Моделиране на процеси
UML

Упражнение 2

- Какво е моделиране
- Видове диаграми
- Основни нотации (обозначения)
- Примери (creately.com или средство по избор)

Моделиране

- Моделиране означава да създадем абстракция на действителността
- Абстракциите представляват опростен образ на реалността:
  - Те игнорират несъществените детайли
  - Запазват само съществените
- Кое е *съществено* и кое *несъществено* зависи от предназначението на модела

Модели, системи, изгледи

- *Моделът* (model) е абстракция, която описва подмножество от системата
- *Изгледът* (view) описва избрани аспекти от модела
- *Нотациите* (notations) са множества от графични и текстови правила за представяне на изгледите
- Изгледите и моделите на дадена система могат да се припокриват

изобразява се чрез

Модели, системи, изгледи

- Примери
  - Система: самолет
  - Модели:
    - Симулатор на полета
    - Аеродинамичен модел
  - Изгледи:
    - Изглед на електрическата система
    - Изглед на двигателната система
    - Изглед на отоплителната система

Модели на софтуерните системи

- Функционален модел:
  - Use case диаграми
- Обектен модел:
  - Class диаграми
- Динамичен модел:
  - Sequence диаграми
  - Statechart диаграми
  - Activity диаграми

UML

- **унифициран език за моделиране** (***Unified Modeling Language,***)
- графичен език за визуализиране, специфициране, конструиране и документиране на елементите на една софтуерна система
- актуална версия на езика: UML 2.2.
- http://www.uml.org/

UML: основни нотации

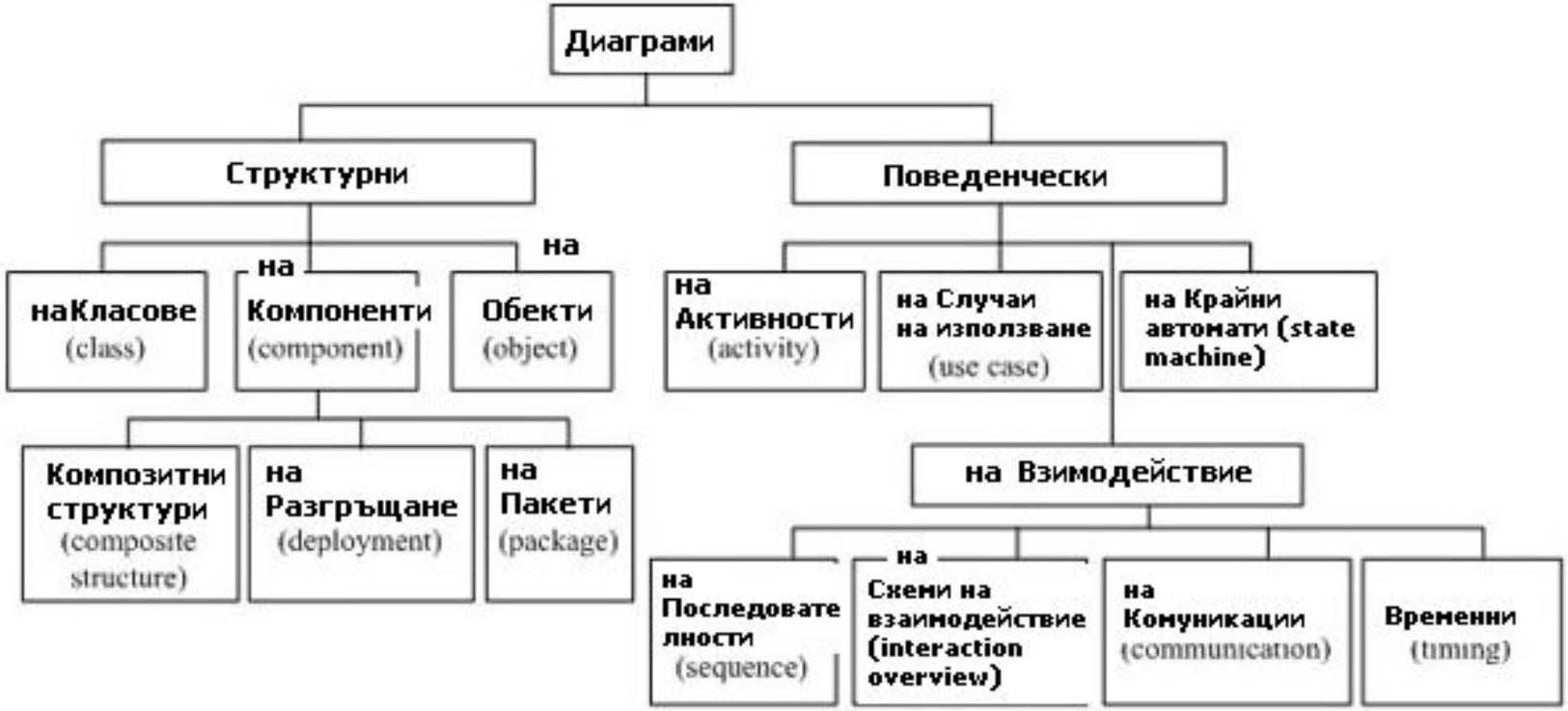
- Правоъгълниците са класове или инстанции на класове
- Елипсите са функции или случаи на употреба (use cases)
- Инстанциите се означават с подчертани имена, например:
  - `myWatch:SimpleWatch`
  - `Joe:Firefighter`

UML: основни нотации

- Типовете (класове, интерфейси и т.н.) са без подчертани имена, например:
  - `SimpleWatch`
  - `Firefighter`
- Диаграмите са графи
  - Върховете им са обекти от моделираната област (entities)
  - Дъгите са връзки между обектите

