

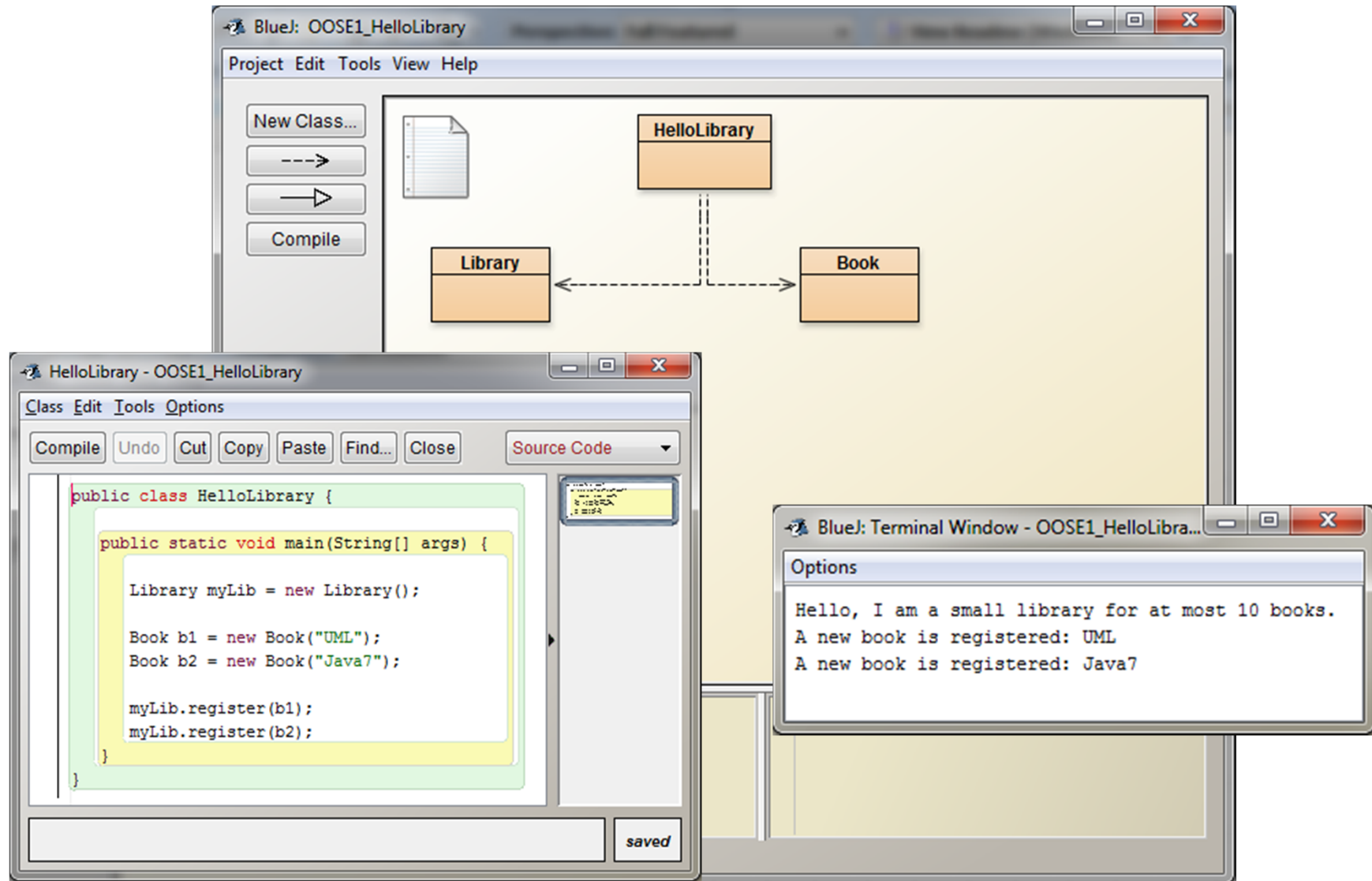
OOSE 1

Java mit BlueJ und UML-by-Example (Fakultative Hörsaalübung)

