

## Упражнение 10-11. Диаграма на последователностите (sequence diagram) в UML

Цел: Запознаване с предназначението и процеса на построяване на UML sequence диаграми.

### Теоретична част

След като вече са определени случаите на употреба и са моделирани обектите в системата чрез диаграми на класове, може да се разработи динамичното поведение на системата.

- Случаите на употреба описват взаимодействието между потребителите и системата.
- Диаграмите на взаимодействие описват начина по който групи от обекти взаимодействат и съвместно моделират поведение, на единствен случай на употреба.

Диаграмите на взаимодействие пуказват обектите в системата и съобщенията, които се разменят между тях. Диаграмите на последователност и на кооперации (еволуиращи в диаграми на комуникации в UML 2.0) са диаграми на взаимодействие в UML: те са предназначени да покажат **особеностите на взаимодействие на елементите на моделираната система, с цел пълно или частично изпълнение на някакъв случай на употреба. В диаграмите на кооперации, в явен вид времето отсъства. Тъй като временният аспект на поведение има съществено значение при моделирането на синхронни процеси описващи взаимодействието на обектите, в UML са въведени диаграмите на последователностите.**

И така: има два вида взаимодопълващи се UML диаграми на взаимодействия, за които Star UML предлага конвертиране от един в друг вид:

- **диаграма на последователностите (sequence diagram)** – диаграма на взаимодействията, в която основен акцент се поставя върху реда на обмяната на съобщения във времето.
- **диаграма на кооперациите** - диаграма на взаимодействията, в която основен акцент се поставя върху структурната организация на обектите, които изпращат и получават съобщения.

Забележка: В UML 2.0 е въведена още една диаграма на взаимодействие: **временна диаграма (timing diagram)**.

### Графични елементи на диаграмите на последователностите

Диаграма последователности (sequence diagram) - диаграма, на която са показани взаимодействията между обектите, подредени според времето на появяване. Тоест, основни елементи на тези диаграми са инстанции на класовете и съобщения (наричани „стимули”) между тях.

Тази диаграма има като че ли 2 измерения:

- Едното – от ляво надясно във вид на вертикални линии, всяка от които изобразява линията на живот на отделен обект, участващ във взаимодействието.
- Второто измерение - вертикалната временна ос, насочена отгоре надолу.

### Забележка:

За съжаление, както чувате често по време на този курс, в UML няма еднозначност. Някои източници разграничават «диаграма на последователностите с роли» (sequence role diagram), базирана на класове с роли и «диаграма на последователностите» (sequence diagram) – базирана, на обекти. Казва се, че обектите изпращат „стимули” един на друг, докато ролите изпращат «съобщения». Има пет вида съобщения/стимули: **Call, Send, Return, Create** и **Destroy**.

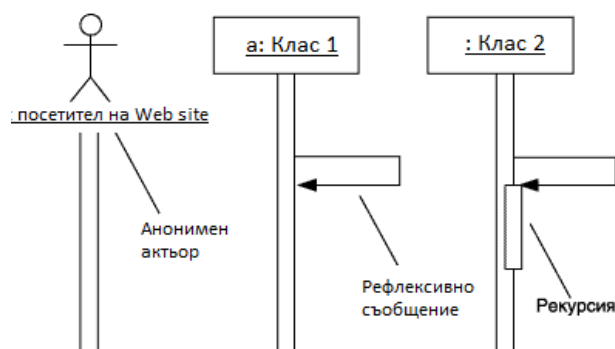
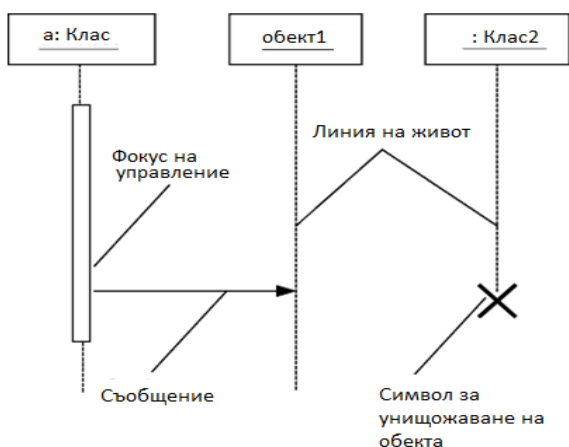
Примерите, които ще пуказване и разработваме ще са основно базирани на обекти, тъй като това е преобладаващото използване на тези диаграми.

