Лекция 4
Основни концепции за функционалния изглед на системата

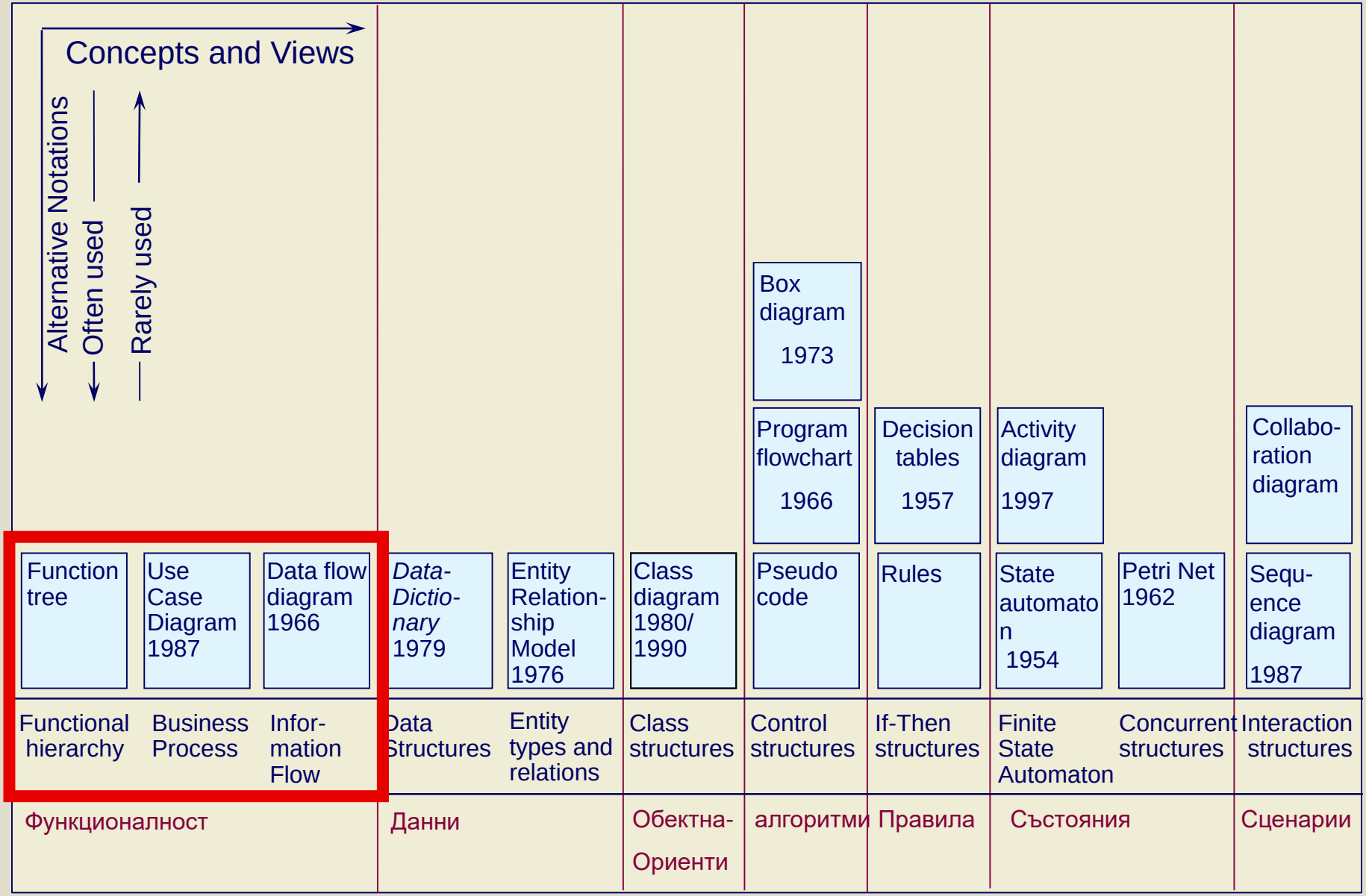
DAAD Project
“Joint Course on Software Engineering”

**Humboldt University Berlin, University of Novi Sad, University of Plovdiv,
University of Skopje, University of Belgrade, University of Niš, University of Kragujevac**

**Parts of this topic use material from the textbook
H. Balzert, “Software-Technik”, Vol. 1, 2nd ed., Spektrum Akademischer Verlag, 2010**

Основни концепции при разработката на софтуер

Balzert vol. 1, 2nd edition 2001



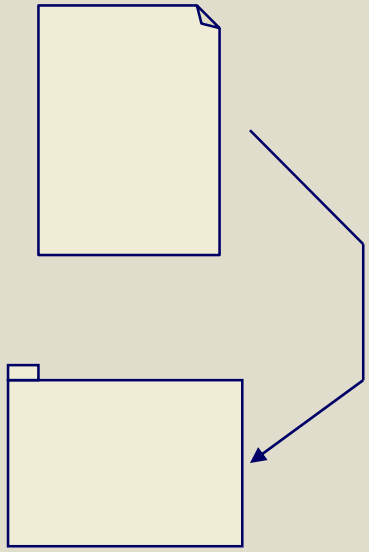
Задачи на основните концепции

Спецификация на изискванията:

езикова,
неформална …

Модел на продукта:

формализиран,
формален …



Основни концепции:

Да направи изискванията за SW продукт по-прецизни.

