UMLet

Free UML Tool for Fast UML Diagrams

Kurzanleitung

V. Eckardt 2017

Klassen- und Objektdiagramm

Die Benutzeroberfläche (Abbildung 1) ist in drei Bereiche eingeteilt: die Zeichenfläche für Diagramme, die Palette und das Eigenschaftsfenster.
Auf der Diagramm-Zeichenfläche positioniert der Anwender die benötigten UML-Elemente. Durch einen Doppelklick auf ein UML-Element in der Palette wird dieses auf die Zeichenfläche kopiert und kann anschließend mit der Maus an die gewünschte Position verschoben werden. (Hinweis: Element durch Anklicken markieren und dann am zentralen Ziehpunkt □ an die gewünschte Position ziehen.)
Im Eigenschaftsfenster kann der Anwender die Eigenschaften des ausgewählten UML-Elements ändern. Über eine Meta-Sprache (s. Tabelle 1 unten) kann bei den meisten UML-Elementen die Darstellung verändert werden. Die zu dem markierten UML-Element gehörenden Texte werden ebenfalls im Eigenschaftsfenster festgelegt.

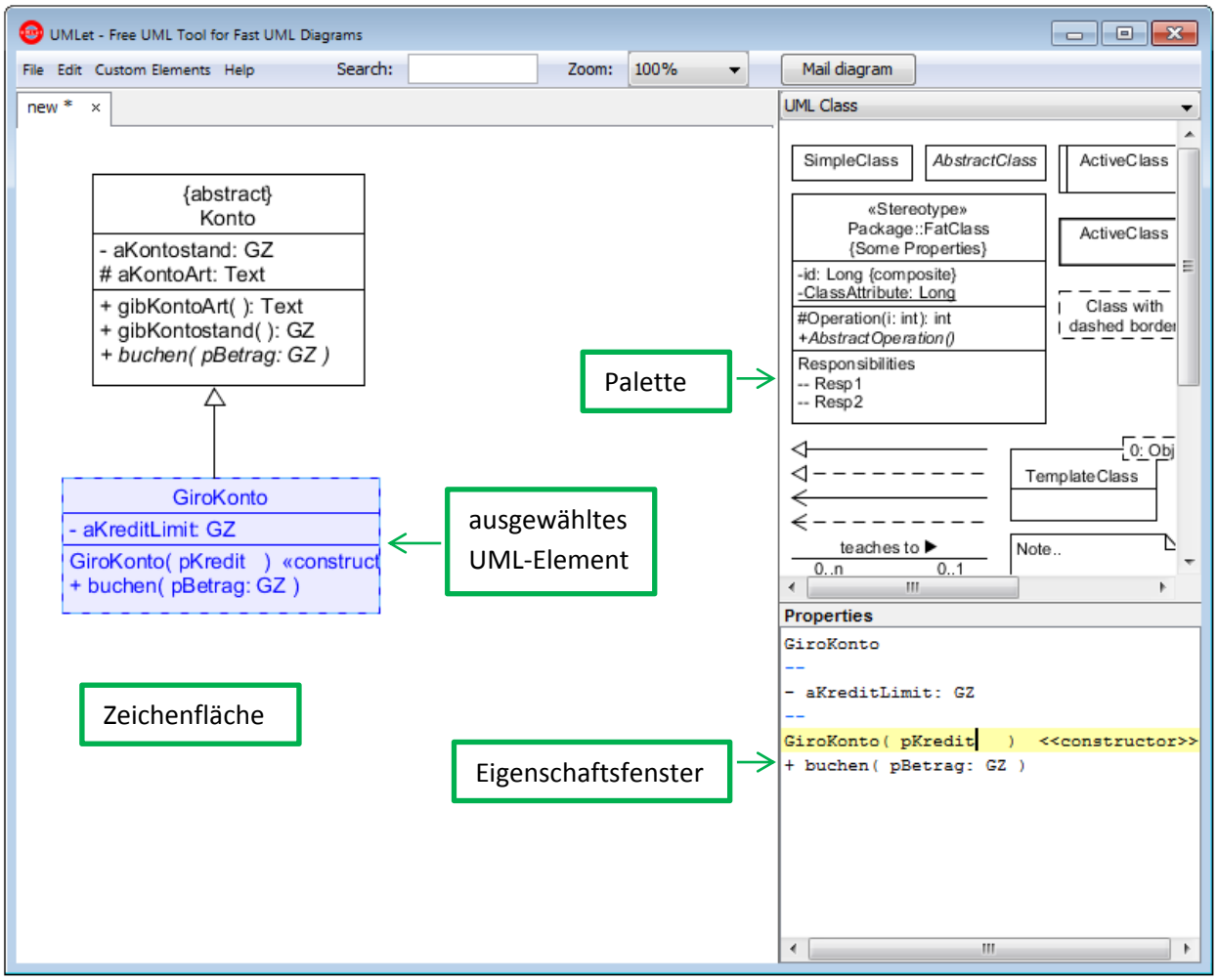


Abbildung 1: Benutzeroberfläche

allgemein anwendbare Kommandos

mehrere Elemente gleichzeitig markieren: *Strg + Mausklick* oder *Strg + Rechteck mit der Maus aufziehen*

das gesamte Diagramm verschieben: in die Mitte des Diagramms klicken und mit der Maus ziehen

gesamtes Diagramm kopieren: Strg + c