Всеки обект графически се изобразява като правоъгълник и се разполага в горната част на своята линия на живот (рис. 8.1, **фигура 1**). Вътре в правоъгълника се записва името на обекта с малка буква, следвано от името на класа, разделени с дуеточие. При това, целия запис се подчертава, което е отличителен признак на обектите: **instanceName: datatype**

Име на инстанцията може и да не се зададе („анонимен” обект), задължително обаче в този случай трябва да се зададе името на класа (datatype). Ако не се укаже името на класа, тогава задължително се указва името на обекта (обект „сирак”). Роля на класовете в тези диаграми по правило не се указва.

Под името може също така да се изброят атрибутите и техните стойности, важни за този случай на употреба.

Обикновено най-ляво е обекта, инициатор на моделирания процес на взаимодействие (обект „a” на **фигура 1**). Следва – обекта, който непосредствено взаимодейства с инициатора, тоест подредбата на обектите е също важна, от съображение за удобство при визуализацията на взаимодействията между обектите.



**Фигура 1.** Графични елементи на диаграмите на последователностите

**Фигура 2.** Актьор, рефлексивно съобщение и рекурсия в диаграмата на последователностите

На началния момент време съответства **най-горната част на диаграмата**. Процесът на взаимодействие се реализира посредством съобщения, които се изпращат от едни обекти към други. Съобщенията се изобразяват във вид на хоризонтални стрелки, като тези, които са по-горе в диаграмата се предават по-рано от тези, които са по-ниско. При това, мащаб на оста на времето не се указва, модела се само подредба във времето, от вида „по-рано – по-късно”.

**Линия на живот на обекта (object lifeline):** това е вертикалната пунктирна линия на диаграмата на последователността, която представя съществуването на обекта в течение на определен период от време (обект 1 и обект a тип „Клас” на фигура 1 съществуват през цялото време).

За обозначаване на момента на унищожаване на обект се използва символ "X". На фигура 1, този символ се използва за унищожаване на анонимния обект, от тип Клас 2.

Правоъгълникът върху линията на живот на обекта обозначава „активно” състояние на обекта. Този правоъгълник се нарича **фокус на управление (focus of control)** – указва периода време, в течение на който обектът изпълнява някакво действие, тоест периода време, в течение на който обектът се намира в активно състояние.

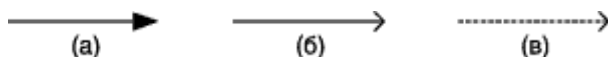
В отделни случаи, инициатор на взаимодействието може да е актьор или външен потребител, тогава той се изобразява като първи обект в диаграмата, със свой фокус на управление (фигура 2).

В отделни случаи, обектът може да изпраща съобщения на себе си (рефлексивни съобщения) – вижте съобщение на обект a:Клас1 във фигура 2. Подобни ситуации възникват, например, при обработка на натиснат клавиш на клавиатурата, при въвеждане на текст в редируем документ и др.

Ако в процеса на рефлексивно съобщение се създаде нов подпроцес или нишка на управление, то се говори за рекурсивен или вложен фокус на управление. На диаграмата на последователност, рекурсията се обозначава като не-голям правоъгълник, свързан с дясната страна на фокуса на управление на този обект, за който се изобразява рекурсивното взаимодействие (анонимен обект на Клас 2 на фигура 2).

### Съобщения

Както казахме, има пет вида съобщения: Call, Send, Return, Create, и Destroy. Съобщенията най-често представляват извиквания (Call) на операции на викания обект. Това са „синхронни” съобщения. При завършването на операцията, извиканият обект връща обратно съобщение към инстанцията, която е инициирала извикването.



**Фигура 3.** Графично изображение на съобщения

Синхронно съобщение (обозначено е на фигура 3.а): извикване на процедура, изпълнение на операция или обозначение на отделни вложени потоци за управление. При такова съобщение получателят може да получи фокуса на управление.

Асинхронно съобщение (обозначено е на фигура 3.б): предава се в произволен момент от време; обикновено при такова съобщение получателят не получава фокуса на управление.

Връщане от извикване на процедура (фигура 3.в).

### Временни диаграми (timing diagrams)

Тези диаграми са разновидност на диаграмите на последователностите и са предназначени за нагледно изобразяване на потока на изменение на състоянието на няколко роли (класове, компоненти). Последните (класове, компоненти) не се изобразяват вертикално, а хоризонтално, и основния акцент е върху нагледното изобразяване на състоянието им, по точно на това как те се променят във времето. Тази възможност е особено полезна при моделиране на вградени системи.