Пример:

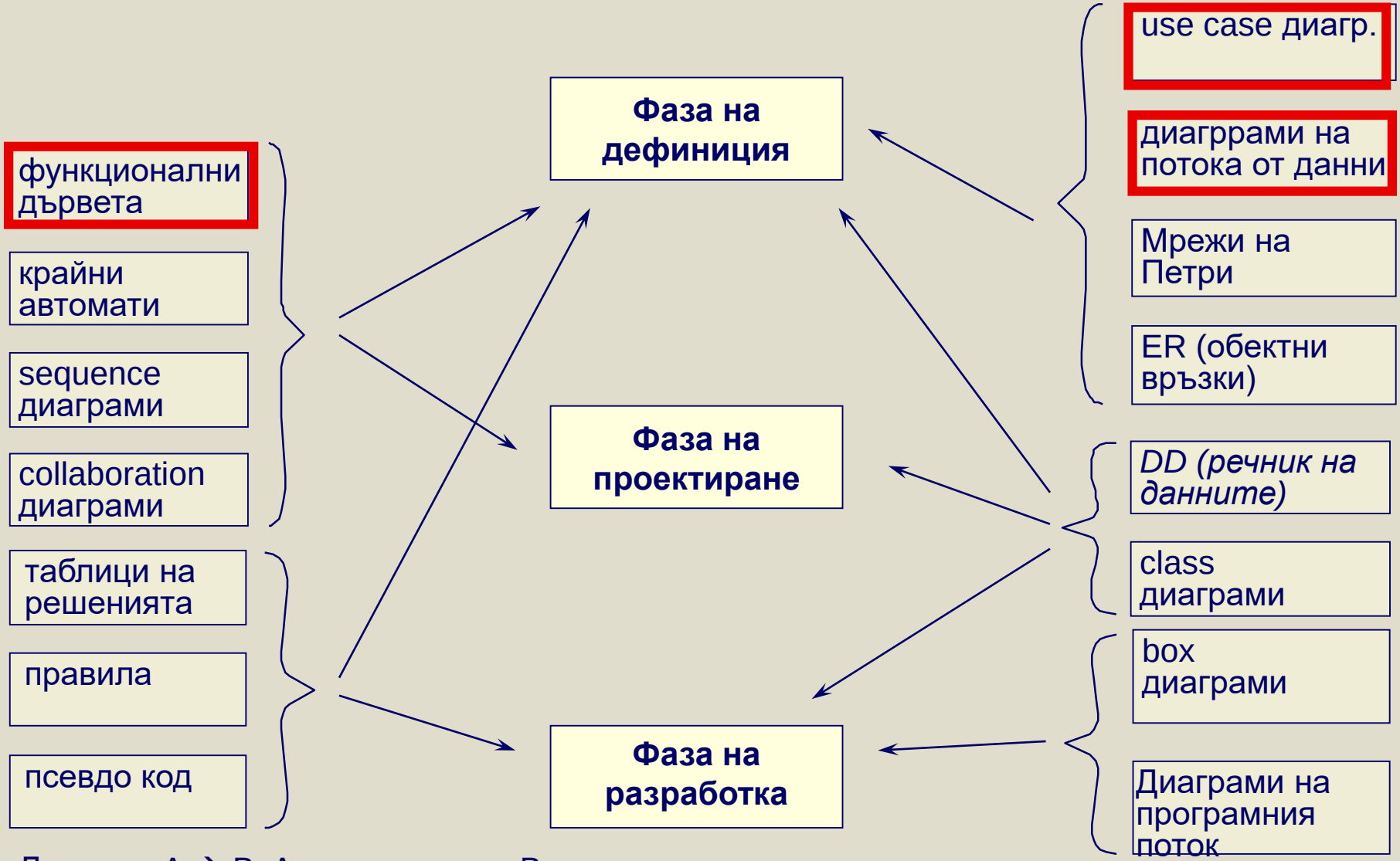
Спецификация на изискванията

- линеен списък от функции на продукта

Функционално дърво

- Функциите на продукта са подредени в йерархия

Основните концепции и фазите за разработка на софтуер



Легенда: A → B: A се използва в B

Класификация на основните концепции според техните нотации

7. Основни концепции за функционалния изглед на системата

a) Функционални дървета

b) Диаграми на потока от данни

c) Use case диаграми (бизнес процес)

Функции, Функционални дървета

- *Функция =*
  - Задача на софтуера, функция на продукта (функционална спецификация)

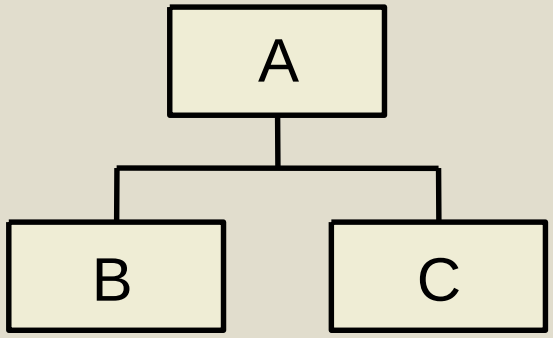
    → Фаза: Анализ и дефиниция
  - Функция, Процедура, Метод