(Das Diagramm kann über die Zwischenablage anschließend z. B. in ein Word-Dokument eingefügt werden. Diagramme können aber auch über das Menü: *File | Export as* im Format BMP, JPG, SVG u. a. gespeichert werden.)

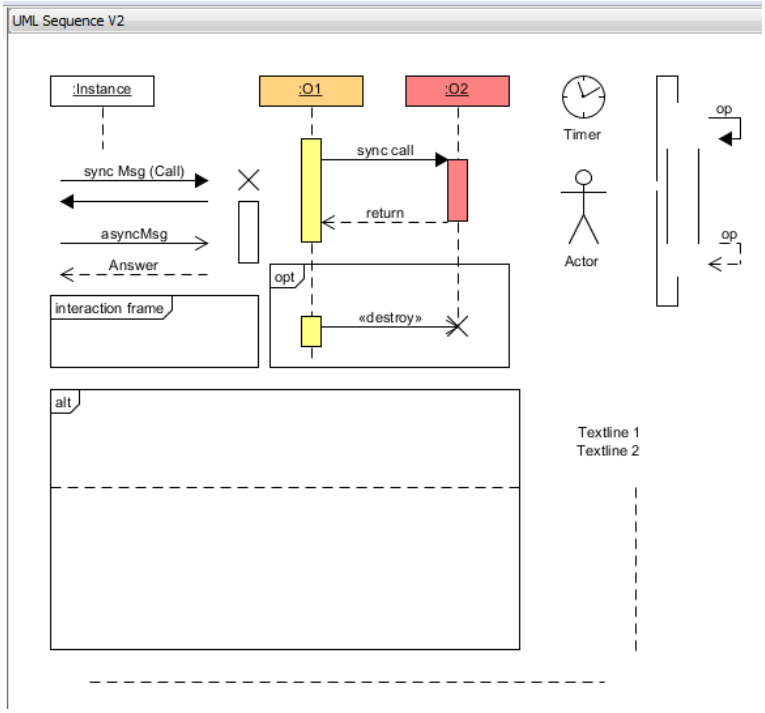
Größe der Darstellung ändern: in die Zeichenfläche klicken, mit + (vergrößern), - (verkleinern)

	innerhalb von Klassen- oder Objektsymbolen
--	durchgezogene, waagrechte Linie
-.	*gestrichelte, waagrechte Linie*
-..	*punktierte , waagrechte Linie*
Text	Text in Fettschrift
/Text/	Text kursiv (abstrakte Operation)
Text	Text unterstrichen (Klassenattribut oder Klassenoperation)
	außerhalb von Klassensymbolen
lt=<<-	Vererbungspfeilspitze
lt=..	punktierte Linie (z. B. zum „Anbinden“ eines Notizsymbols)
	Assoziationen
lt=<->	Linie mit Pfeilspitzen an beiden Enden (bidirektionale Assoziation)
lt=-	Linie ohne Pfeilspitzen
lt=->	Linie mit Pfeilspitz an einem Ende (unidirektionale Assoziation)
r1=Rollenname 1	Rollenname (oberes Ende bzw. linke Seite)
r2=Rollenname 2	Rollenname (unteres Ende bzw. rechte Seite)
m1=Wert	Kardinalität/Multiplizität zu r1
m2=Wert	Kardinalität/Multiplizität zu r2
m2pos=x,y	Position des Texts von m2 um x,y Pixel verschieben (m1pos=x,y, r1pos=x,y und r2pos=x,y arbeiten entsprechend)
lt=<<<<-	Aggregation (ohne Pfeilspitze)
lt=<<<<<-	Komposition (ohne Pfeilspitze)
lt=<<<<<->	Komposition (mit Pfeilspitze am r2-Ende)