**Виж. „Програмни спецификации . Ръководство за лабораторни упражнения” тема VIII.**

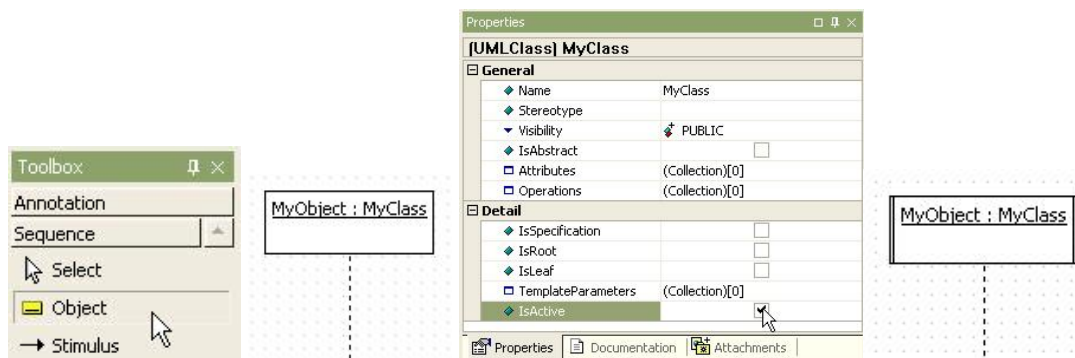
### Литература:

1. <http://old.intuit.ru/department/se/vismodtp/3/>
2. <http://www.cs.sjsu.edu/~pearce/modules/lectures/uml/interaction/SequenceDiagrams.htm>
3. <http://old.intuit.ru/department/pl/umlbasics/8/2.html>
4. <http://old.intuit.ru/department/pl/umlbasics/8/>
5. <http://old.intuit.ru/department/se/vismodtp/3/>
6. <http://www.scribd.com/doc/82434056/19/Component-Diagram>
7. <http://staruml.sourceforge.net/docs/user-guide%28en%29/ch04.html>
8. [http://staruml.sourceforge.net/docs/user-guide%28en%29/ch05\\_3.html](http://staruml.sourceforge.net/docs/user-guide%28en%29/ch05_3.html)
9. <http://www.youtube.com/watch?v=kApq-E2mtn0>

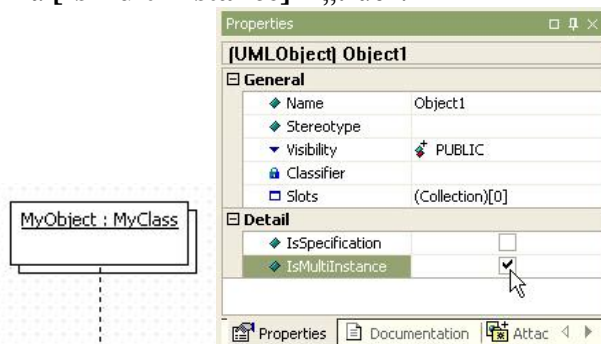
### Приложение: Построяване на диаграма на последователностите със StarUML.

#### Създаването на обект със StarUML е в следните стъпки:

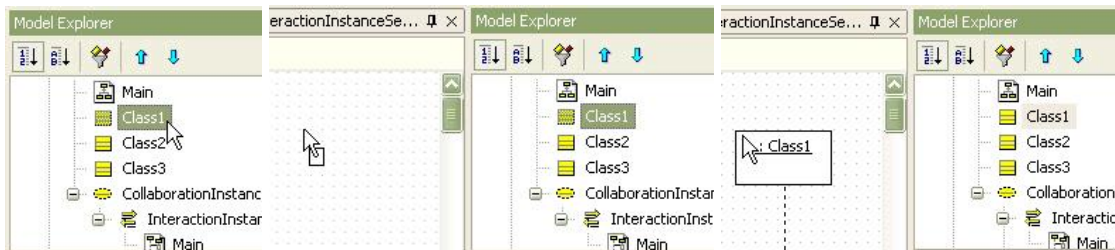
Избира се [Toolbox] -> [Sequence] -> [Object]. Посочва се позицията, в която ще се разположи обекта в [main window] и се въвежда името му. Ако трябва да бъде активен обект, се присвоява на [IsActive] стойност „true”(вдясно). Ако не е присвоена, резултатът е показан вдясно.