    →Фази: Проектиране и Разработка

- *Функционално дърво =* Йерархия на функциите
  - Функции → Подфункции → …

Графично представяне

Йерархия



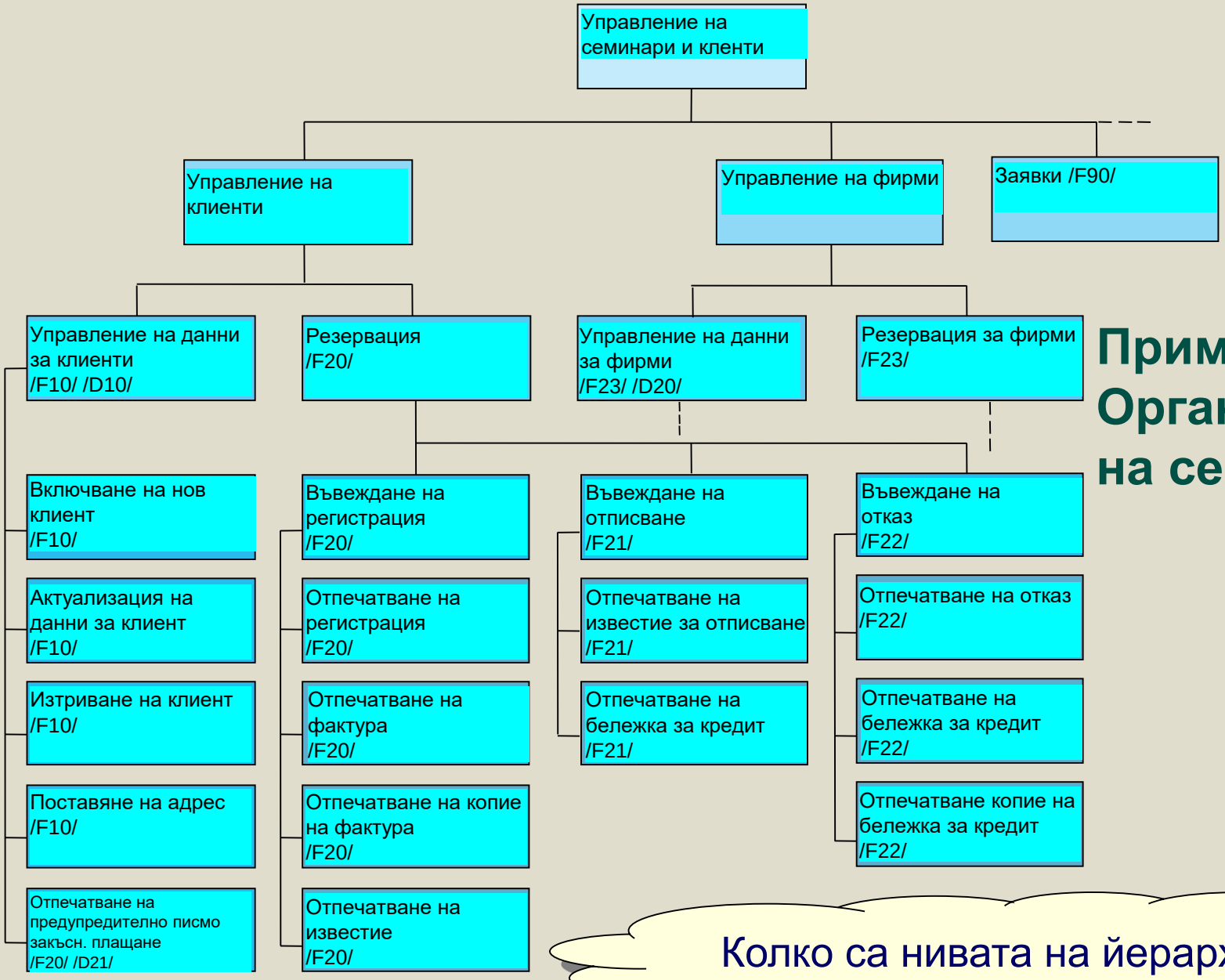
A, B, C: имена на функциите

Основа на йерархия:

Фаза на дефиниция:
A се състои от B и C
→ FT = Зададена йерархия

Фаза на проектиране:
A извиква B и C
→ FT = йерархия на извикване

предупреждение:
Зададената йерархия не е необходимо да съдържа йерархията на извикване
(например:
задача → изпълнима функция)



Пример: Организация на семинар

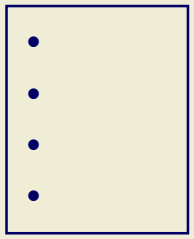
Колко са нивата на йерархия?