Tabelle 1: Meta-Sprache für Klassen- und Objektdiagramme
(teilweise auch in anderen Diagrammtypen verwendbar)

Sequenzdiagramme

Zum Erstellen von Sequenzdiagrammen bietet UMLet mehrere (unterschiedlich komplexe) Varianten an. Mit Hilfe der Palette „UML Sequenz“ kann man ähnlich wie bei einem Klassendiagramm ein Sequenzdiagramm aus einzelnen Grafikelementen zusammen setzen.
Zur Darstellung von Selbstdelegation und anderen speziellen Elementen muss die Palette jedoch erweitert werden. Mithilfe von (selbst erstellten) Textelementen können Texte frei positioniert werden.



Beispiel für eine um zusätzlich benötigte Elemente erweiterte Palette.

Sequenzdiagramm-All-in-on-Palette

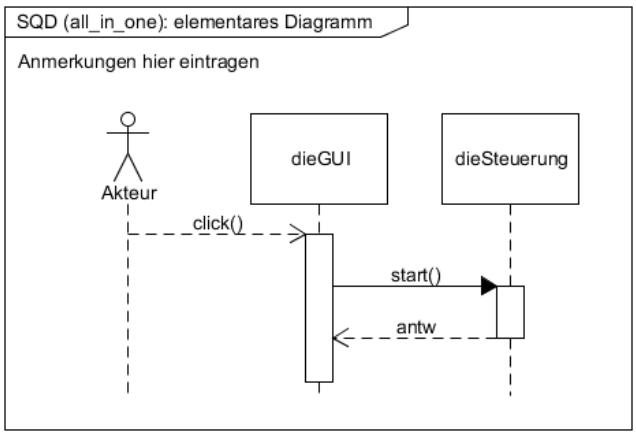
Mit der Sequenzdiagramm-All-in-on-Palette lassen sich Sequenzdiagramme erheblich schneller erstellen. Man muss allerdings in Kauf nehmen, dass man nur geringen Einfluss auf die Platzierung der einzelnen Elemente und Texte hat. Die UML-Elemente (Diagramme) der Palette „UML Sequence (All in one)“ besitzen eine eigene Meta-Sprache, die im Folgenden beschrieben wird.

Prinzip: Mit *obj= Objektname ~ a* bzw. *obj= Objektname ~ b* usw. werden Lebenslinien erzeugt.
Mit *b->>>c: Botschaft( ) ; on=c* wird eine Linie mit ausgefüllter Pfeilspitze von *b* nach *c* gezeichnet und ein Fokusbalken auf *c* eingetragen (synchrone Botschaft).
Mit *c.>b; off=c* wird eine gestrichelte Linie mit offener Pfeilspitze von *c* nach *b* gezeichnet und der Fokusbalken auf *c* beendet.

Zum Erstellen eines eigenen Diagramms setzt man am besten ein möglichst einfaches Diagramm aus einer All-in-one-Sequenzdiagramm-Palette auf die Zeichenfläche und passt dieses im Eigenschaftsfenster anschließend an die eigenen Anforderungen an.

