returnType {property}

visibility – видимост, по подразбиране visibility = public

stereotype – категория за операция

Някои възможни stereotypes:

- implementer – изпълнява бизнес-логика
- manager – създава/унищожава обект
- access – позволява на други класове да променят атрибути
- helper – private/protected помощни операции

name – име, следва конвенцията Camel case

Секция “operation”

parameter_list – списък с формални параметри, по подразбиране - няма параметри

(parameter1: type [multiplicity], parameter2: type [multiplicity], …)

parameter – параметър, type – тип на параметъра, multiplicity – кардиналност

returnType – тип на операцията, по подразбране returnType = void

property – свойство на операция, по подразбиране = стойността може да се променя

Някои възможни properties:

- abstract – абстрактна, трябва да се предефинира (override)
- leaf – не може да се предефинира
- static – статична операция, обща за всички инстанции

Връзки в Class diagram

Връзки между класове

- Асоциация
  - Двупосочна асоциация (или само асоциация)
  - Директна асоциация
  - Агрегация
  - Композиция
- Наследяване

Връзка между клас и интерфейс

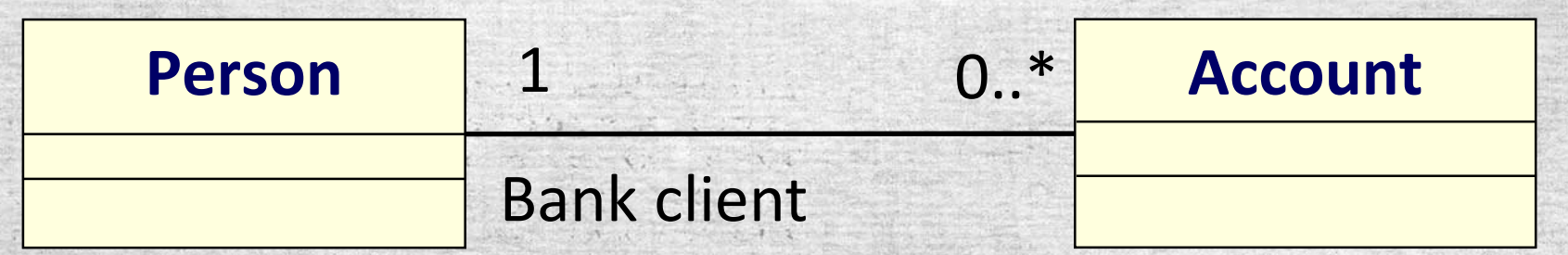
- Реализация

Връзка между пакети

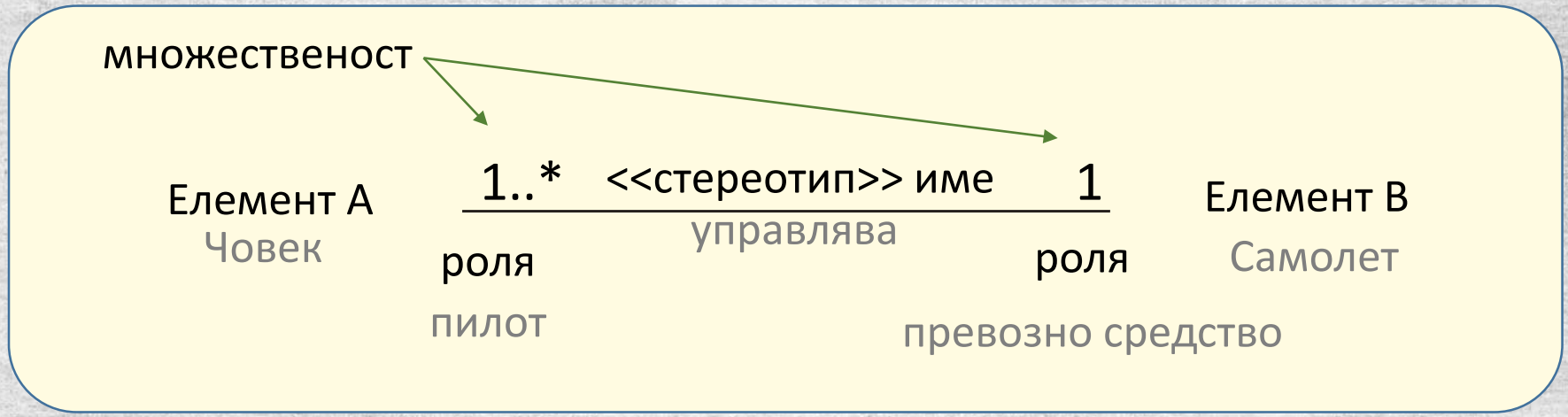
- Зависимост

Асоциация (двупосочна)

Асоциация – връзка между класове, които имат обща структура или поведение. Асоциацията (двупосочната асоциация) обикновено се използва, за да покаже множественост и роли.