Функционални дървета: оценка

+ Изпитана концепция за систематичен развой и декомпозиция на функциите

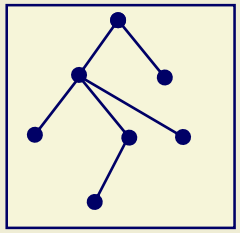
Спецификация на изискванията:

линеен списък от функции



Функционално дърво:

йерархична подредба на функции (по-прецизна)



7. Основни концепции за функционалния изглед на системата

a) Функционални дървета

b) Диаграми на потока от данни

c) Use case диаграми (бизнес процес)

Поток данни

Функция

хранилище

Външни обекти

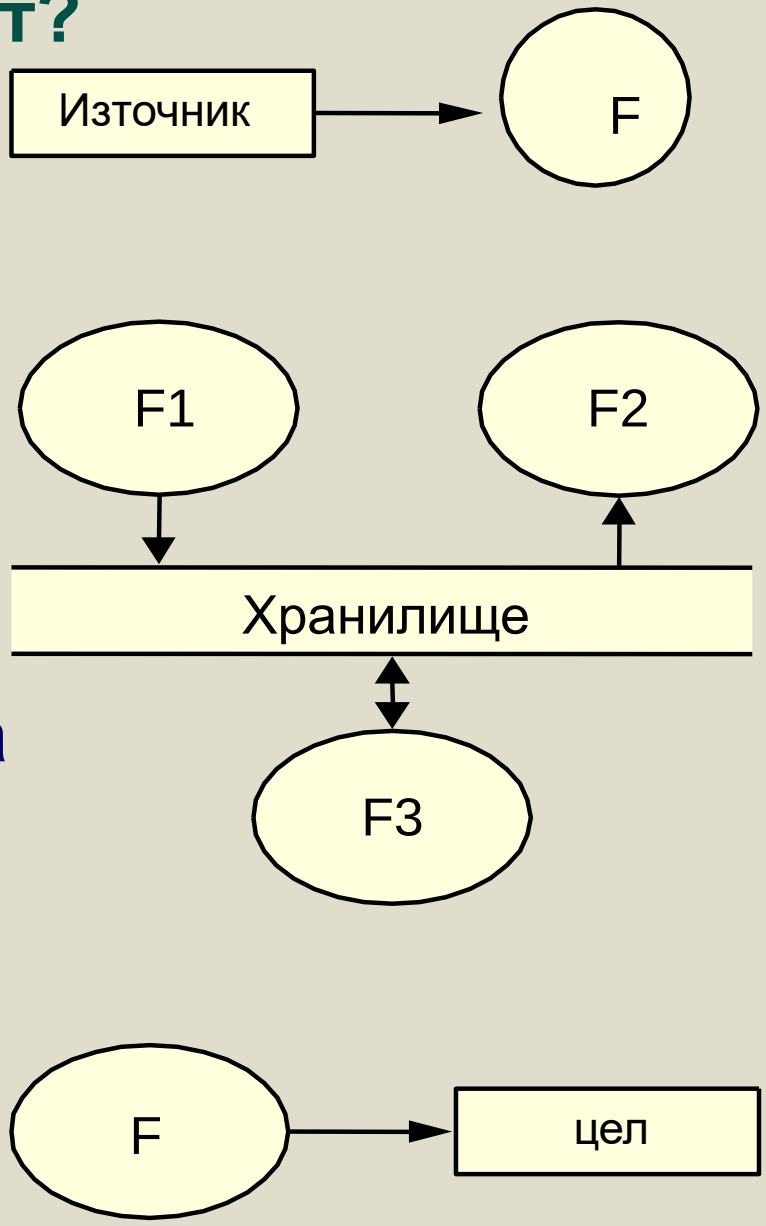
Част(обект) от средата

Информационен източник

Нотация на потока от данни, DeMarco (1979)

Диаграми на потока от данни: Какво представят?

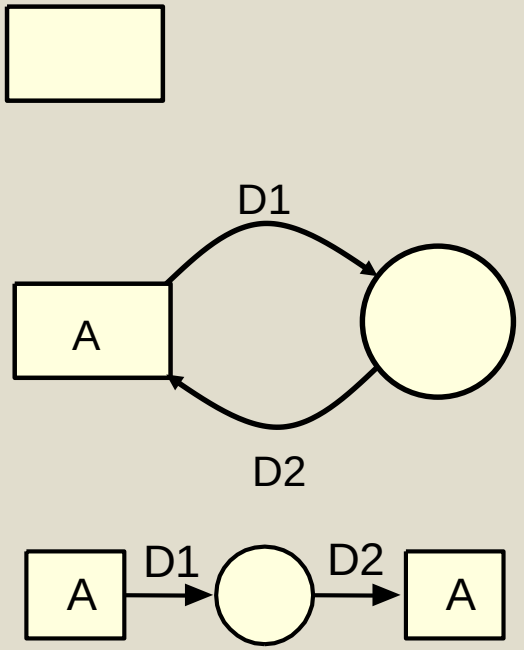
- Основна идея: системата, която се разработва е работеща
- → Информационния поток се представя в работеща система
- → Информационните пътища между
  - функции
  - хранилища
  - Външни обекти: източници, цели



Пример за диаграма на потока от данни:организация на семинар

Синтактични правила за диаграмите на потока от данни (DFD) (1)

- Всяка DFD съдържа *най-малко един външен обект*
- Всеки външен обект по принцип се *съдържа само веднъж*
  - Изключение: Ако DFD не е достатъчно ясна, тогава външния обект може да бъде повторен
- Всеки поток от данни има име
  - Изключение: поток от данни, който води/идва към/от хранилището.Приема се че те съдържат всички данни от хранилището.