SS 2016

Birgit Demuth

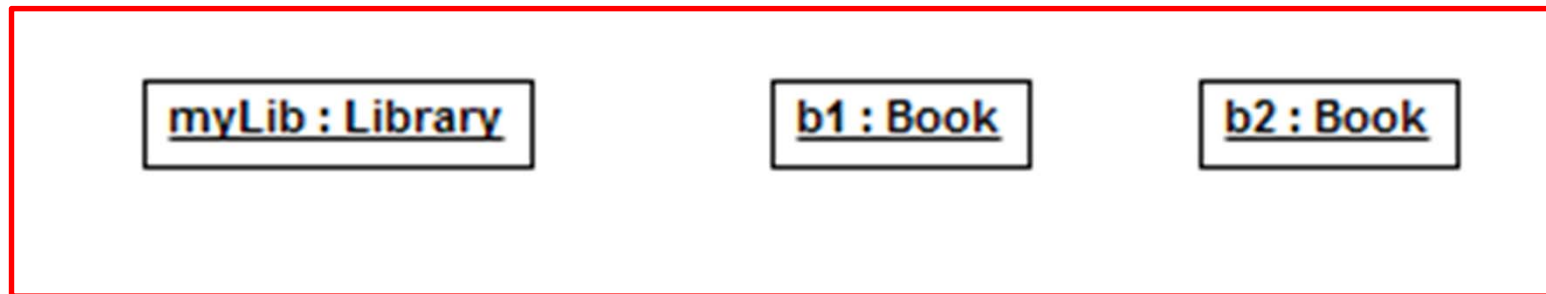
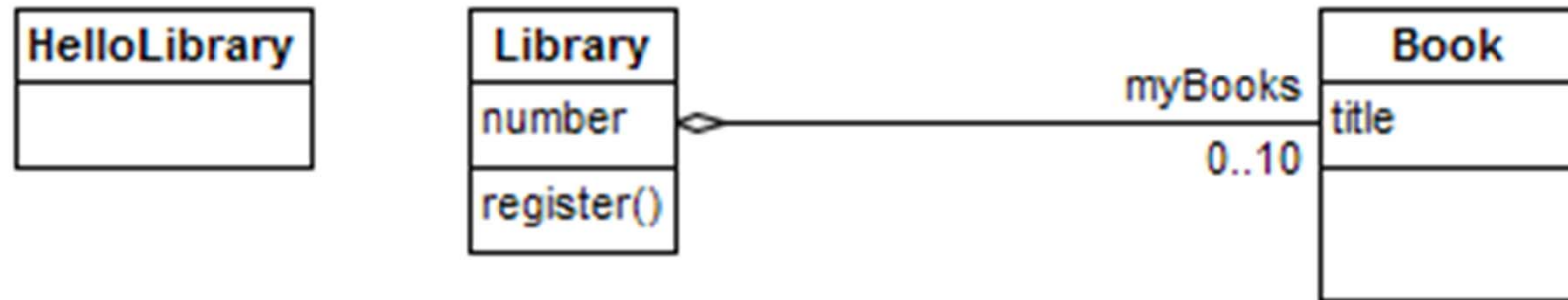
Arbeit mit BlueJ (version 3.1.7) – Demo (1)



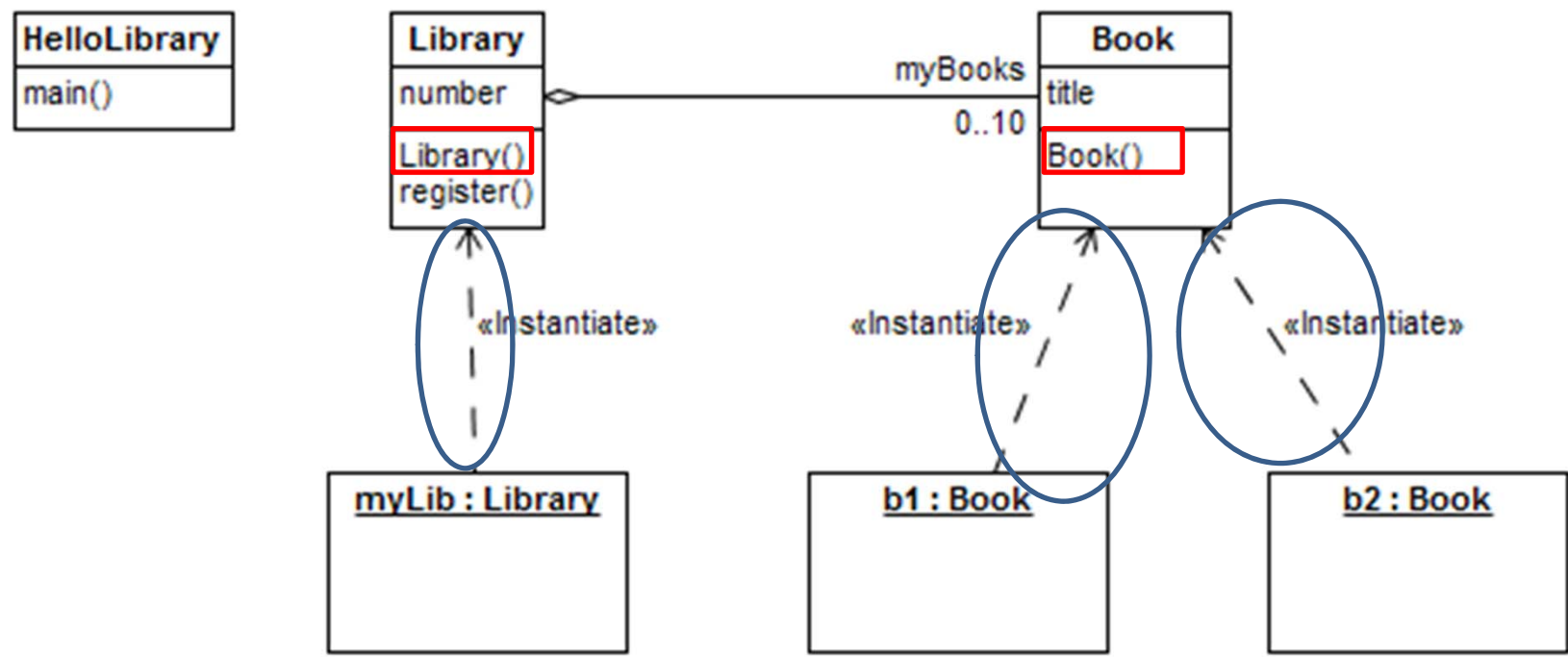
Unser erstes UML-Analyseklassendiagramm (HelloLibrary Anwendung)