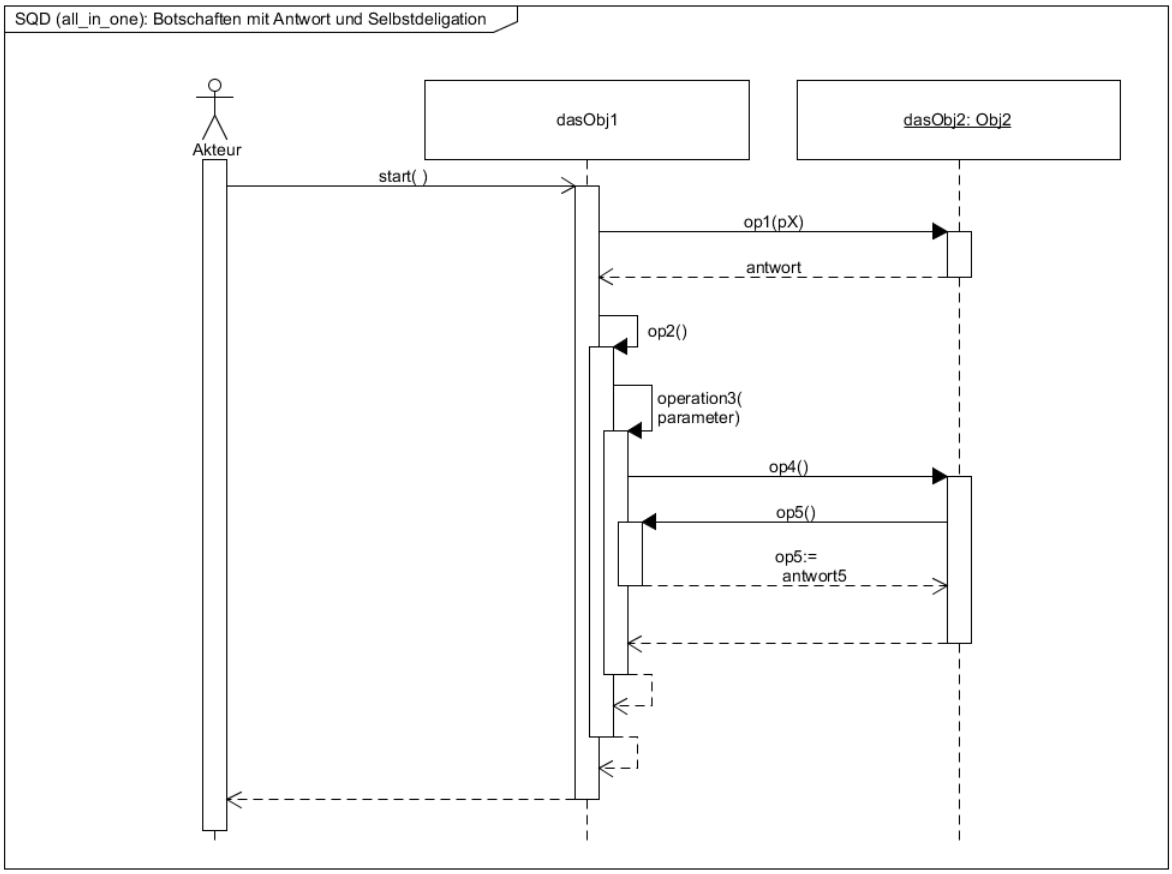
Beispiele für Sequenzdiagramme (aus der Palette „SQD_all_in_one_TG-IT_UML-2“):