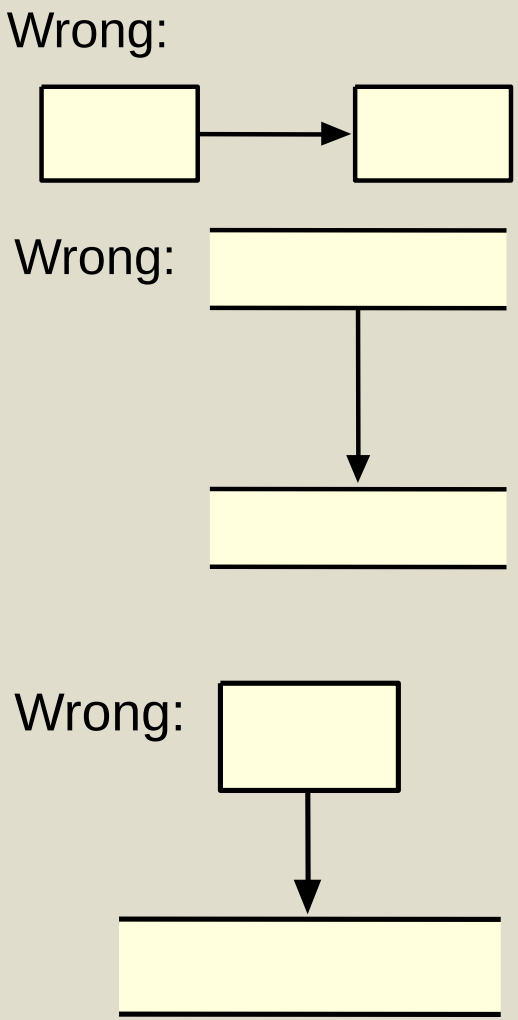
Синтактични правила за DFDs (2)

- Между външните елементи няма потоци от данни
- Между хранилищата няма директни потоци от данни
- Между външните обекти и хранилищата няма директни потоци от данни

→ Функциите са винаги по между им

→ Основната задача на DFD:

описание на функциите на продукта



Семантични правила за DFDs (1)

Интерфейс

1. Външен обект се избира, така че той да представя оригинален източник или задача на информацията

   Пример: информацията за клиент произлиза от клиентите, не от менажерите на клиенти

2. Избор на външен обект означава абстракция от конкретно въвеждане на информация (от клавиатурата) или извеждане (отпечатване).

Семантични правила за DFDs (2)

Име на потока от данни

3. Името на потока от данни се състои от съществително или от прилагателно и съществително

   Пример:
   ‚Сметка-N‘, ‚валиден N-Сметка‘

4. Отбягвайте имена на потоци от данни с общо значение: ‚Данни‘, ‚Информация‘, ...

Семантични правила за DFDs (3)

Имена на функции

- Името на функцията съдържа единичен *повелителен глагол* следван от име на единичен *конкретен обект* или единично *съществително* следвано от *повелителен глагол*.

7. Общи не съдържателни имена на функции трябва да се избягват (‚процес‘, ‚управление‘, ...)

Диаграми на потока от данни: оценка

+ Лесни по устройство и четаемост
+ Добра нотация за разговор между партньорите (клиент и доставчик)
+ Съдържат повече информация от функционалните дървета
- DFD ще станат много големи и нечетаеми за цялата система;

  решение: нива на йерархия (метод на структурния анализ)

Общ изглед: функционални дървета и диаграми на потока от данни

- Обща цел:
  - Да опишат функциите на продукта
  - → но в различни аспекти

Функционални дървета:	Диаграми на потока от данни:
▸ Разделят приложението на по-прости функционални идентичности (функции) →Приложението се представя като йерархия от функции; →Това представяне дава възможност на разработчиците да следват един top-down подход; →Функциите като цяло представят приложната област.	▸ Ориентирани към друга цел: • функциите се описват с техните влияния: - преобразуване поведението на функциите - как въвеждането (извеждането) на информация в/от функция влияе върху въвеждането(извеждането) на други функции