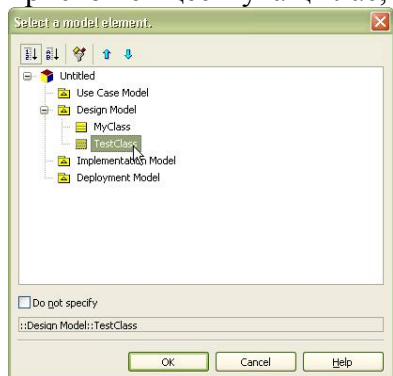
Мулти обект(multi object) се представя така. За да се превърне обекта в мулти обект(multi object) се установява [IsMultiInstance] в „true”.



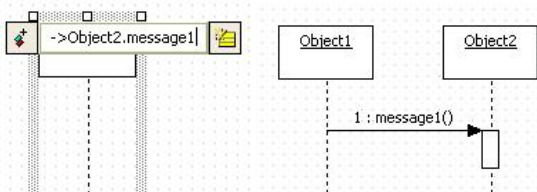
Създаването на обект от клас става от [model explorer] с drag and drop в полето [main window].



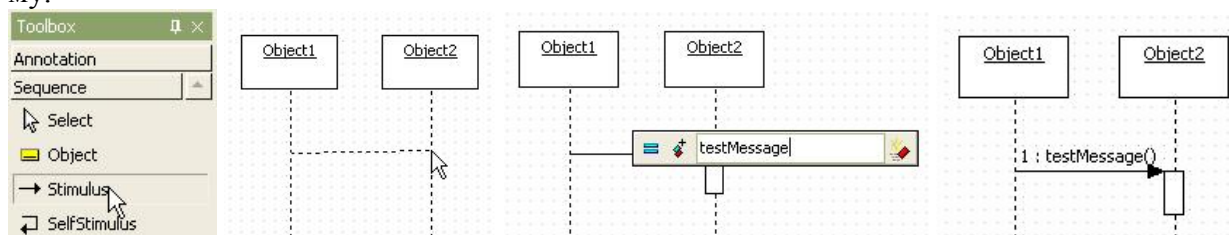
Ако не е присвоен клас към обекта, се избира името на обекта и с двоен клик се активира диалог за въвеждане на име на класа. Така се създава нов клас и се присвоява на обекта, а ако трябва да се присвои съществуващ клас, се посочва и се избира от диалог **[Select a model element]** точния клас.



За да се създаде изходящ стимул от избрания обект към друг обект, се „кликва” два пъти върху изходния обект и в диалоговия прозорец се въвежда името на стимула след "->" низ ("<" низ за входящо и "<->" за изходящи с отиване и връщане. След натискане на [Enter] от избрания обект за целта се създава обект от изходящия стимул и се поставя последен.

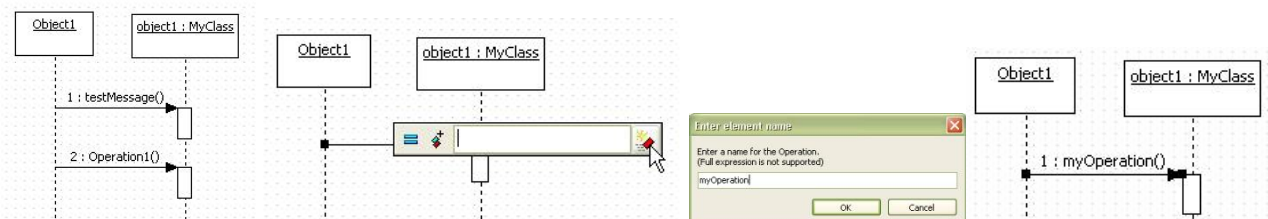


За да се създаде стимул, се избира от **[Toolbox] -> [Sequence] -> [Stimulus]**, след което се изтегля обекта в **[main window]** от единия до другия обект. Отваря се диалогов прозорец и се въвежда името му.

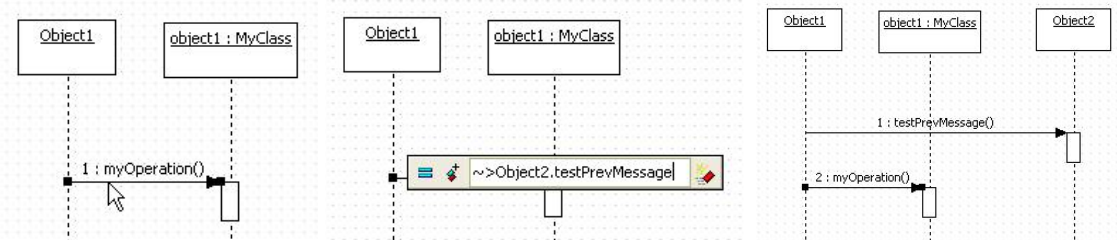


За да се използва операция от клас като стимул. Ако трябва да се присвои операция за стимулиране, се избира , след което от диалоговата кутия **[Select an operation]** се избира операцията с [OK].

