

## Диаграми на взаимодействия- Резюме

Показват как обектите си взаимодействат чрез изпращане на съобщ.

### Видове диаграми на взаимодействия:

1/Диаграми на **последователност**: Показват **хронологичния ред на съобщенията** между обектите в реално време и за сложни с-ми

2/Диаграми на **комуникация** (колаборация/сътрудничество): Показват **вързките между обектите и съобщенията**, които те си изпращат, въздействия върху обектите и алгоритъм.

3/**Interaction Overview** /преглед на взаимодействието/: Показват контролния поток с **възли, които могат да съдържат диаграми на взаимодействия** или взаимодействие (ref).

4/ **Диаграми на времето**: разглеждат условията, които се променят в рамките на и между жизнените линии по линейна времева ос.

**Диаграми на последователност** - Описват модел на взаимодействие между обекти, подредени в хронологичен ред. Включват :

**Съобщения**: Показват комуникацията между обектите с очакване за активност. Могат да бъдат **синхронни** /solid arrowhead; with an **return** message- dashed line/ или **асинхронни** /line arrowhead/; Self and Recursive/Reentrant, Callback/; Lost and Found

**Жизнени линии**: Представят съществуван. на обекта в опр. момент.

**Съобщения за създаване и унищожаване**: Показват създаването или унищожаването на обекти по време на взаимодействието.

**Фрагменти** в диаграмите на последователност:

**alt** (алтернативи): Моделират if...then...else конструкции.

**opt** (опция): избор за изпълнение на фрагмент в зависим. от условие.

**loop** (цикъл): Моделират итерации с определен брой повторения.

**break** (прекъсване): Моделират изкл. последователност от събития.

**par** (паралелно): Моделират паралелно изпълнение на поведения.

**strict** (строго): Изискват строго подреждане на операндите.

**critical**: Критичен регион, където блок от операции/взаимодействия се изпълняват атомарно, т.е. без намеса от др. нишки/ процеси

**ignore**: Игнорират незнач. съобщения – ignore{get, set}

**consider {add, pop}**: съобщ. да се вземат предвид, а др.се игнорират

**assert** (утвърждение): утвърждение, че последователностите на операнда са единствените валидни продължения.

**neg**: Описват невалидни комбинирани фрагменти

**ref**: фрагмент за взаимодействие, който използва (извиква) др. взаим.

**sd**: gate - свързване съобщ. във фрагмент със съобщ. извън фрагмент

**inter- and intra object messages** - един обект може да има повече от една линия на живот (lifeline), излизаща от него. Показват съобщения между различни обекти (**inter-object**), така и съобщения в рамките на един и същ обект (**intra-object**)

**State Invariant** - ограничение, наложено върху линия на живот (lifeline), което трябва да бъде вярно по време на изпълнение. Изобразява се като правоъгълник със заоблени краища (полукръгли).

**Centralized control** - малък брой обекти **управляват потока**, като изпращат и получават съобщения от други обекти. Тези управляващи обекти определят реда, в който останалите обекти ще бъдат активирани. **Взаимодействието между останалите обекти е минимално** или напълно липсва. Предимства: 1/ Лесно управление

на реда на събитията; 2/ reusable components Подходящ при: 1/ Редът на изпълнение на подсъбитията може да се променя 2/ очаквате да добавяте нови подсъбития 3/ да запазите части от функционалността като отделни, преизползваеми компоненти

**Decentralized control**- при йерархия тип "част от" или "състои се от" или при фиксирана хронологична последователност; да **капсулирате** (скриете детайлите) и по този начин да създадете **абстракции** на функционалността.

**Диаграми на комуникация** (колаборация) - Описват модел на взаимодействие между обекти чрез връзките и съобщенията, които те си изпращат;

**Съдържание:** Включват обекти, връзки и съобщения, които описват как обектите комуникират - **взаимоотношенията между обектите**

**Връзки:** Показват отношенията между обектите, през които се изпращат съобщения.

**Съобщения:** Показват комуникацията между обектите с очакване за активност - маркирана стрелка до връзка:

[sequenceNumber:] methodName(parameters) [: returnValue]

**Диаграми на последователност** са по-добри за показване на **хронологичния ред на съобщенията**, а **диаграми на комуникация**- връзките между обектите и съобщенията, които те си изпращат.

Concurrency – различни процеси се изпълняват едновременно или припокриващо се, но не непременно в един и същи момент

Повторение (Recurrence) - Повторението определя условно или итеративно изпълнение на **нула или повече съобщения**, които се изпълняват в зависимост от зададеното условие.

- **Branch (разклонение):** - условие, което трябва да бъде изпълнено, за да се продължи изпълнението. Например, ако условието е вярно, се изпълнява дадено съобщение.
- **Loop (цикл):** Това означава, че съобщението се повтаря, докато се изпълнява условие или определен брой пъти.