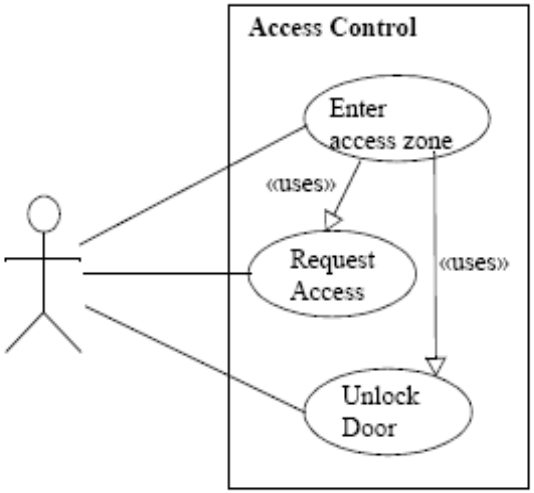
Диаграми в UML

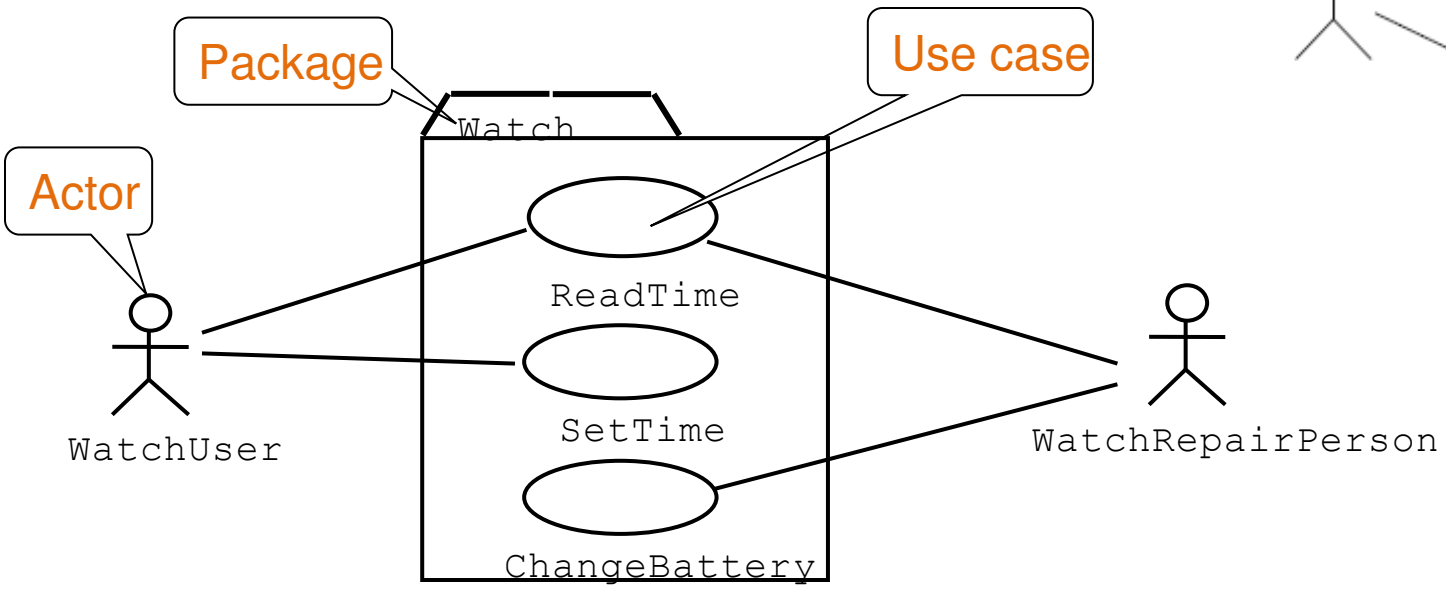
1. Class diagram (класова диаграма)
2. Component diagram (компонентна диаграма)
3. Composite structure diagram (диаграма на съставна структура)
4. Deployment diagram (диаграма на разгръщане)
5. Object diagram (обектна диаграма)
6. Package diagram (диаграма на пакетите)
7. Activity diagram (диаграма на дейност)
8. State Machine diagram (диаграма на машина на състоянията)
9. Use case diagram (диаграма на типичните случаи на употреба)
10. Communication diagram (комуникационна диаграма)
11. Interaction overview diagram (UML 2.0) (диаграма за преглед на взаимодействие)
12. Sequence diagram (диаграма на последователност)
13. UML Timing Diagram (UML 2.0) (времева диаграма)
14. UML Profile Diagram


Диаграми
Структурни
Поведенчески
наКласове (class)
на Компоненти (component)
на Обекти (object)
на Активности (activity)
на Случаи на използване (use case)
на Крайни автомати (state machine)
Композитни структури (composite structure)
на Разгръщане (deployment)
на Пакети (package)
на Взимодействие
на Последователности (sequence)
на Схеми на взаимодействие (interaction overview)
на Комуникации (communication)
Временни (timing)

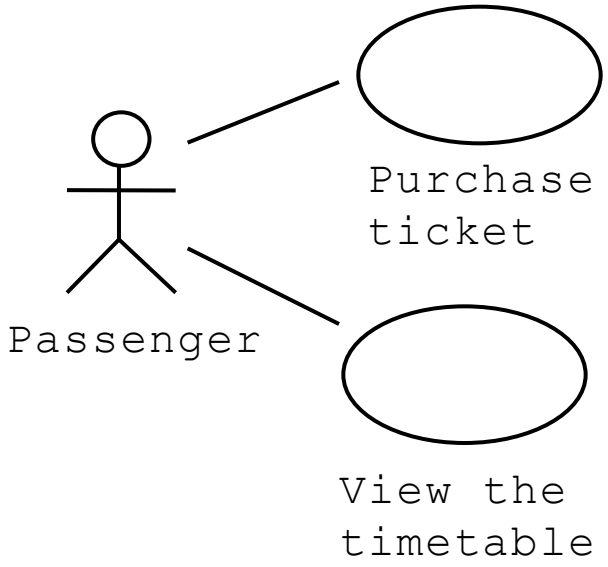
Use case диаграми (случаи на употреба)

- Описват поведението на системата от гледна точка на потребителя
- Какво може да правят различните видове потребители





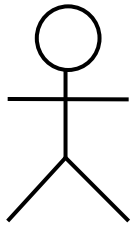
Use Case диаграми



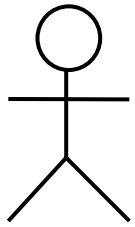
- Използват се при извличане на изискванията за описание на възможните действия
- *Актьорите (Actors)* представят роли (типове потребители)

- *Случаите на употреба (Use cases)* описват взаимодействие между актьорите и системата
- Use case моделът е група use cases
  - Предоставя пълно описание на функционалността на системата

Актьори (Actors)



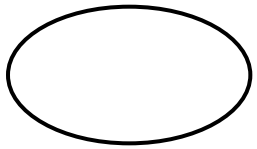
Passenger



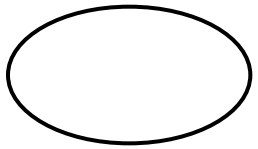
GPS satellite

- Актьорът е някой, който взаимодейства със системата
  - потребител
  - външна система
  - външната среда
- Актьорът има уникално име и евентуално описание
- Примери:
  - Пътник: човек във влака
  - GPS сателит: предоставя GPS координати

Use Case



Purchase ticket



View the timetable

- Един use case описва една от функционалностите на системата
- Състои се от:
  - Уникално име
  - Свързан е с актьори
  - Има входни условия
  - Съдържа поток от действия (процес)
  - Има изходни условия
  - Може да има и други изисквания

Use Case – пример

Име: `Purchase ticket`

Участващи актьори:

`Passenger`

Входни условия:

- `Passenger` е пред продавача на билети
- `Passenger` има достатъчно пари за билет

Изходни условия

- `Passenger` има билет

Поток от действия (описание на процеса):

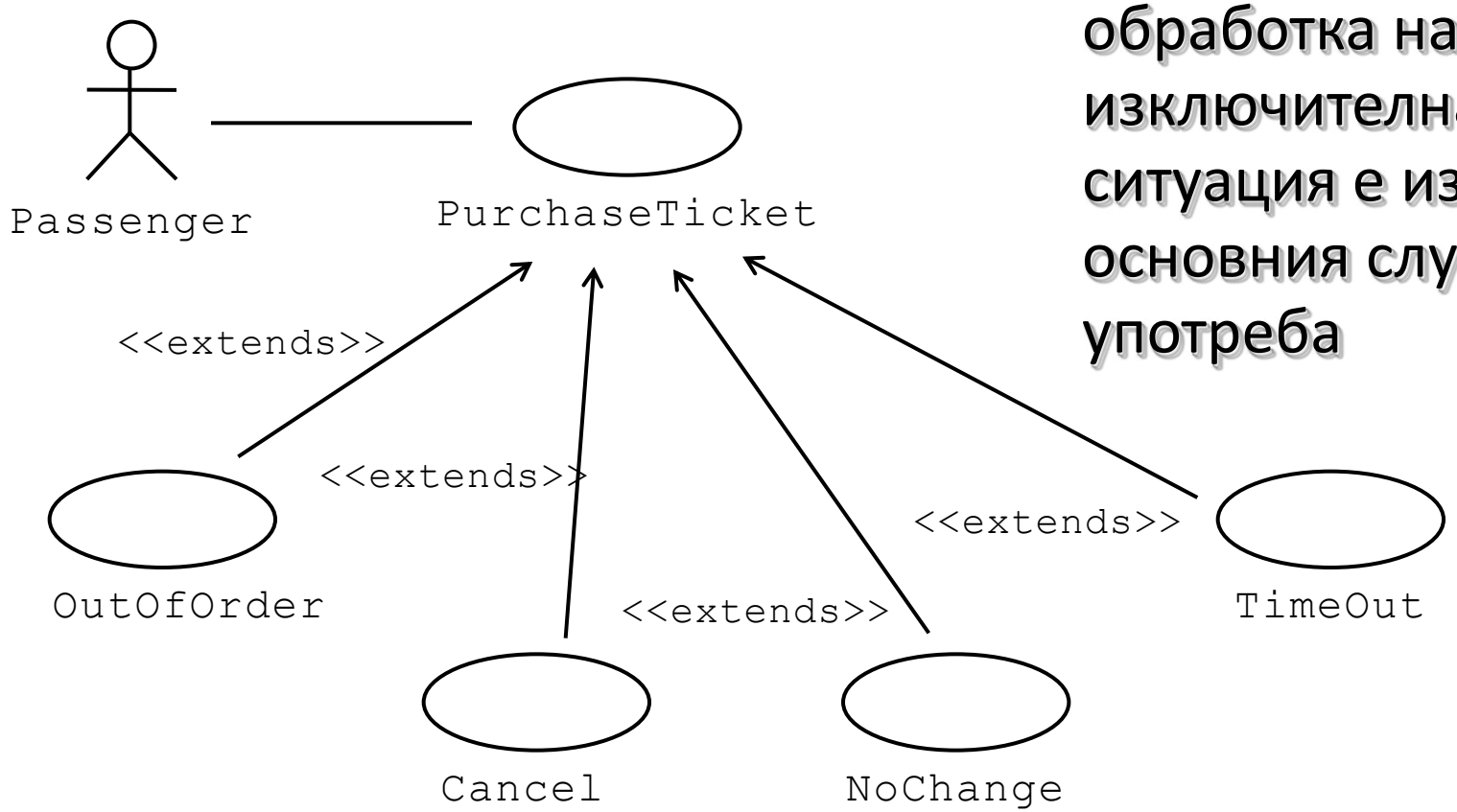
1. `Passenger` избира крайна гара
2. Продавачът съобщава дължимата сума
3. `Passenger` подава пари
4. Продавачът връща ресто
5. Продавачът издава билет