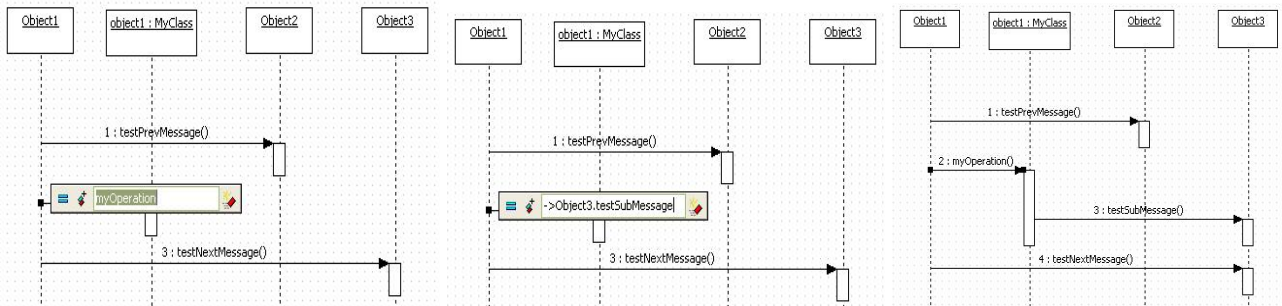
Може да се добави и втора операция. За да се създаде операция на клас като стимул, се избира за диалог и се въвежда име на операцията. Тя се появява и в **[model explorer]**.



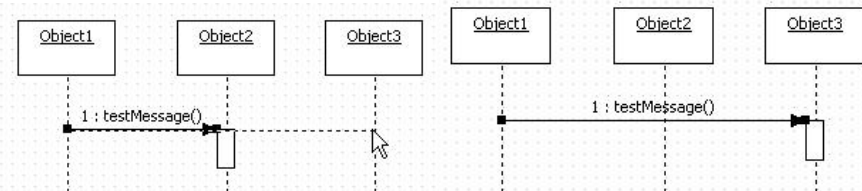
При създаване на текущия стимул от предишен стимул, се използва контекстното създаване - „кликва” се два пъти върху стимул, или се избира стимул и се натиска [Enter]. Така са създадени нов обект и стимули и са разположени над избрания стимул.



При създаване на подстимул се избира стимул и се натиска [Enter]. В диалоговата кутия се въвежда име на обекта и подстимула след "->" низ("<->" за входящи стимули). След [Enter] се създава новия обект и стимул и се разполагат отдолу на селектирания стимул.



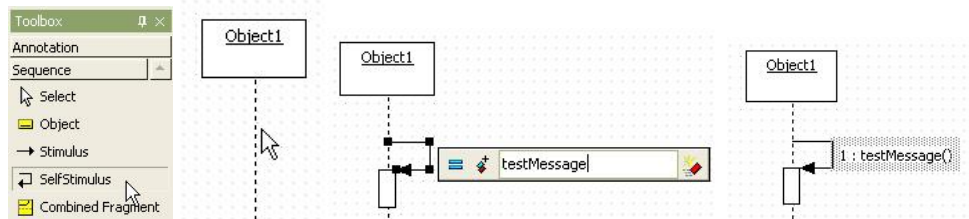
Ако трябва да се свърже стимула с друг обект, се кликва в края на стимула и се изтегля с Drag and drop до другия обект.



Промяна на [ActionKind] на стимула.

ActionKind	Shape
CALL	→
SEND	→
RETURN	--->
CREATE	<<create>> →
DESTROY	<<destroy>> →

За да се създаде **автостимул**: [Toolbox] -> [Sequence] -> [SelfStimulus]. Посочва се обекта или линията, излизаща от него в [main window]. В диалоговата кутия се записва името.



За да се създаде комбиниран фрагмент (**combined fragment**): [Toolbox] -> [Sequence] -> [Combined Fragment] и се посочва къде ще се разположи в [main window].

Toolbox

- Annotation
- Sequence
- SelfStimulus
- Combined Fragment
- Interaction Operand
- Frame