Sequenzdiagramm 1

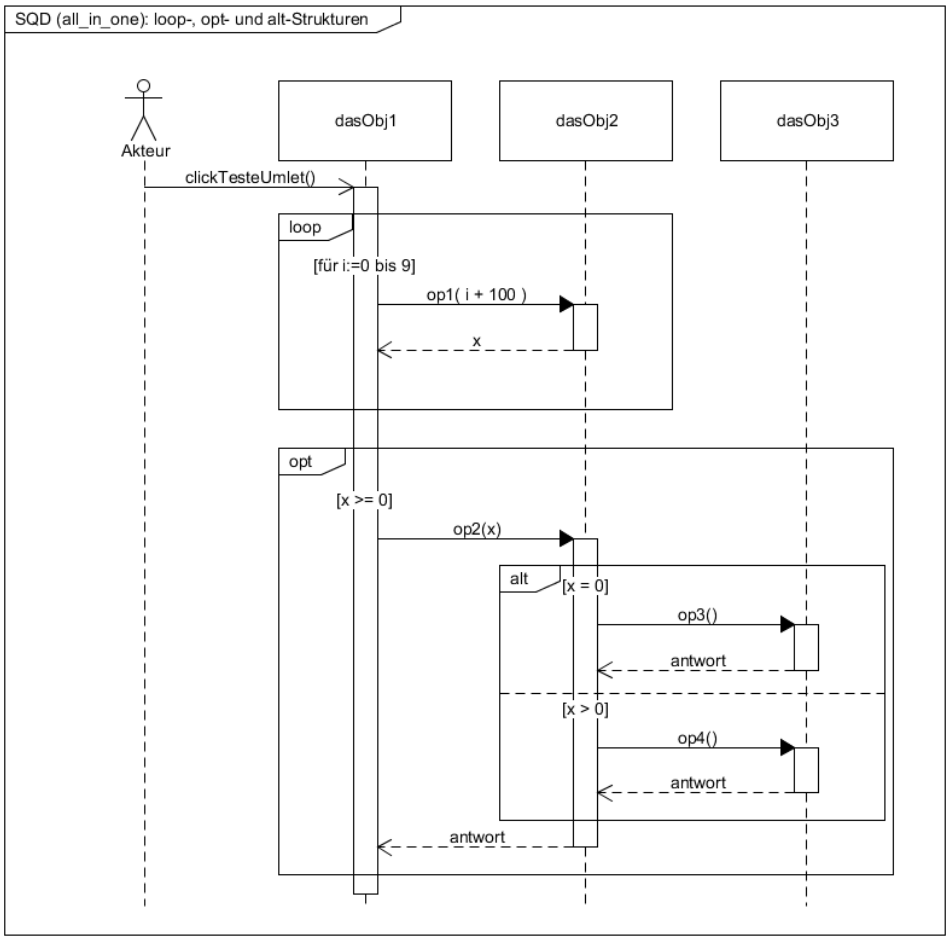


Eigenschaftsfenster

```
title= SQD (all_in_one): elementares Diagramm
desc=Anmerkungen hier eintragen
obj= Akteur ~ a ACTOR
obj= dieGUI ~ b
obj= dieSteuerung ~ c
// Botschaften senden
a.>b : click(); on=b
b->>>c: start(); on=c
c.>b : antw; off=c
```