Изпуснахме ли нещо?

Случаите на проблем!

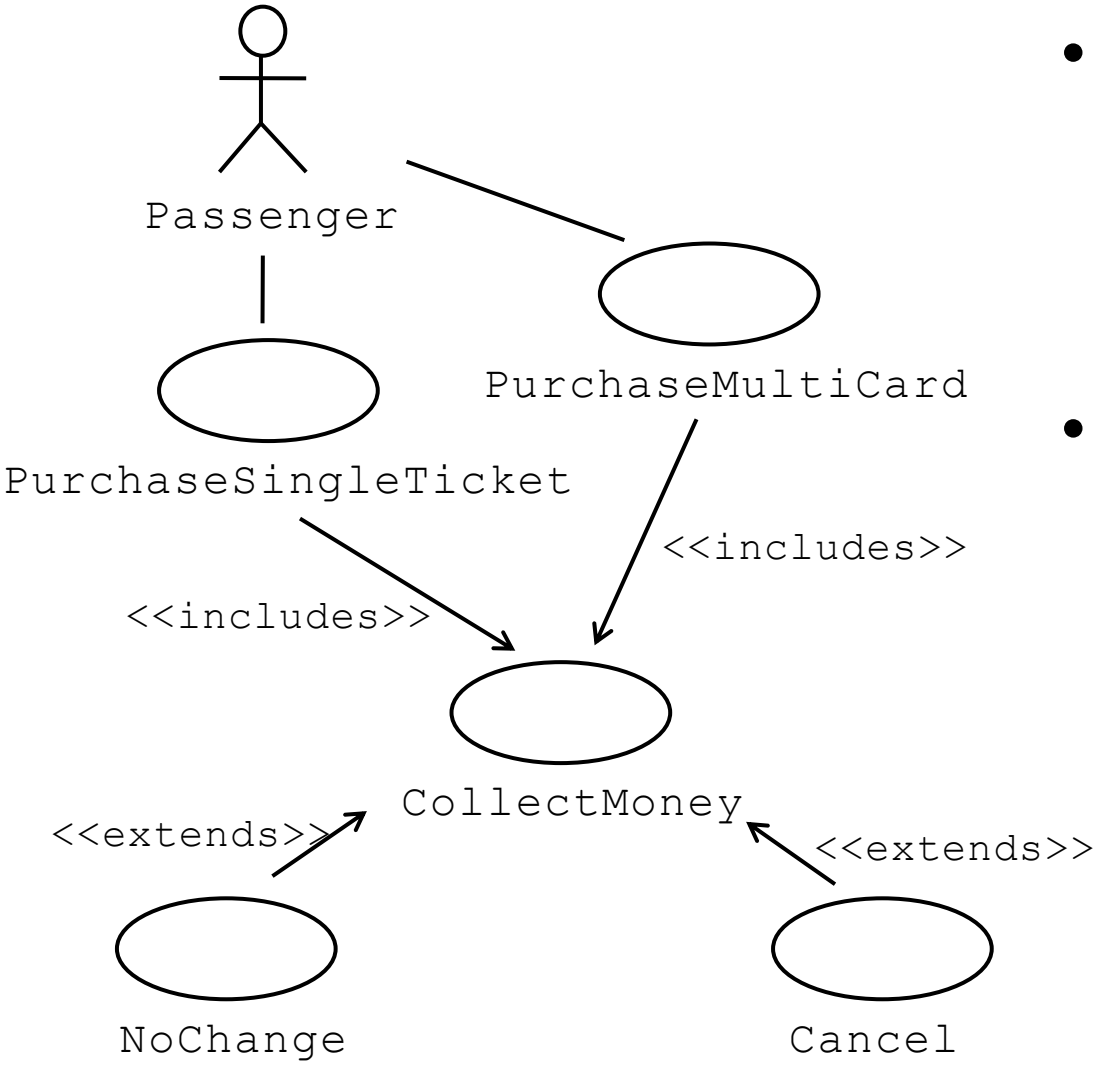
Релацията <<extends>>

- <<extends>> представя изключение или рядко възникващ случай



- Процесът за обработка на изключителна ситуация е извън основния случай на употреба

Релацията *<<includes>>*

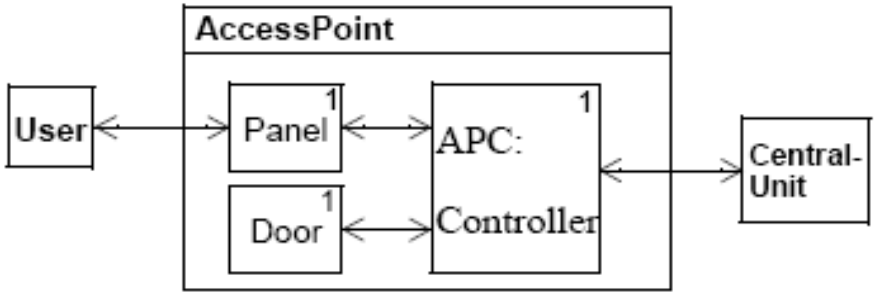


- <<includes>> е поведение, изваден извън даден use case
- Позволява преизползване на споделена функционалност

Class диаграми

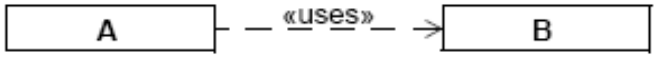
Описват класовете (атрибути и методи) и връзките между тях