Klassen und Objekte

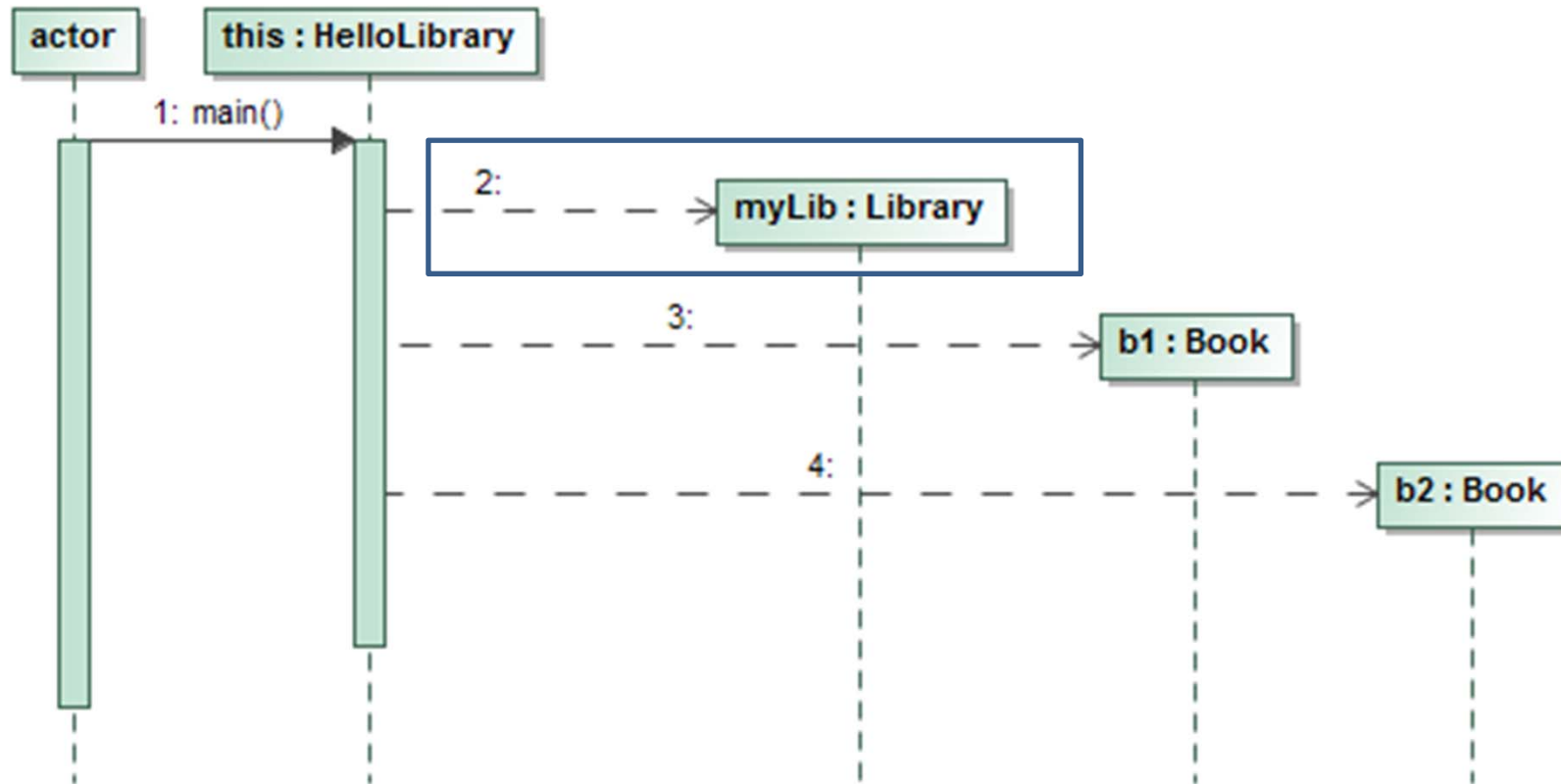


Konstruktor und Instanziierung von Objekten



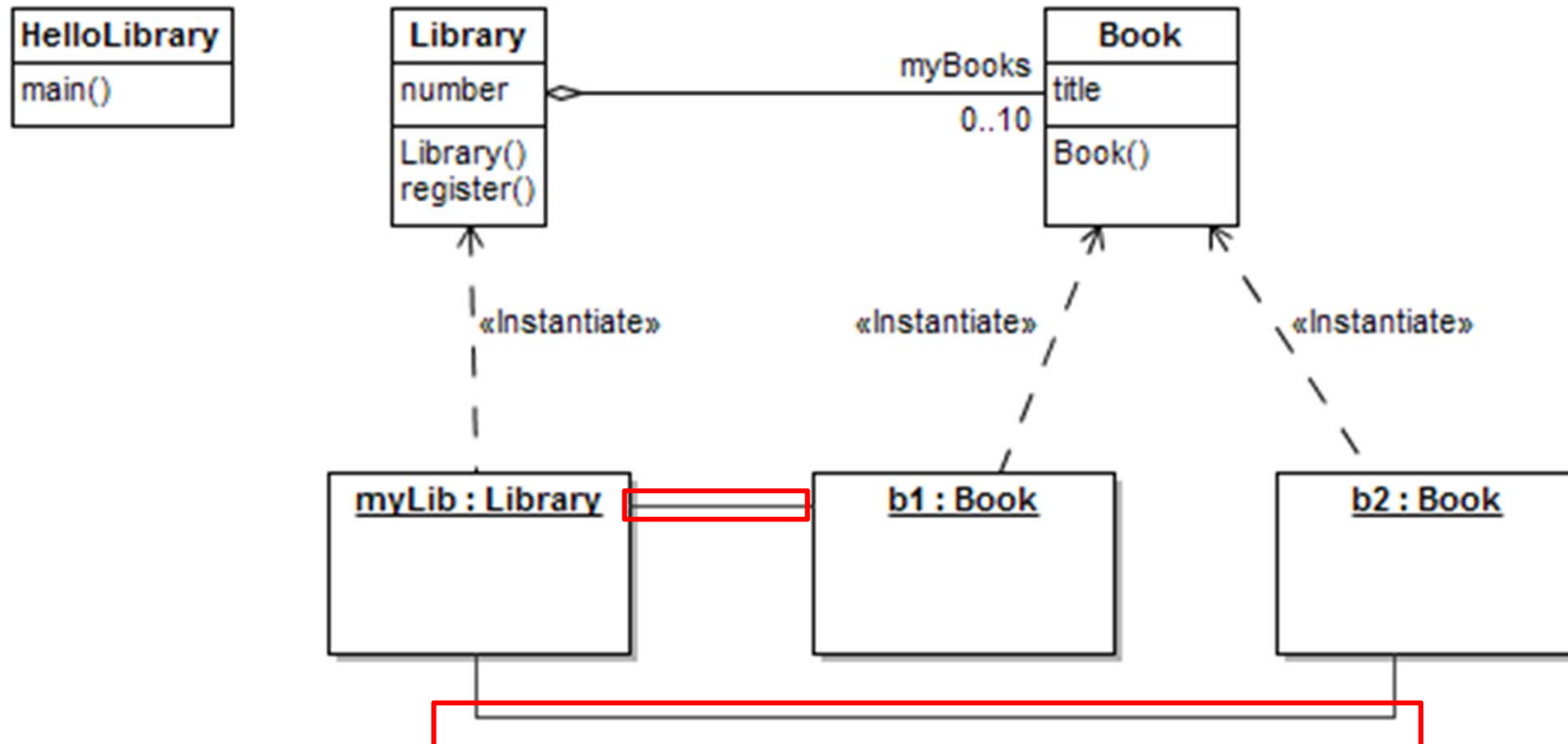