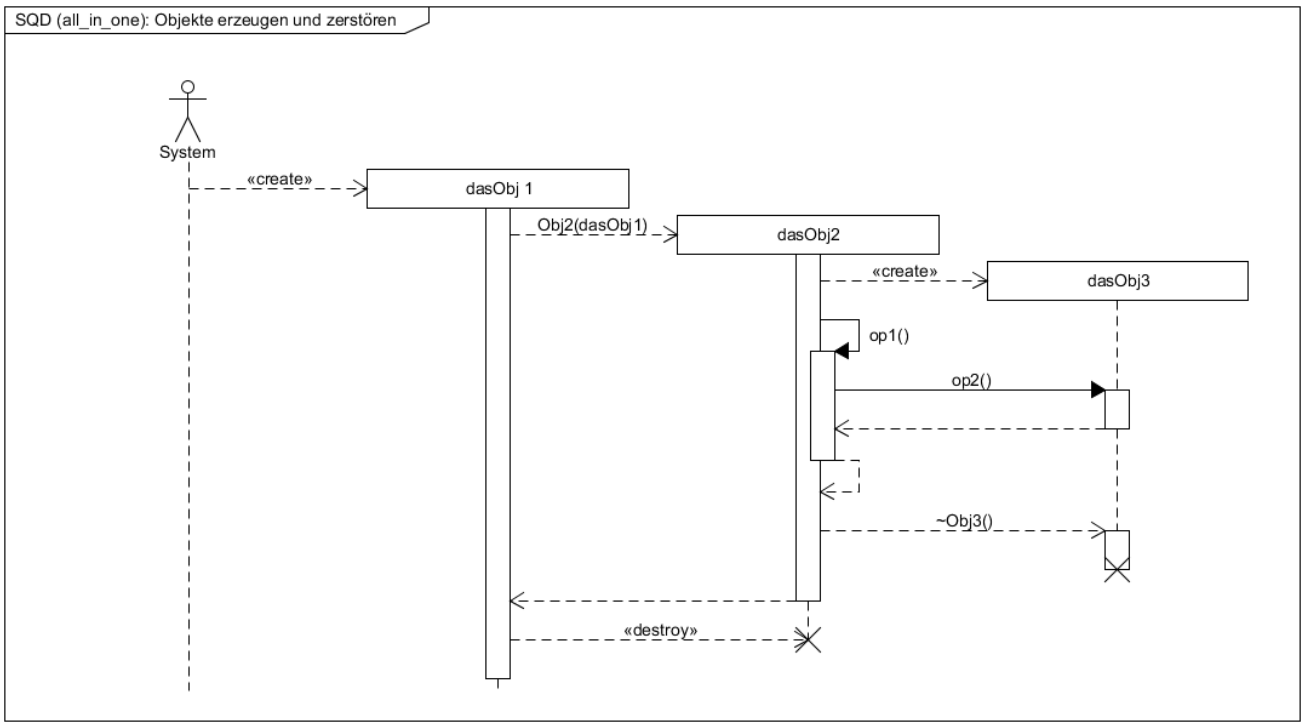



```
title= SQD (all_in_one): Botschaften mit Antwort und Selbstdeligation
obj= Akteur ~ a ACTOR EXECUTION  // zur Demo: Akteuer hier mit Fokusbalken
obj= dasObj1 ~ b
obj=_dasObj2: Obj2_~ c    // zur Demo: Objektname hier unterstrichen

// Botschaften senden
a->b : start( ); on=b;          // offene Pfeilspitze ; Fokusbalken

b->>>c : op1(pX); on=c;     // ausgefüllte Pfeilspitze; Fokusbalken beginnt
c.>b : antwort; off=c                    // gestrichelte Linie; Fokusbalken endet

// Selbstdeligation, verschachtelt
 b->>>b+ : op2()
 on= b
   //                          v  Leerschritt bewirkt hier einen Umbruch
   b->>>b+ :operation3( parameter)
   on= b
   // Delegation und Rückdelegation
   b->>>c : op4(); on=c
   c->>>b : op5(); on=b
   b.>c : op5:=\n        antwort5; off=b
   //              ^  Umbruch durch Steuerzeichen erzwingen
   c.>b; off=c

   // Selbstdeligation beenden
   b.>b+; off=b
 tick=1       // Abstand um eine Einheit erzeugen
 b.>b+; off=b
tick=1

b.>a; off=b
```




```
title= SQD (all_in_one): loop-, opt- und alt-Strukturen
obj= Akteur ~ a ACTOR
obj= dasObj1 ~ b
obj= dasObj2 ~ c
obj= dasObj3 ~ d

// Botschaften senden
a->b : clickTesteUmlet(); on=b;       // offene Pfeilspitze ; Fokusbalken

// loop-Struktur über Lebenslinie b bis c; Text über b setzen
combinedFragment= loop ~ loop1  b c;
b:[für i:=0 bis 9]
b->>>c : op1( i + 100 ); on=c;
c.>b : x; off=c
-- = loop1       // Ende der loop-Struktur ( =loop1 ist optional)
tick=

//opt- und alt-Struktur verschachtelt
combinedFragment= opt ~ opt1  b d;
b:[x >= 0]
b->>>c : op2(x); on=c

combinedFragment= alt ~ alt1  c d; c:[x = 0]
c->>>d : op3(); on=d
d.>c : antwort; off=d
..=alt1; c:[x > 0]              // gestrichelte Trennlinie und Text unter der Linie einfügen
c->>>d : op4(); on=d
d.>c : antwort; off=d ; -- = alt1;

c.>b : antwort; off=c ; -- = opt1
```

Sequenzdiagramm 4



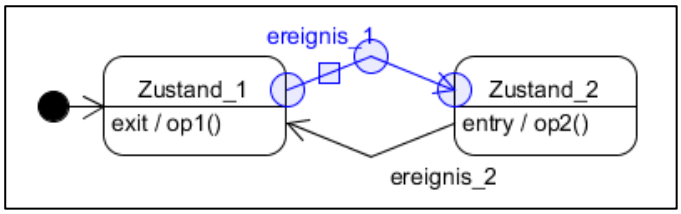
Beschreibung des Sequenzdiagramms 4 im Eigenschaftsfenster in der UMLet-Meta-Sprache