Множественост



Множественост – допустим минимален и максимален брой инстанции на елемента за тази връзка

Стойности: 0, цяло положително число или неопределен брой = много (*)

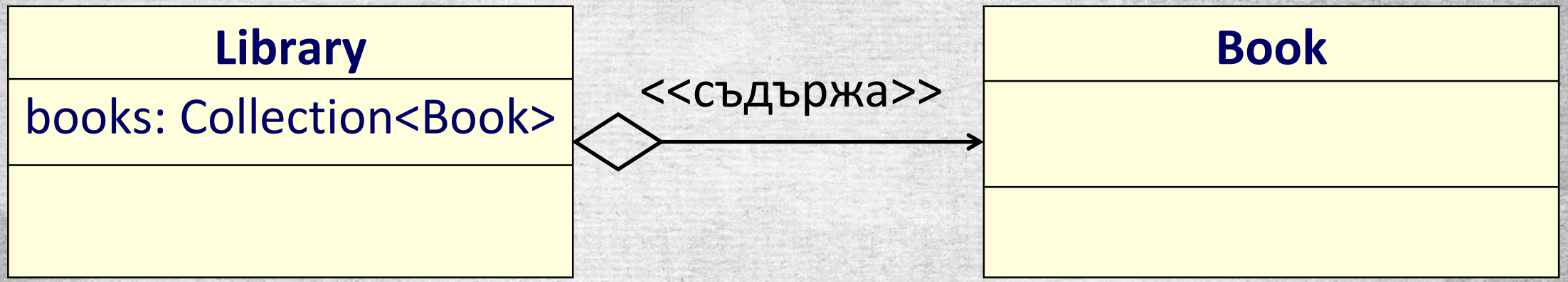
Задава се като интервал (min..max) или ако стойностите съвпадат – като едно число

Отношения: 1:1, 1:M, M:M

Директна асоциация (Directed association)

Директната асоциация е вид асоциация.
Показва, че единият клас е контейнер за другия клас, като съдържа единичен обект от него.
Клас Person съдържа обект от клас Account.

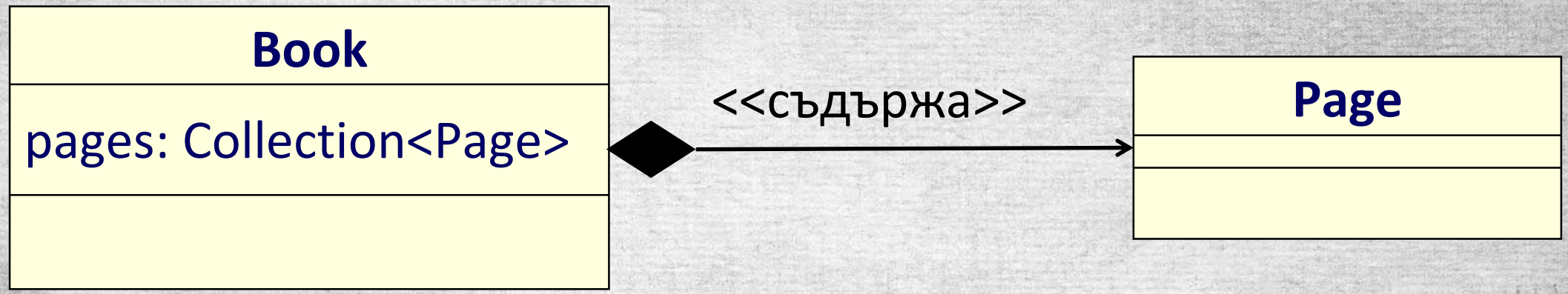
Агрегация (Aggregation)



Агрегацията е вид асоциация.
Показва, че единият клас е контейнер за другия клас, като съдържа група обекти от него. Тези обекти могат да съществуват и самостоятелно.

Клас Library съдържа колекция от обекти на клас Book.