Instanziierung von Objekten

Darstellung im Sequenzdiagramm

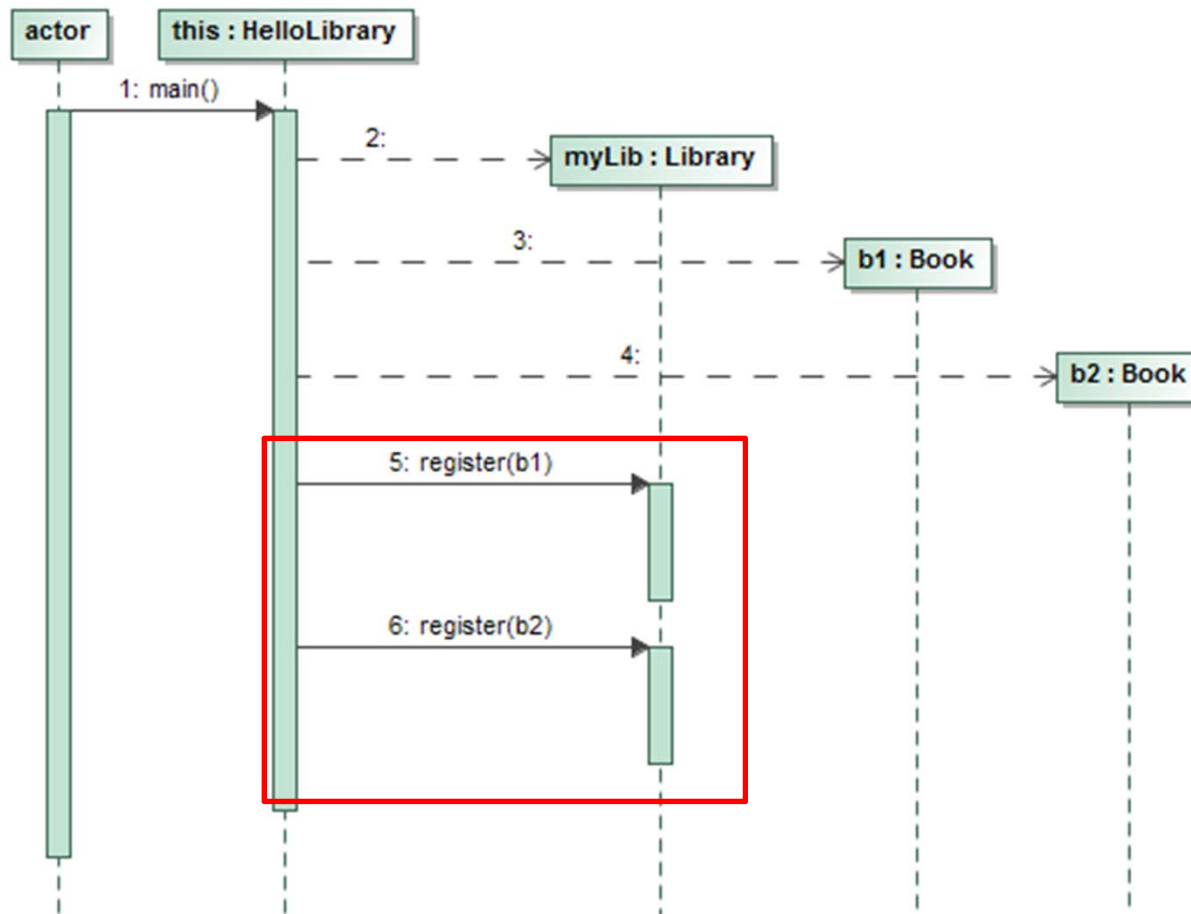


Registrierung von Objekten