– Асоциация – обикновено бинарна

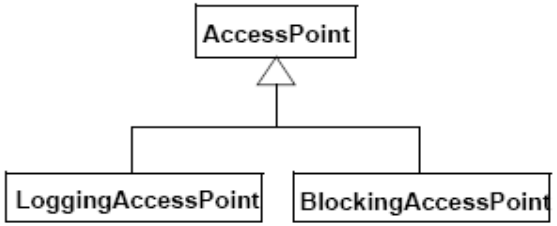


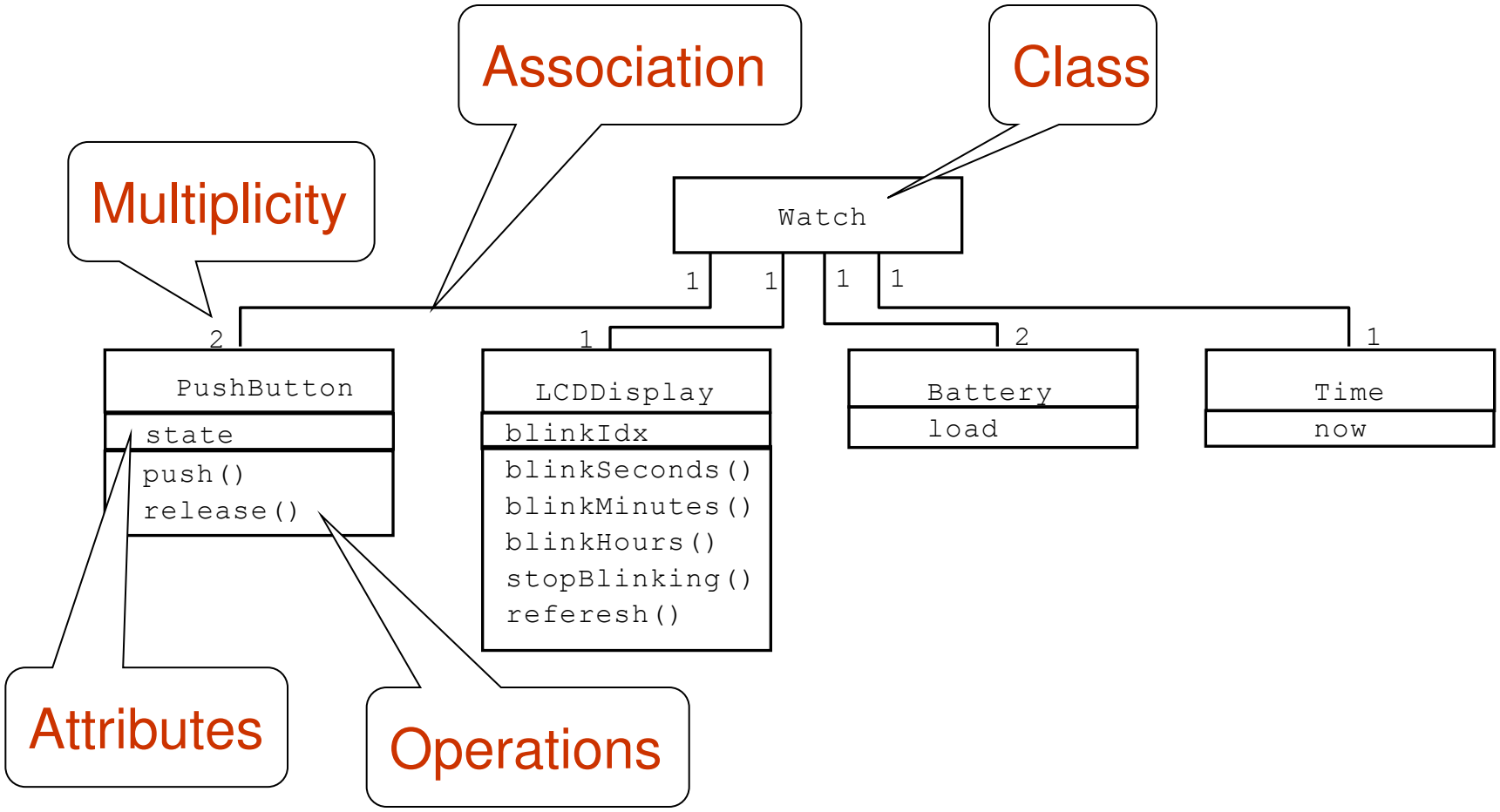
– Агрегация и композиция

– Зависимост

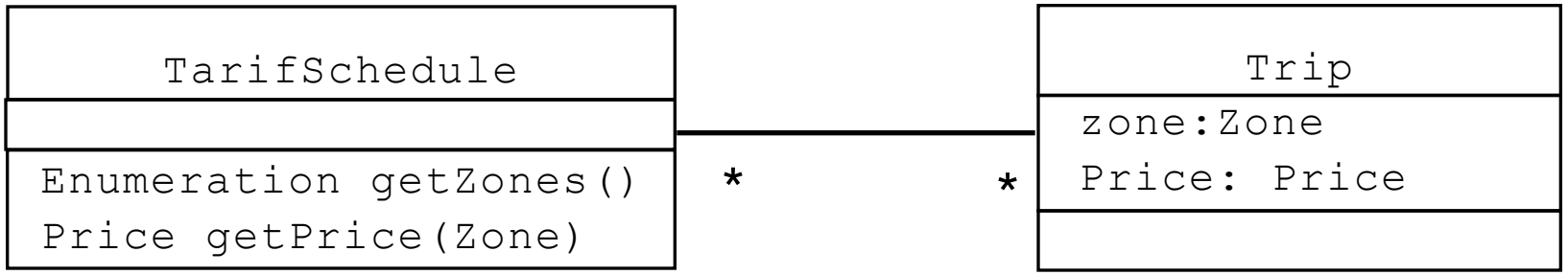


– Генерализация (наследяване)



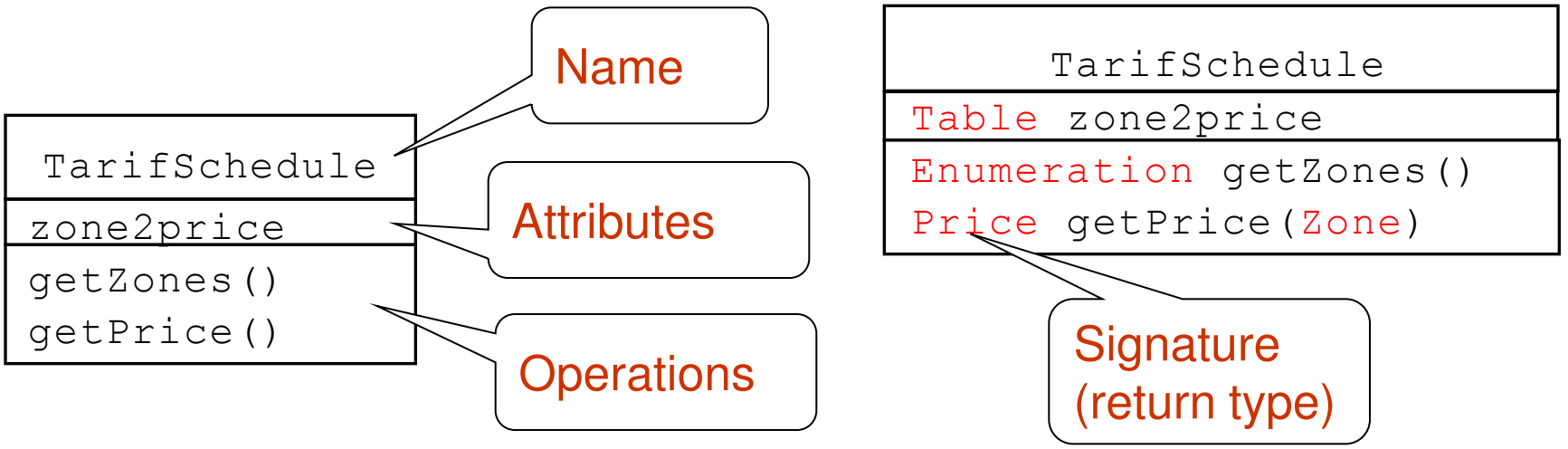

Association
Class
Multiplicity
Watch
1
1
1
1
2
1
2
1
PushButton
state
push()
release()
LCDDisplay
blinkIdx
blinkSeconds()
blinkMinutes()
blinkHours()
stopBlinking()
referesh()
Battery
load
Time
now
Attributes
Operations

Class диаграми



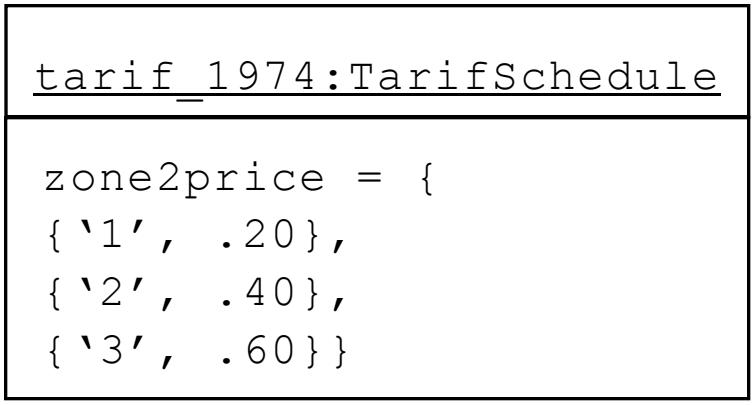
- Описват структурата на системата
- Използват се:
  - По време на анализиране на изискванията за моделиране на обектите от реалния свят
  - По време на дизайна за моделиране на подсистемите
- Дефинират класове и връзки между тях