7. Основни концепции за функционалния изглед на системата

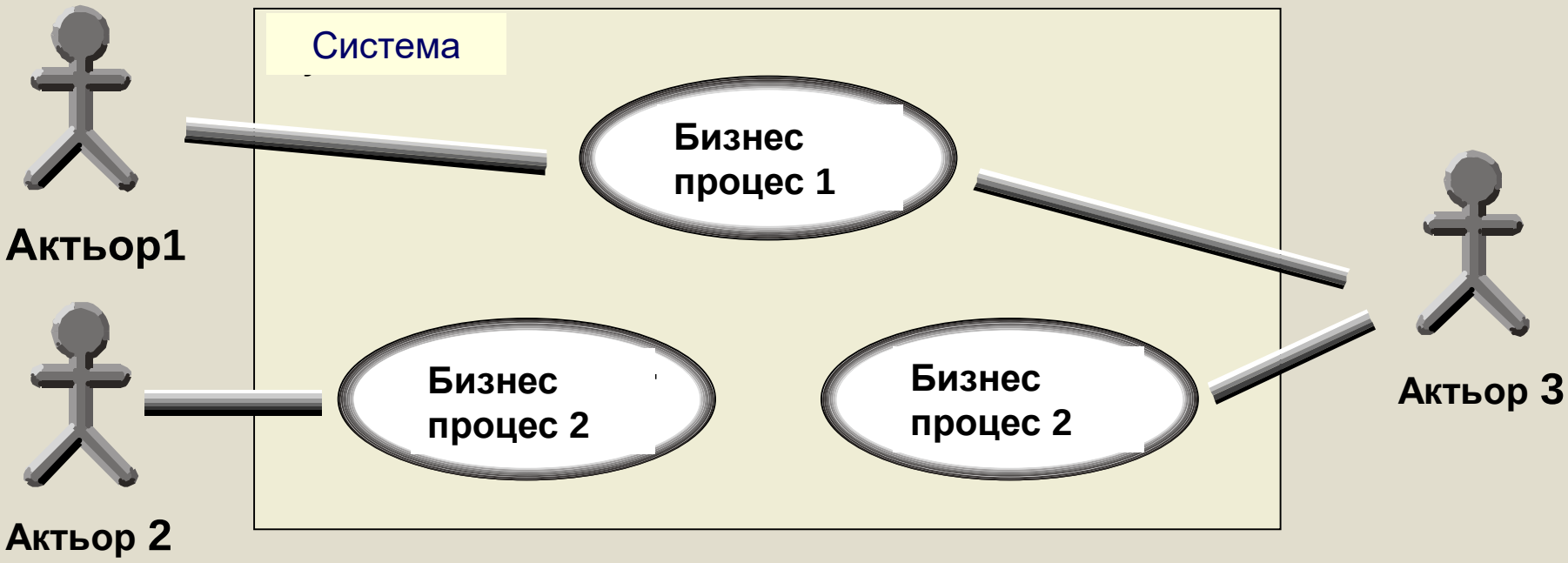
a) Функционални дървета

b) Диаграми на потока от данни

c) Use case диаграми (бизнес процес)

Use Case Диаграми
(диаграми на бизнес процеса)

Описва всички бизнес процеси, връзките им един с друг и с актьорите.



Бизнес процес = Use Case

- Основен бизнес процес: работен поток, който ще бъде изпълняван от софтуера и не само от него.
- Use Case
  - Част от основния бизнес процес, който описва комуникацията на клиента със софтуерната система

Use case в UML

(според Booch, Rumbough, Jacobson)

Use case =

- Фундаментална подфункция от системата (услуга на системата)
- със стойност за потребителя (видим резултат)
- реализирана чрез последователност от взаимодействия между потребителя и системата

→ Use cases описват функционални изисквания

→ Функционално описание на системата

= множеството от всички use cases

Необходимо е допълнително описание

Описание на use cases

Описателни техники:

- текстово (текстова схема)
- collaboration диаграми
- sequence диаграми
- activity диаграми
- крайни автомати
- мрежи на Петри

Описание на use cases: пример „Информиране“
(от: спецификация на изискванията „организация на семинар“)

Схема за текстово описание на use cases

F10 (PF10)
Use case: информиране: от въпрос до информация
Цел: клиента получава исканата информация или информацията му се изпраща
Категория: основна
Предусловия:
Успешни постусловия: клиента получава исканата информация
Неуспешни постусловия: исканата информация не може да бъде извлечена
Актьори: менажер на клиенти, клиент, фирма
Първоначални събития: клиента пише (писмо, fax, e-mail) или се обажда
Описание:
1. обработка на данните от клиента
2. извличане на информация

Разширение:
1. Актуализация на данните за клиента
2. Поставяне на етикет за адрес (за изпращане на информация)

Алтернативи:
1. Включване на нов клиент

use case = последователност от действия

Схема за use cases (1)

- **Use Case:** Име (Какво ще прави?)
  - Цел: Основната цел, която ще бъде постигната при успешно изпълнение на use case
  - **Категория:** основна, второстепенна, или незадължителна
  - **Предусловия:** Очаквано състояние предшестващо изпълнението на use case
  - **Успешни следусловия:** Очаквано състояние след успешно изпълнение на use case
  - **Неуспешни следусловия:** Очаквано състояние, когато целта не е достигната
  - **Актьори:** Роли на хора или други системи, които вземат участие в use case