Darstellung im Objektdiagramm

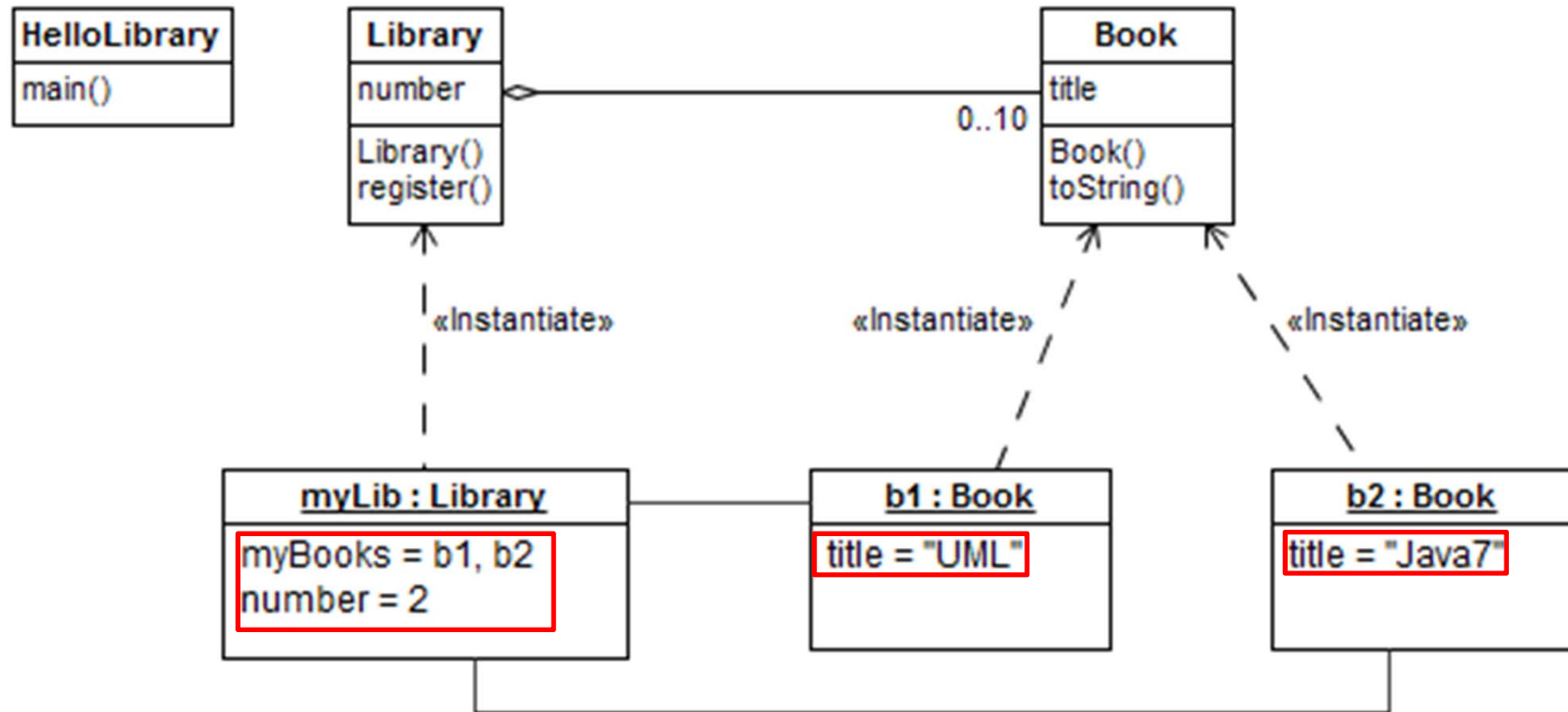
(Erzeugung von **Links** durch Aufruf der Methode `register()` ())