Класове



- *Класът* е същност от реалния свят
- Има *име*, състояние *(атрибути)* и поведение *(операции)*
- Всеки атрибут има *тип*
- Всяка операция има *сигнатура* (връщан тип)

Инстанции

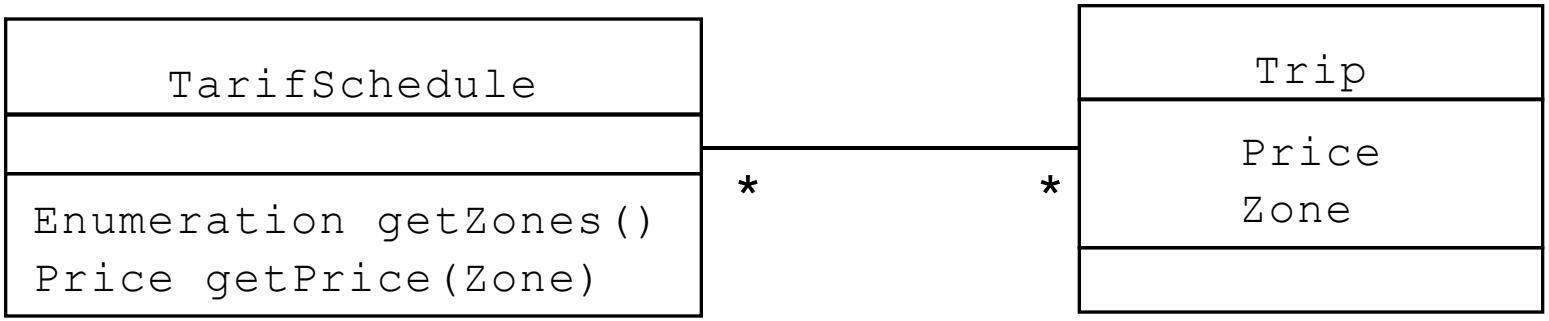


- *Инстанцията* е конкретен екземпляр от даден клас (обект)
- Имената на инстанциите са подчертани и могат да съдържат класа
- Атрибутите се задават заедно със стойностите си

Актьори, класове и инстанции

- Актьор:
  - Външен за системата обект, който взаимодейства с нея, например "пътник"
- Клас:
  - Абстракция, моделираща същност от реалния свят, част от системата, например "потребител"
- Обект:
  - Специфична инстанция на клас, например "пътникът Бай Киро"

Асоциации

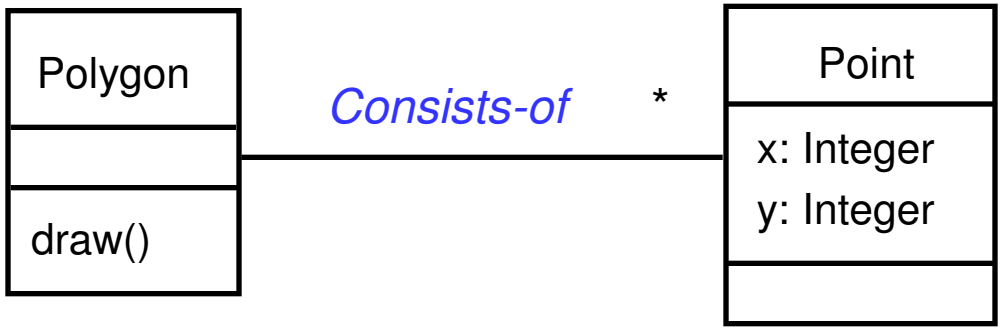


- Асоциациите представляват връзки между класовете
  - Моделират взаимоотношения
- Могат да дефинират множественост (1 към 1, 1 към много, много към 1, 1 към 2, ...)

Асоциации 1-към-1 и 1-към-много

- Асоциация 1-към-1:

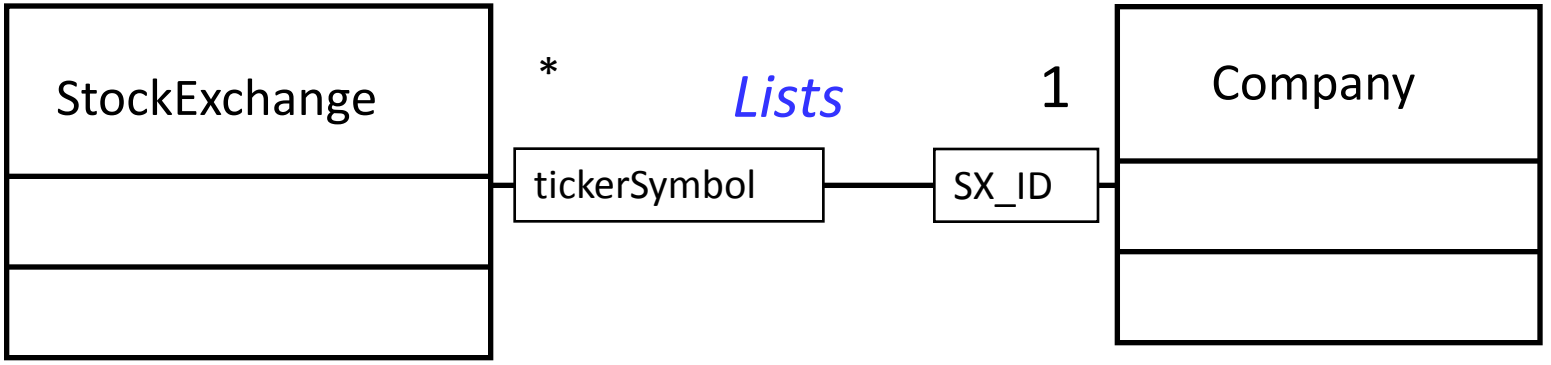
- Асоциация 1-към-много:



Асоциации много-към-много

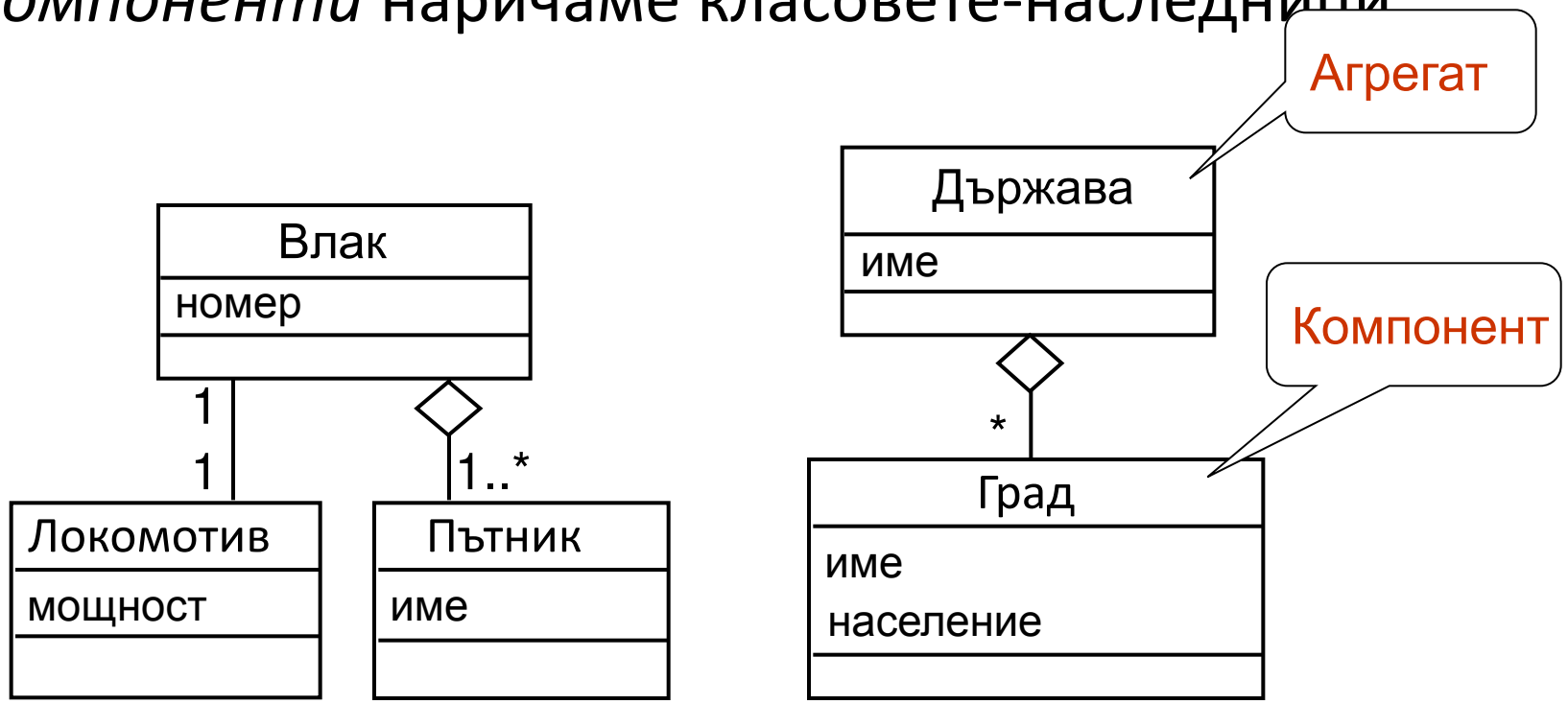
- Асоциация много-към-много:

- Асоциация много-към-много по атрибут:



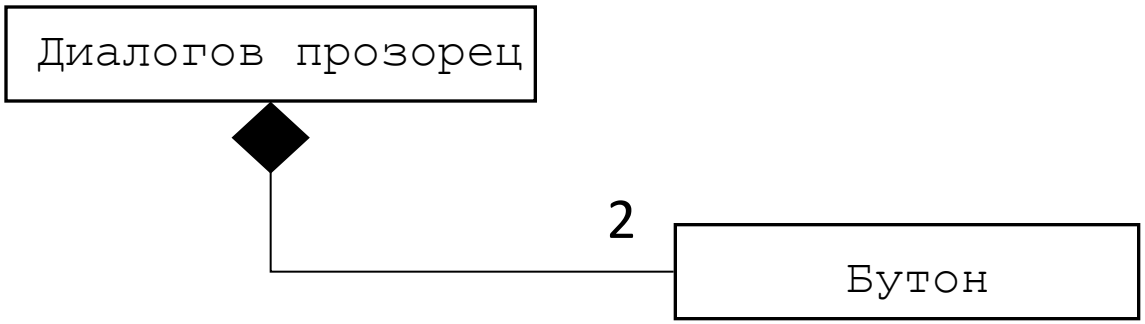
Агрегация

- Агрегацията е специален вид асоциация
- Моделира връзката "цяло / част"
- *Агрегат* наричаме родителския клас
- *Компоненти* наричаме класовете-наследници



Композиция

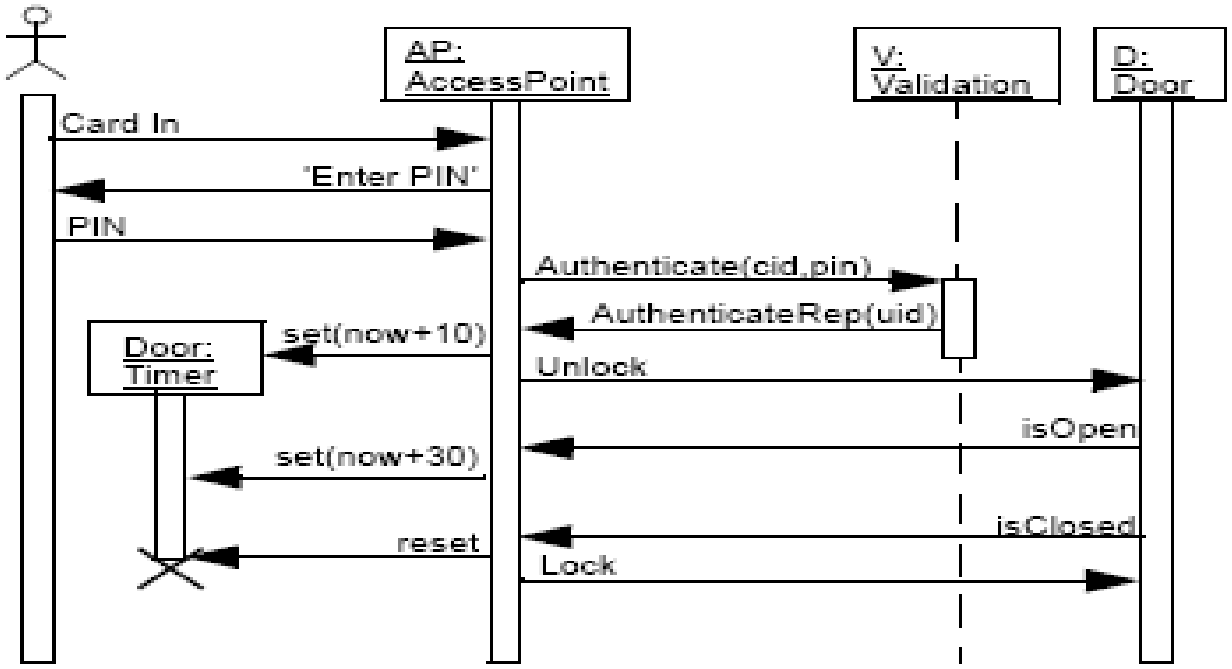
- Запълнен ромб означава *композиция*
- Композицията е агрегация, при която компонентите не могат да съществуват без агрегата (родителя)



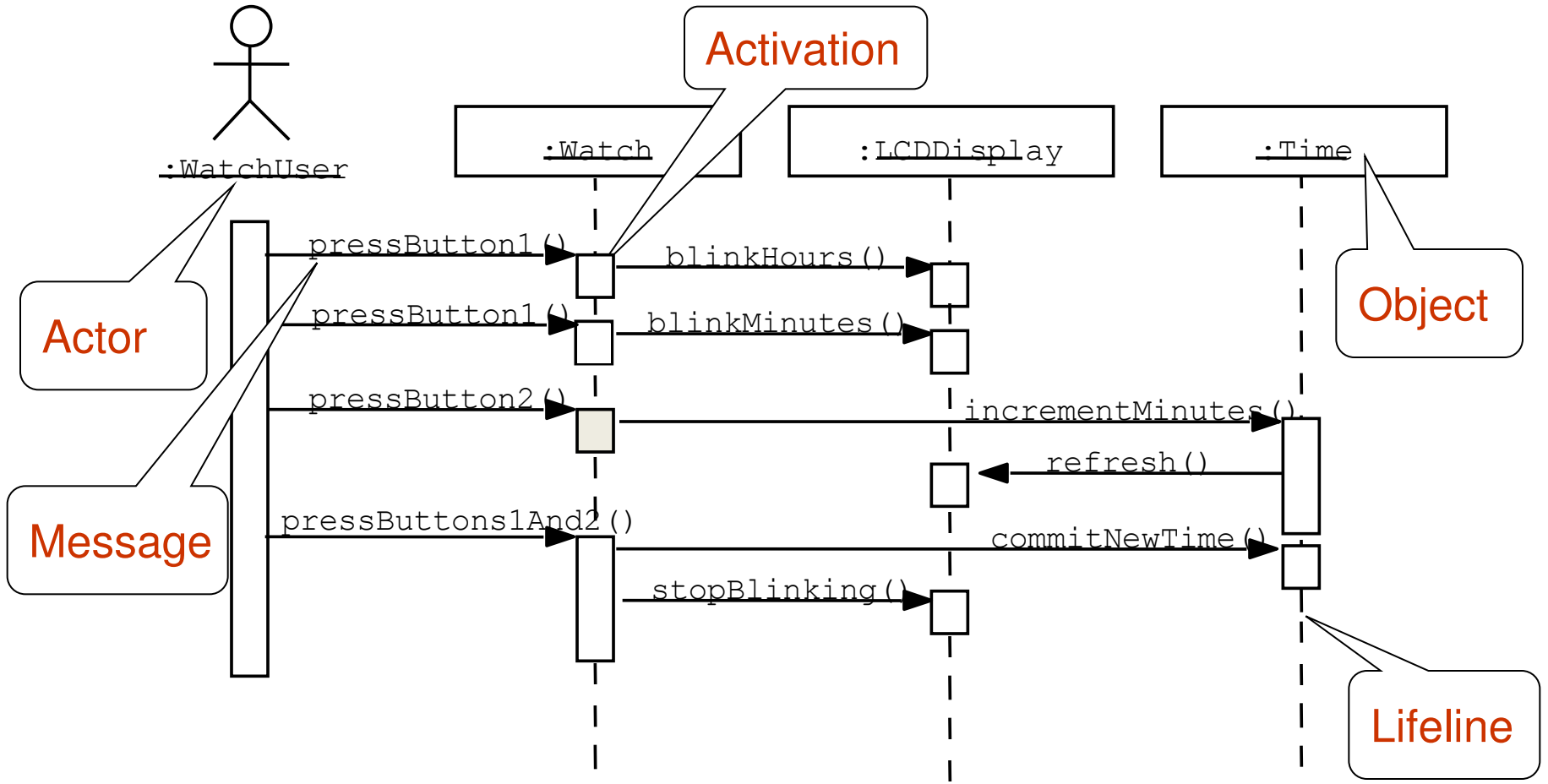
Sequence диаграми (диаграми на последователностите)