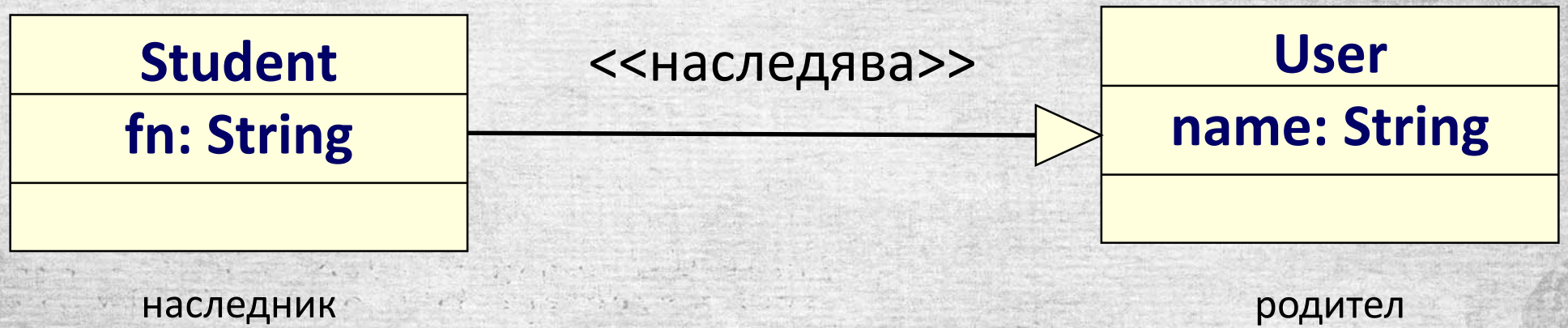
Композиция (Composition)



Композицията е вид асоциация.
Показва, че единият клас е контейнер за другия клас, като съдържа група обекти от него. Тези обекти не могат да съществуват самостоятелно.

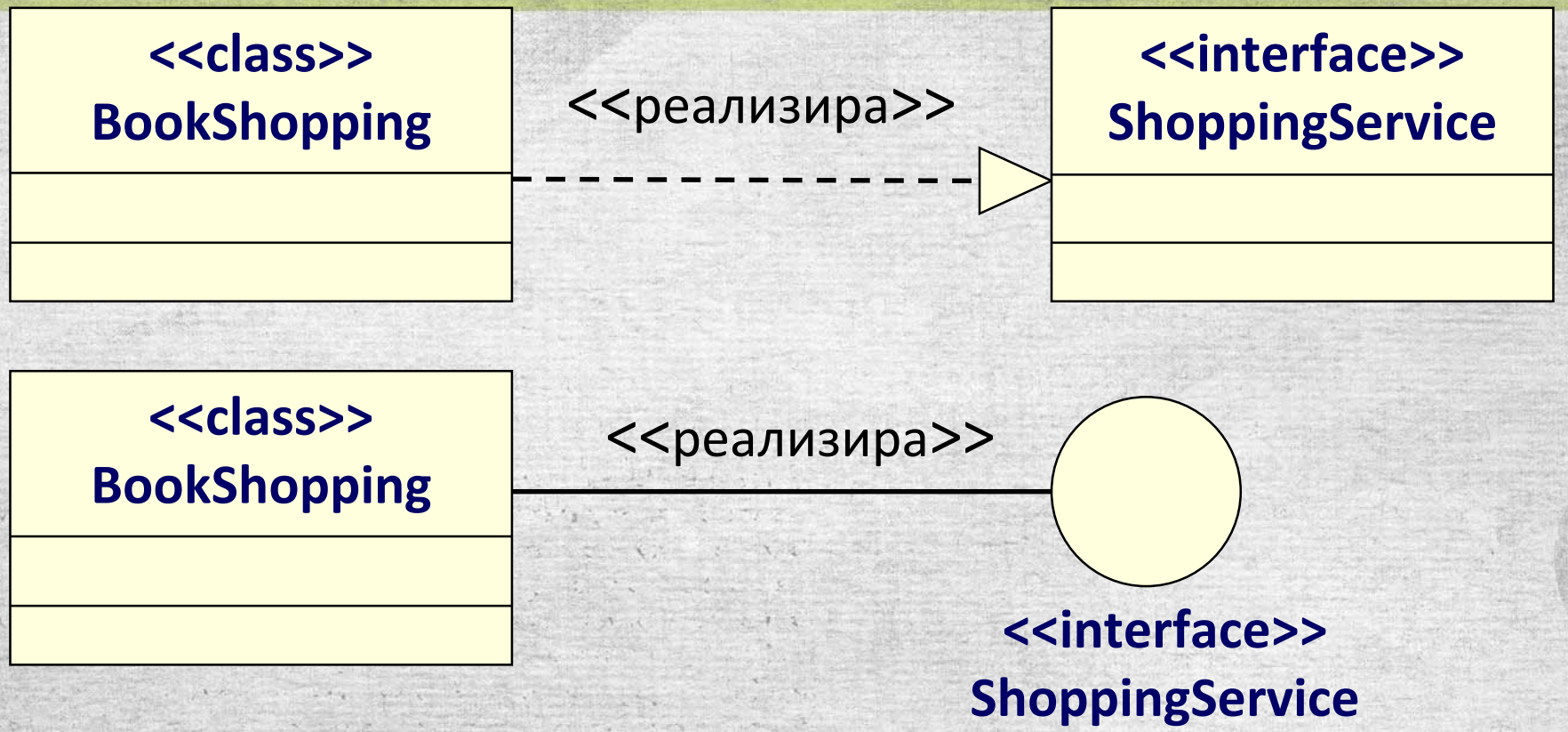
Клас Book съдържа колекция от обекти на клас Page.