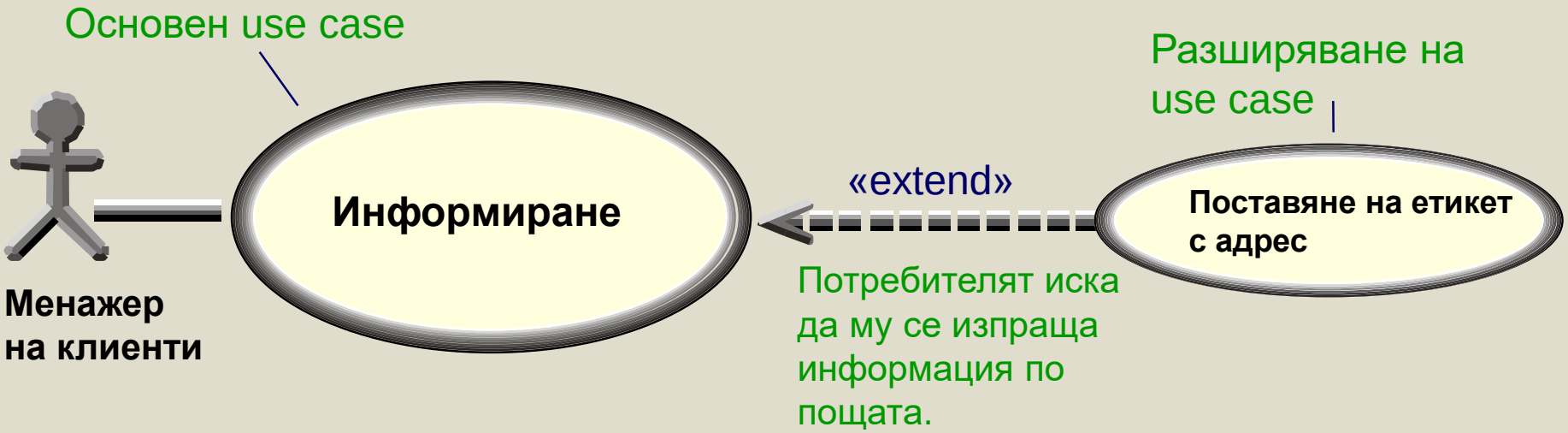
Схема за use cases (2)

- Първоначални събития: Когато това събитие се случи use case ще се инициализира
- Описание:
  - 1 Първо действие
  - 2 Второ действие
- Разширение:
  - 1A Разширение (upgrade) на първото действие с възможност на функция
- Алтернативи:
  - 1A Алтернативно изпълнение на първото действие
  - *1B Допълнителни алтернативи за първото действие.*

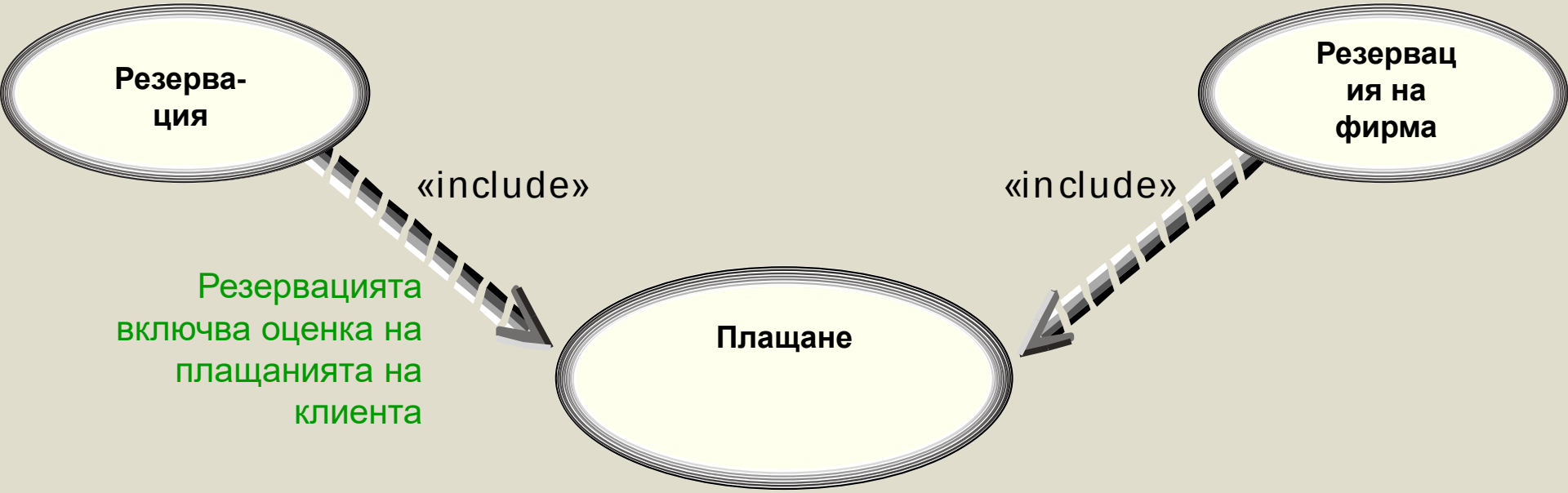
Връзки между Use Cases (UML)