- Описват схематично взаимодействието между потребителите и системата
- Отделните действия са подредени във времето
- Основни елементи
  - Обекти с тяхната продължителност на живот
  - Съобщения между тях във времето

Sequence диаграми



Sequence диаграми



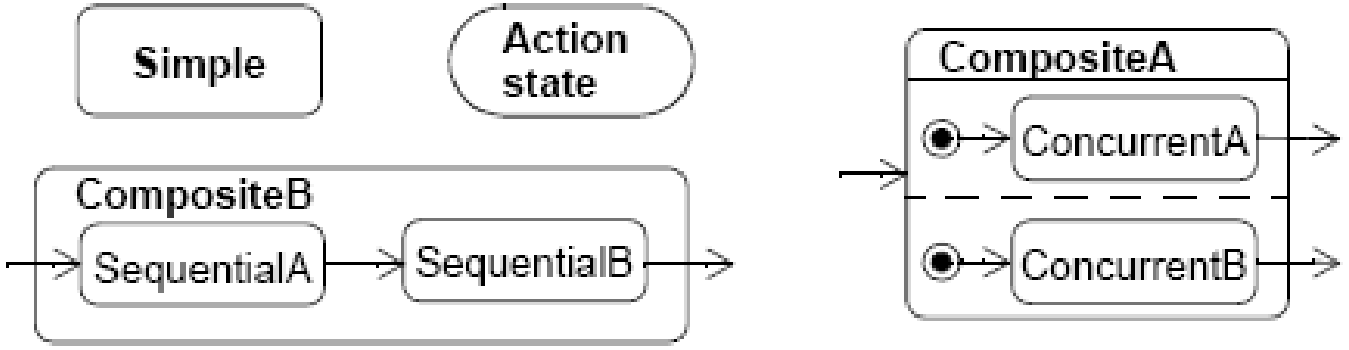
Диаграма на състоянията

- Диаграмата е свързана с клас или метод и показва:
  - Състоянията на един обект (или взаимодействие)
  - Реакцията на обект на стимули (събития) като действия или отговори
  - Описва възможните състояния на даден процес и възможните преходи между тях

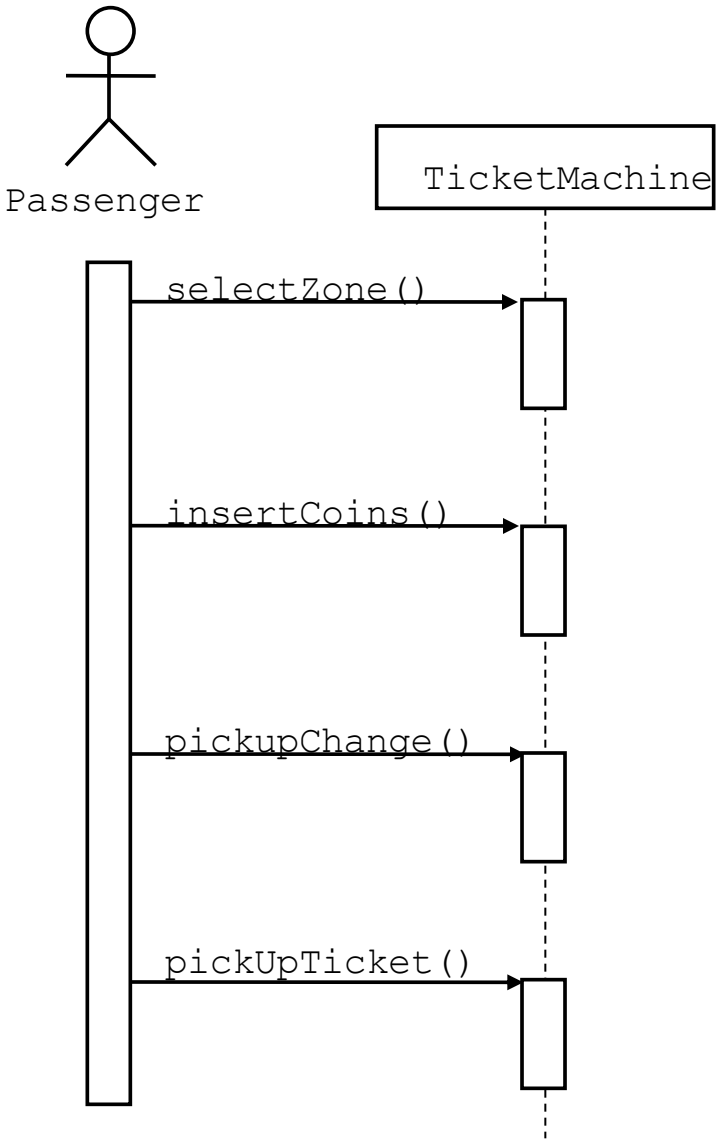
Диаграма на състоянията (Statechart)

- Диаграмата е свързана с клас или метод и показва:
  - Състоянията на един обект (или взаимодействие)
  - Реакцията на обект на стимули (събития) като действия или отговори
- Основни понятия
  - Състояние – което:
    - удовлетворява някое условие
    - изпълнява някакво действие или
    - чака някакво събитие

  Едно състояние може да се разложи на няколко паралелни или взаимно изключващи се подсъстояния