Наследяване (Generalization)



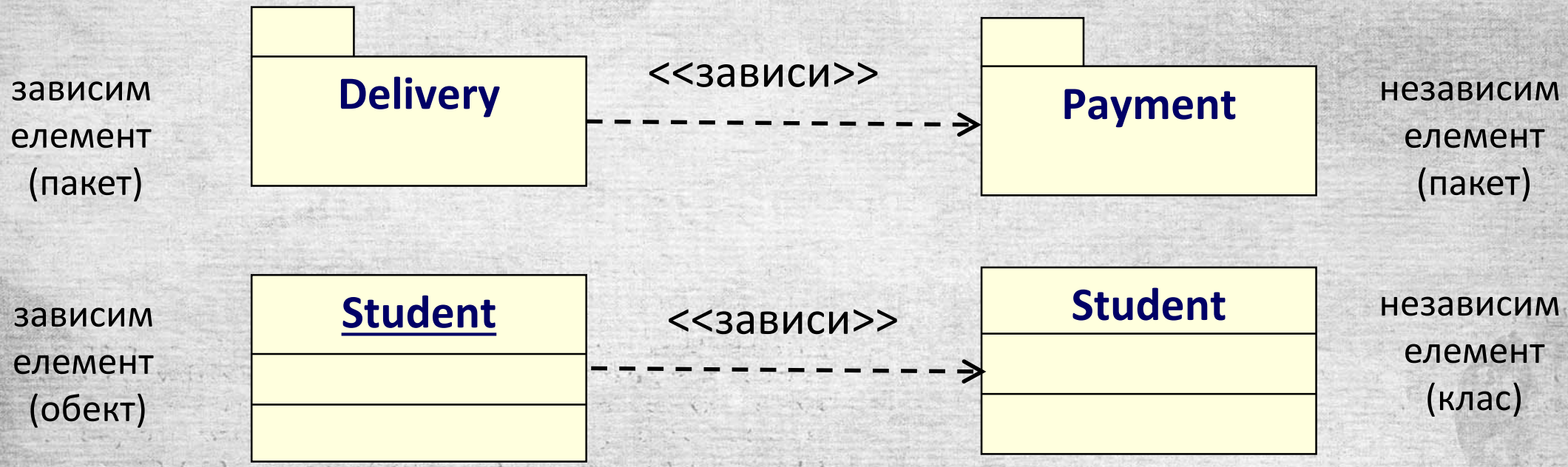
Връзка между два класа (superclass и subclass), която показва наследяване.
Клас Student (subclass) наследява клас User (superclass).

Реализация (Realization)