- extend - връзка
- include - връзка
- generalize - връзка

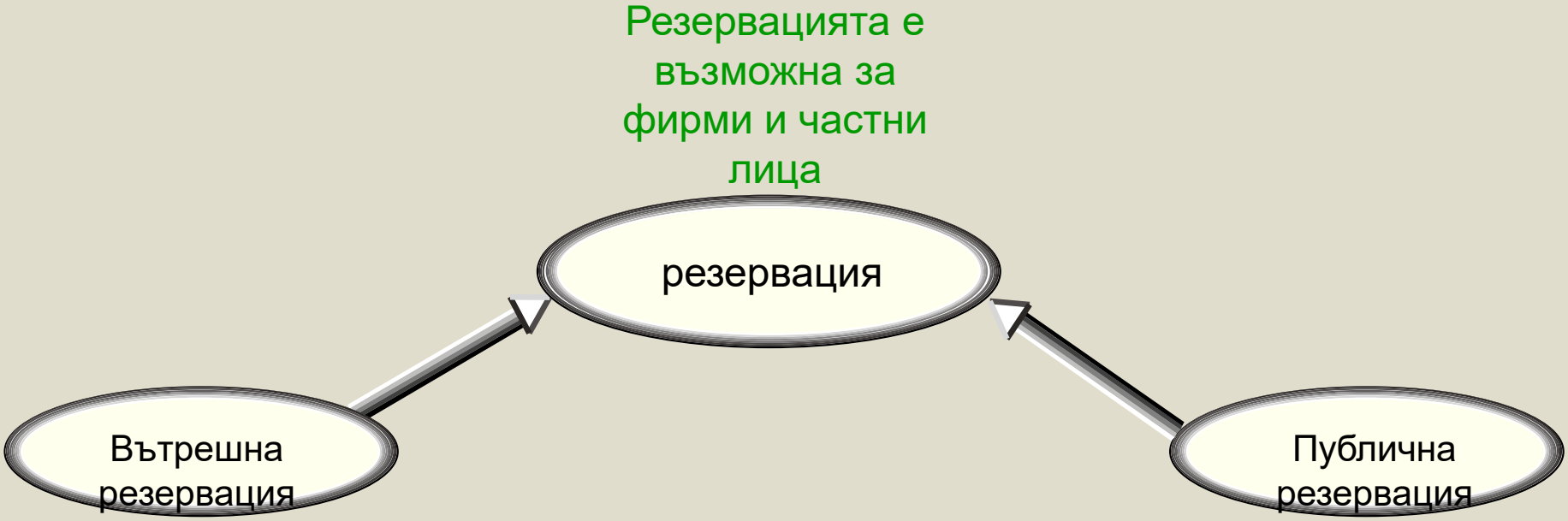
extend - връзка



include - връзка



Generalize - връзка



Пример: use case диаграма с всички видове връзки

Use Case в сравнение с Функция

- Use cases са *специални функции на продукта*
- Use cases са ориентирани към *основните задачи на системата*, които са организирани, чрез взаимодействащи си процеси между потребителя и системата.
- Use cases имат *стойност* за потребителя
- Пример „организация на семинар“:
  - Версия 2.3: *30* самостоятелни функции
  - Версия 3.0: *8* use cases

Use Case в сравнение с Функция: Пример

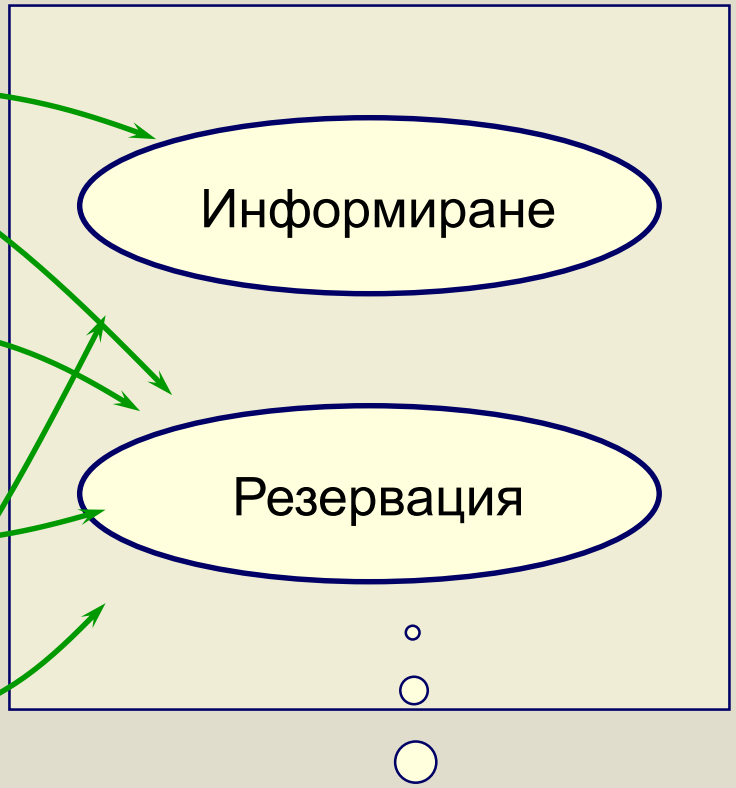
V2.3c 30 самостоятелни функции

/F 10/
Регистрация на клиент, редактиране и изтриване (клиент = участник /заинтересован)

/F 20/
Регистрация на клиент с верификация

/F 60/
Изпращане на поща за потвърждение на регистрация

/F 130/
Отделяне на етикет с адрес.



Елементарни функции:

- нямат директна стойност за потребителите
- имат стойност само като част от use case

главни системни функции

Оценка

+ Фокусират се върху основния работен поток, а не върху елементарни функции
+ Концентрират се върху стандартните работни потоци
+ Лесни за разбиране дори от клиента
– Има опасност да се впуснем в твърде големи детайли