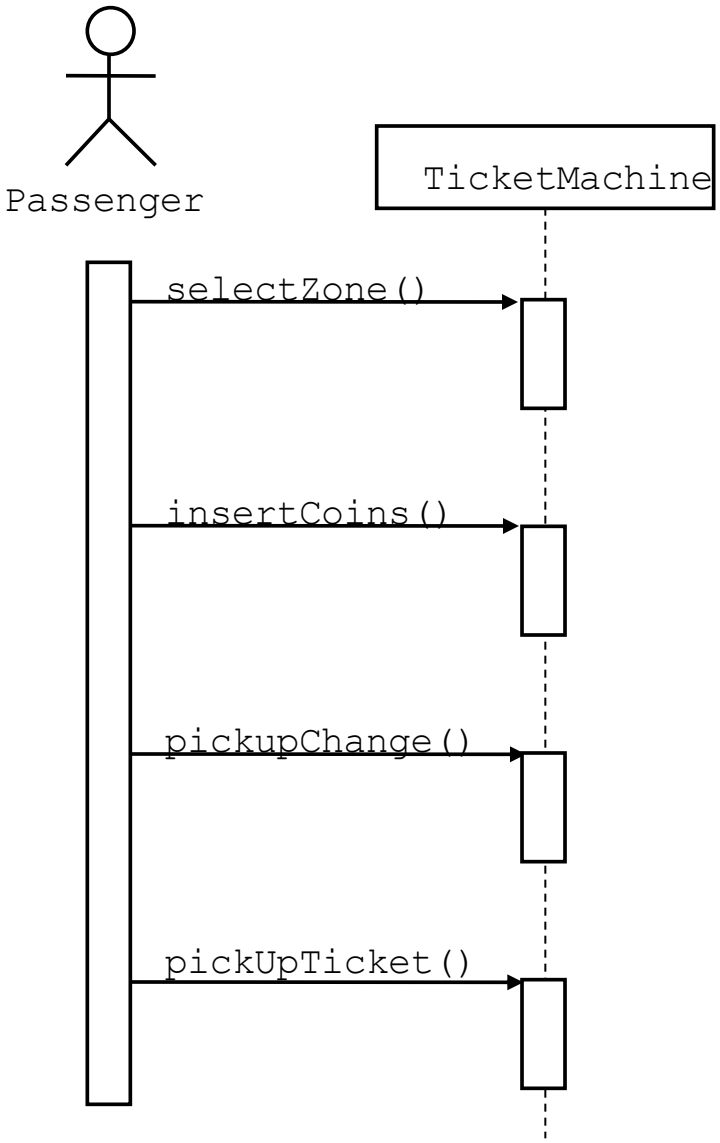


Sequence диаграми



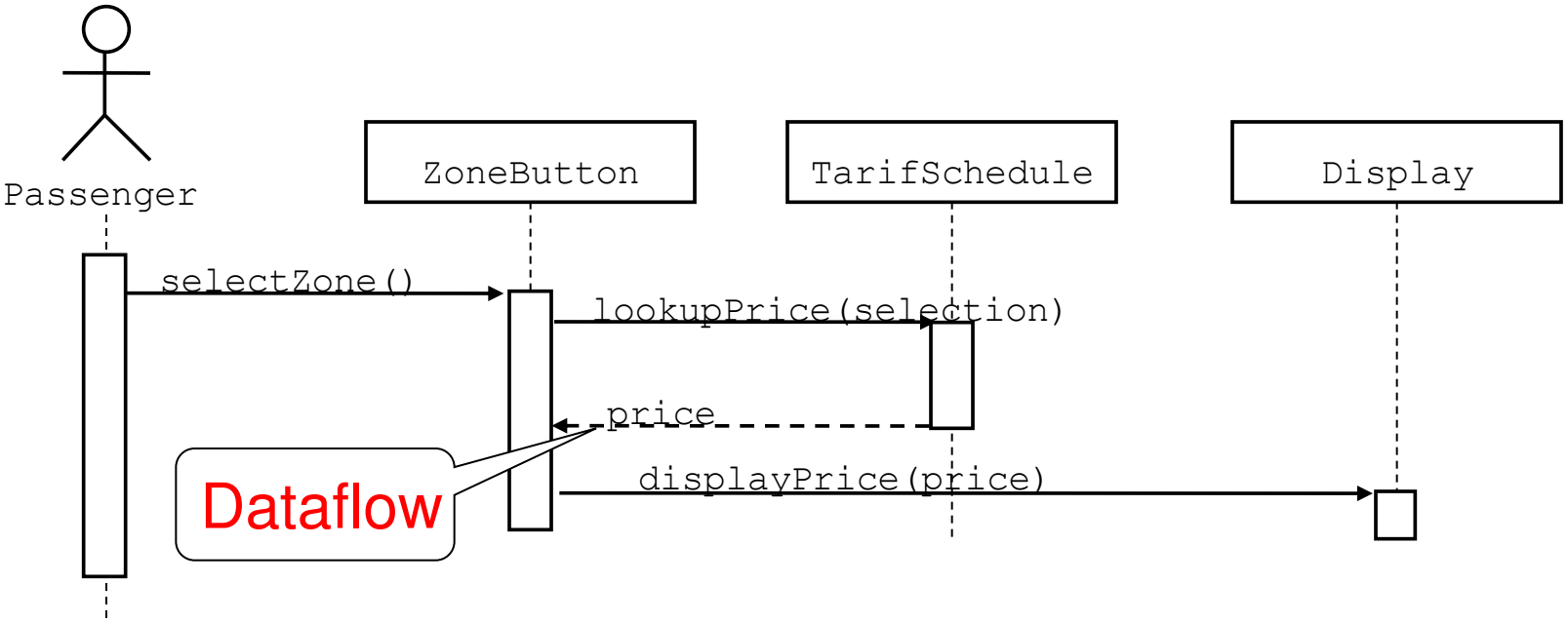
- Използват се при моделиране на изискванията
  - За по-добро описание на use case сценариите
  - Позволяват описание на допълнителни участници в процесите
- Използват се при дизайна
  - За описание на системните интерфейси

Sequence диаграми



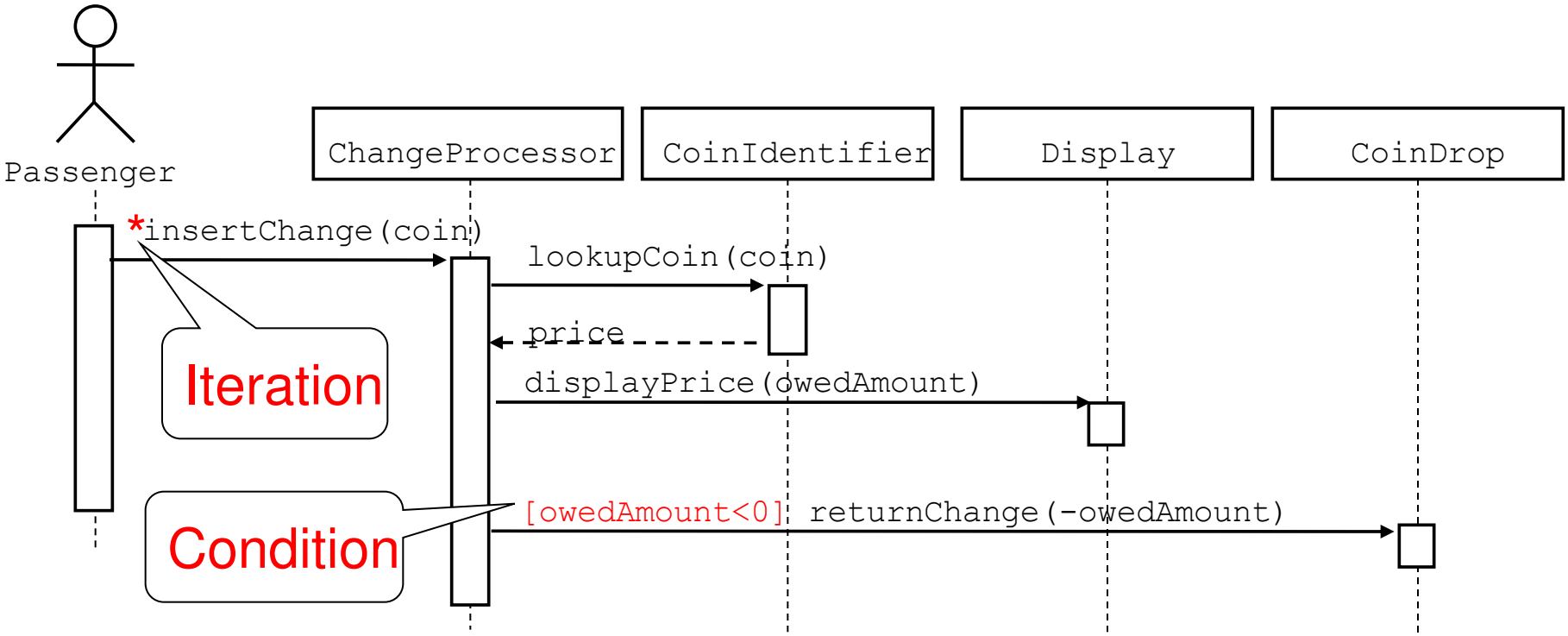
- *Класовете* се представят с колони
- *Съобщенията* (действията) се представят чрез стрелки
- *Участниците* се представят с широки правоъгълници
- *Състоянията* се представят с пунктирана линия

Съобщения



- Посоката на стрелката определя изпращача и получателя на съобщението
- Хоризонталните прекъснати линии изобразяват потока на данните

Итерация и условия



- Итерацията се означава с * преди съобщението
- Условията се означават с булев израз в [ ]