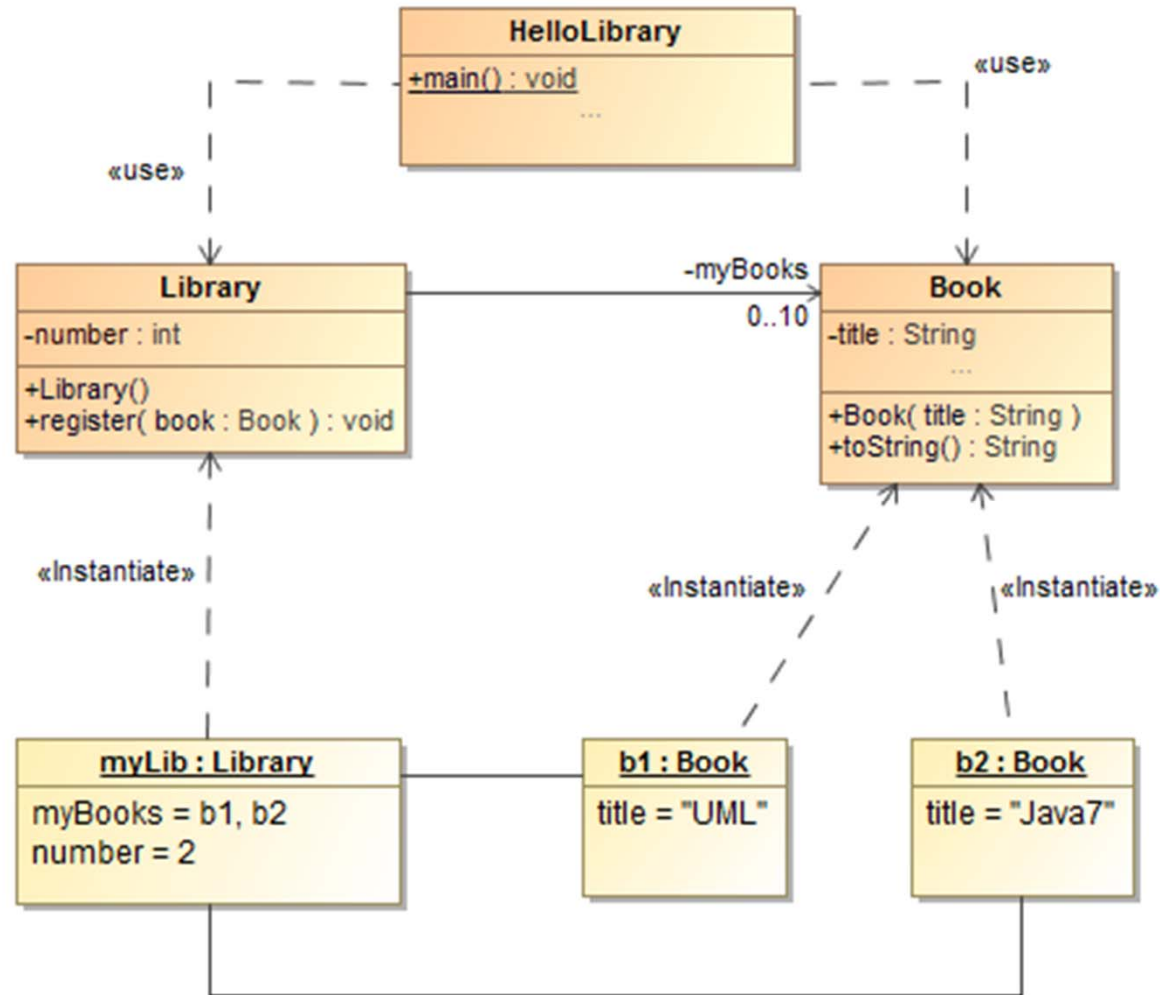
Registrierung von Objekten Darstellung im Sequenzdiagramm



Objektdiagramm mit Slots (Attribut = Wert)