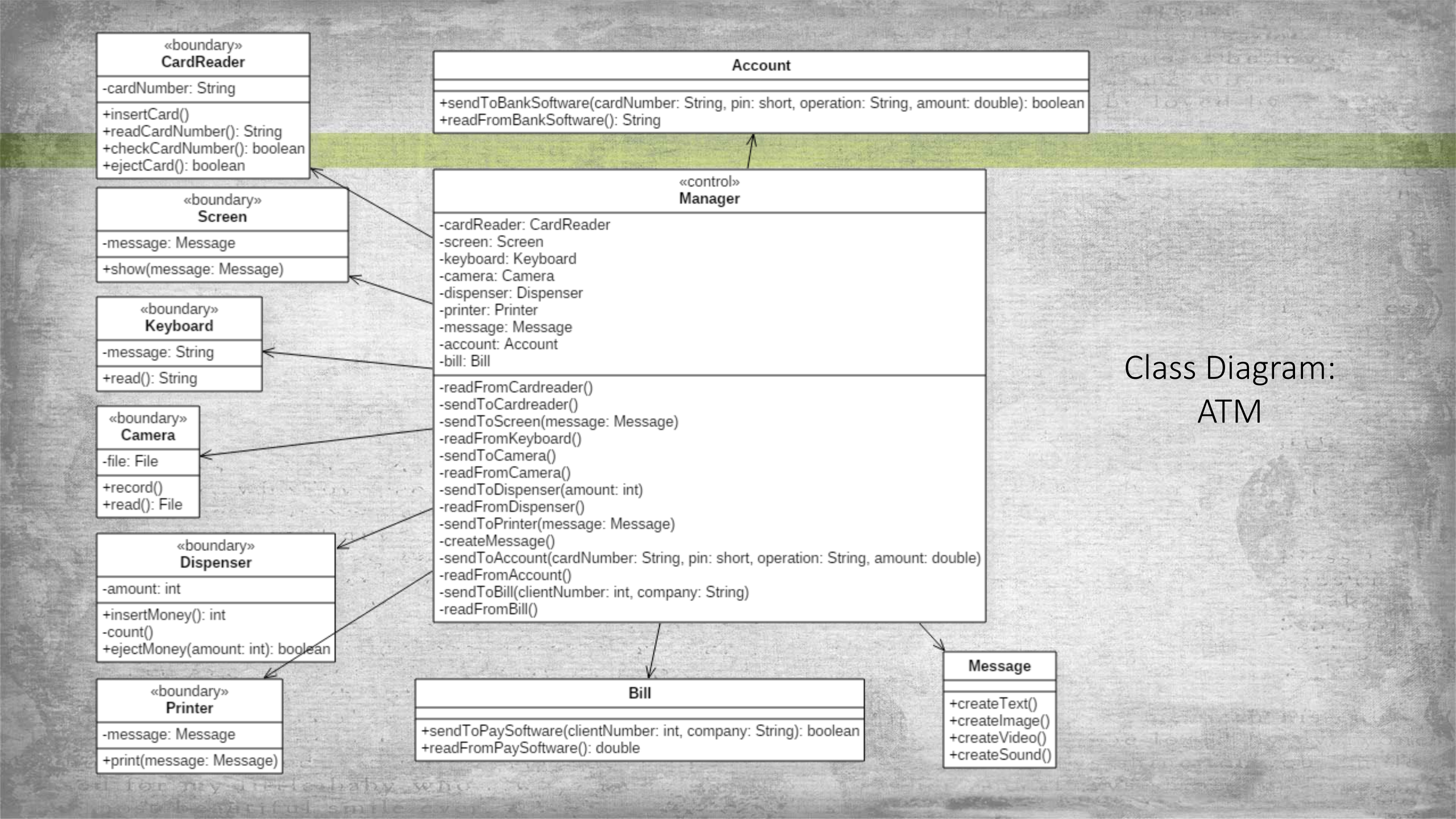


Клас BookShopping реализира зададения шаблон от интерфейса ShoppingService.

Зависимост (Dependency)

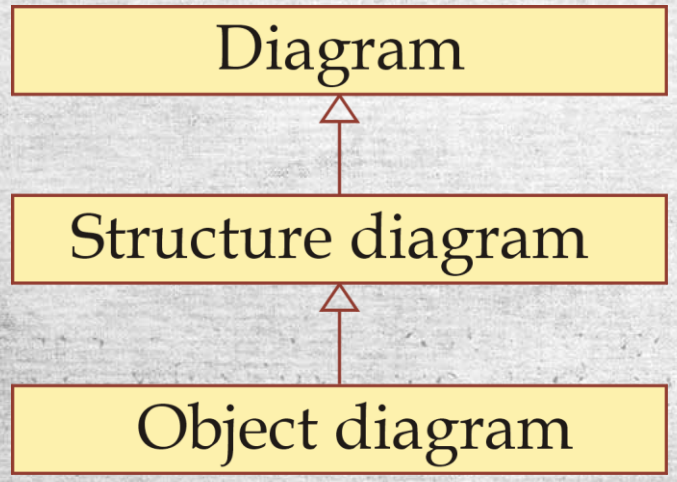


Връзка между два елемента, при която промяната на независимия елемент води до промяна на зависимия елемент.



Class Diagram: ATM

Object diagram



- Структурна диаграма.
- Моментна „снимка“ или „инстанция“ на клас-диаграмата.
- Показва обектите, тяхното състояние и отношенията между тях в един определен момент от време.
- В едно състояние, атрибутите на обект заемат точно определени стойности.
- Връзките между обектите в Object диаграмата са еднакви с връзките между класовете в Class диаграмата.