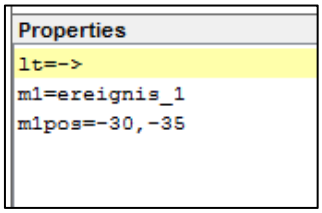
```
title= SQD (all_in_one): Objekte erzeugen und zerstören
obj=System~a ACTOR
obj= dasObj 1~ b CREATED_LATER EXECUTION
obj= dasObj2 ~ c CREATED_LATER EXECUTION          // Lebenslinie mit Fokusbalken
obj= dasObj3 ~ d CREATED_LATER                    //              ohne Fokusbalken erzeugen

// Objekte erzeugen
a.>b : <<create>>            // Die 1. Botschaft an die Lebenslinie „erzeugt“ das Objekt
b.>c:Obj2(dasObj1)
c.>d : <<create>>

//Selbstdeligation
  c->>>c+ : op1()
  on= c
  c->>>d : op2(); on=d
  d.>c; off=d
  c.>c+; off=c

tick=1                       //  Abstand einfügen

// Objekte zerstören
c.>d : ~Obj3(); on=d                  // mit Fokusbalken
off=d;  destroy=d
c.>b; off=c
b.>c : <<destroy>>; destroy=c        // ohne Fokusbalken
```

Zustandsdiagramm

Die Elemente eines Zustandsdiagramms können mithilfe der Palette „UML State Machine“ mit der Notation, wie sie bei Klassendiagramme verwendet wird, im Eigenschaft-Fenster bearbeitet werden.

Die Linien von Transitionen können durch „ziehen“ geformt werden. Die Position des Tests (Ereignis) kann bei Bedarf wie bei Assoziationen im Klassendiagramm gesteuert werden (s. Beispiel unten).

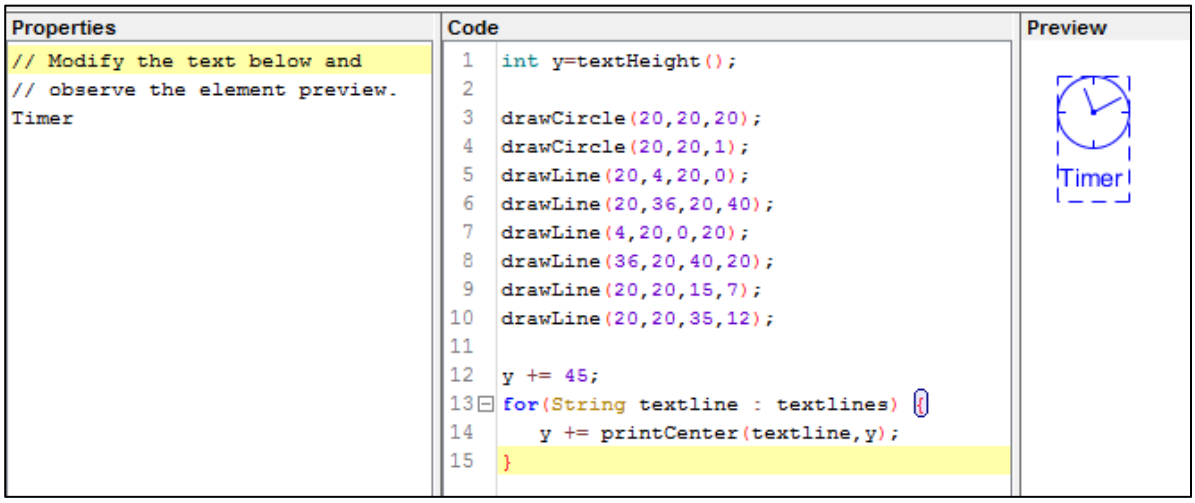




Neue Elemente erstellen

Über das Menü *Custom Elements | New* können neue UML-Elemente in Java programmiert und zu einer Palette hinzugefügt werden.

Beispiel Timer-Symbol



weitere Beschreibungen siehe:

http://www.umlet.com/ce/ce.htm (custom element edit)

UML-Elemente in der Palette verändern

Über das Menü *File | Edit Current Palette* kann die Palette, die gerade angezeigt wird, bearbeitet werden.

Falls die vorhandenen Paletten nicht verändert werden sollen, besteht die Möglichkeit, mithilfe des Explorers eine Palette zu kopieren: Im Unterverzeichnis *palettes* ist für jede Palette eine Datei mit der Dateierweiterung *.uxf* vorhanden. Wenn eine dieser Dateien kopiert wird, steht nach dem Neustart von UMLet eine weitere Palette zur Verfügung.

Die UML-Elemente in der Palette können im Modus *File | Edit Current Palette* kopiert, verschoben und wie oben beschrieben, verändert werden.

weitere Hilfen siehe: http://www.umlet.com/faq.htm

Download: http://www.umlet.com (aktuelle Version, Stand 6.07.2017: UMLet 14.2)