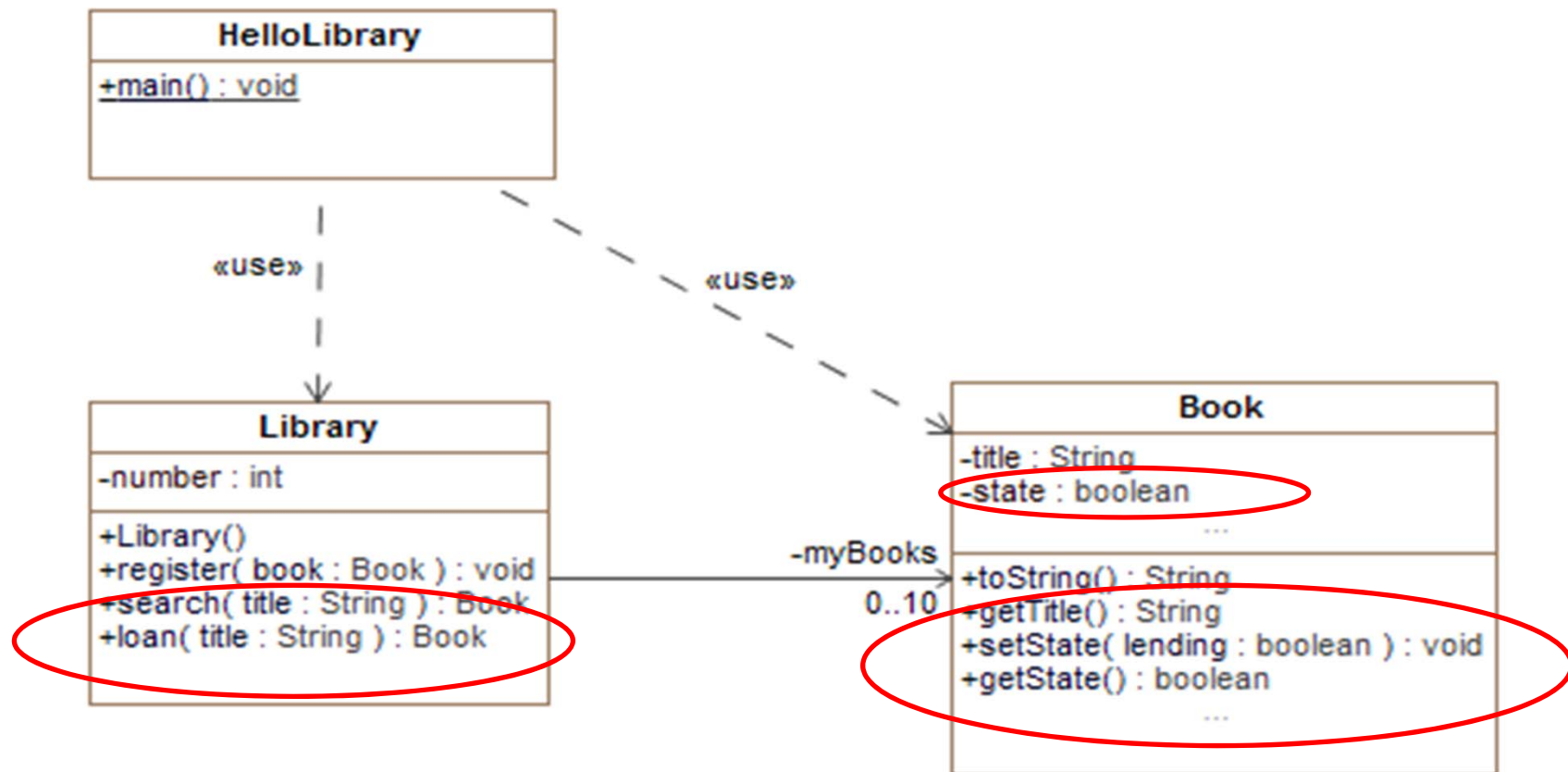


Entwurfsklassendiagramm mit instanziierten Objekten

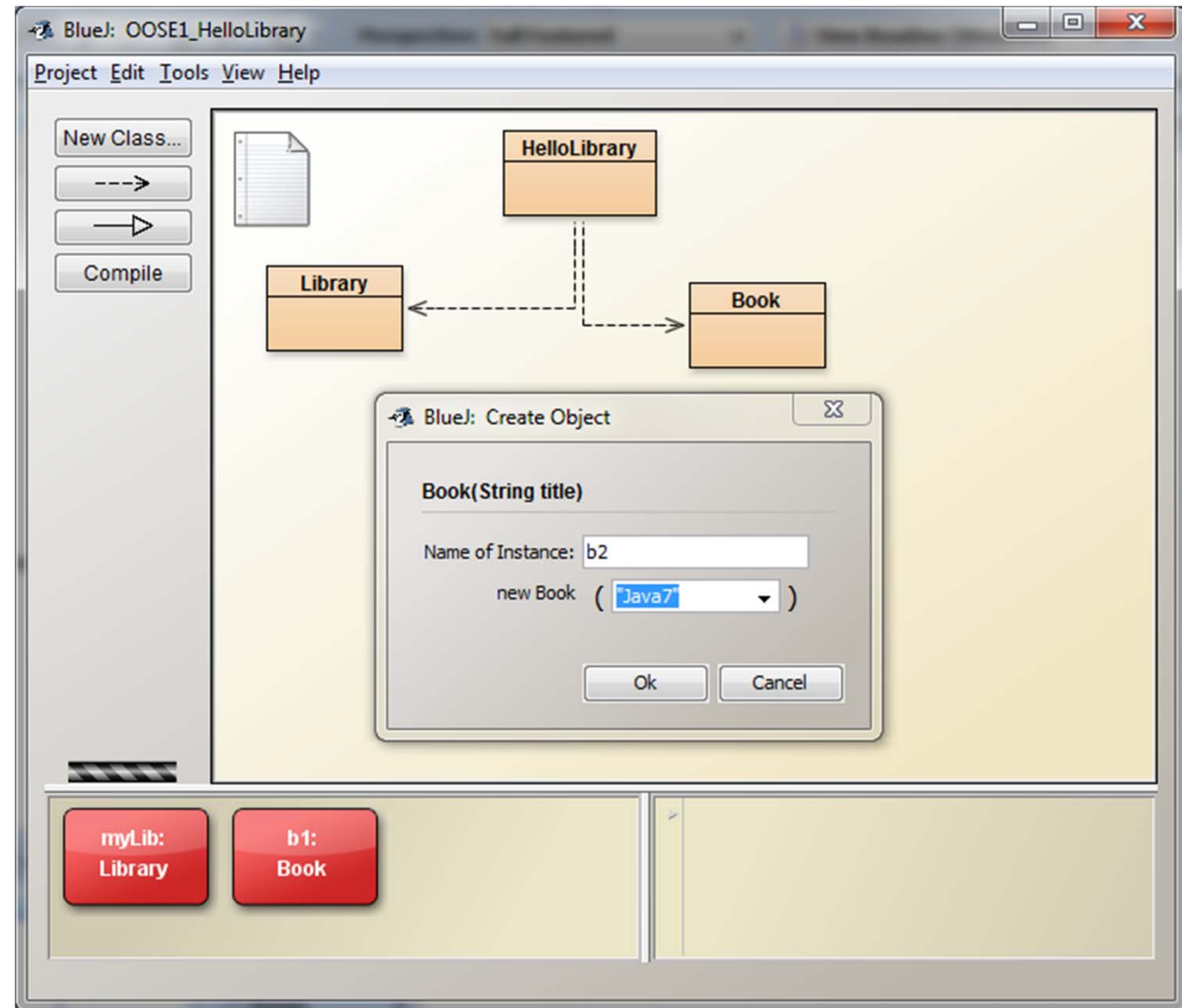


Entwurfsklassendiagramm

HelloLibrary extended (Vorschau Übung U02)

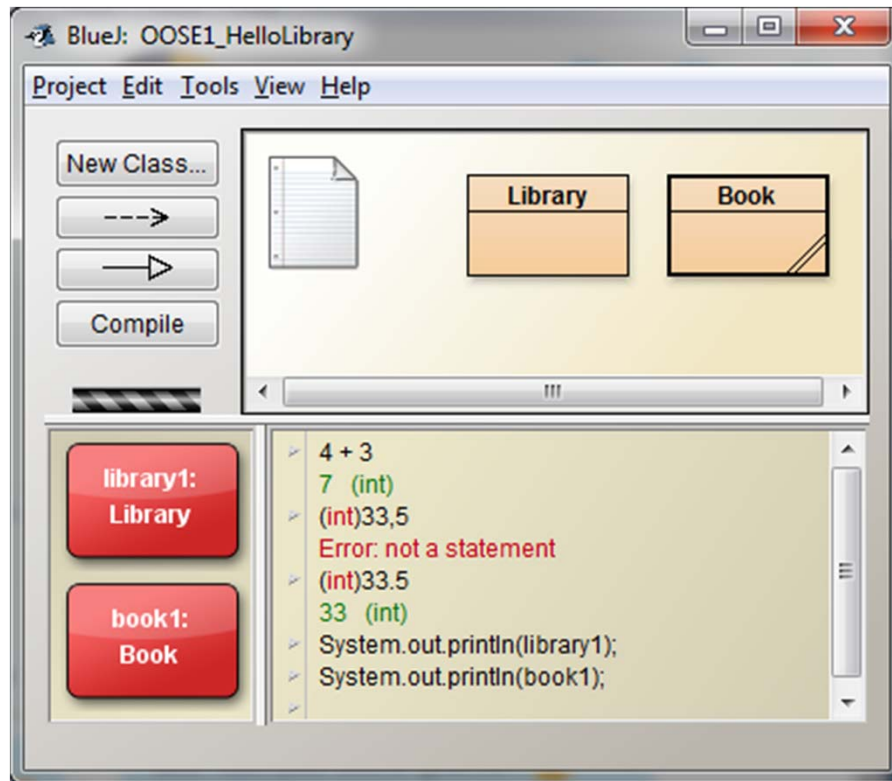


Arbeit mit BlueJ (version 3.1.7) – Demo (Object Bench)



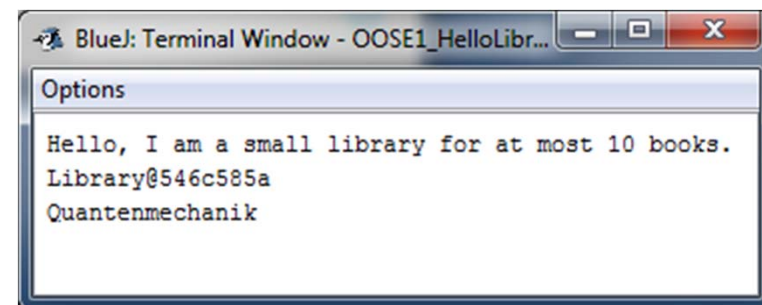
Object Bench:
„Spielen“ mit Objekten

Arbeit mit BlueJ (version 3.1.5) – Demo (Code Pad)



Code Pad:

- Berechnung von Java-Ausdrücken
- Ausführen von Java-Statements
- Eingabe von mehrzeiligen Anweisungen (durch Drücken von Shift Enter) und Ausführen (Enter)



Arbeit mit BlueJ (version 3.1.7) – Demo (Debugger)

The image illustrates the workflow in BlueJ for debugging a method call. It consists of four main windows and several annotations:

- BlueJ: debugdemo**: The main IDE window. A red arrow labeled '1' points to the 'demo1: Demo' button in the bottom left, with the text 'Setzen eines Breakpoints' (Setting a breakpoint) below it. A red arrow labeled '2' points from the 'Demo' class in the class browser to the 'Method Call' dialog.
- BlueJ: Method Call**: A dialog box titled 'BlueJ: Method Call' showing the method signature `int loop(int count)` and the argument `demo1.loop (4)`. It has 'Ok' and 'Cancel' buttons.
- Demo - debugdemo**: The source code editor showing the `loop` method. A red dot (breakpoint) is set on the line `for (int i=0; i<count; i++) {`. The code includes:

```
private String name;
private int answer;

/**
 * Constructor for objects of class Demo
 */
public Demo()
{
    name = "Marvin";
    answer = 42;
}

/**
 * Loop for a while and do some meaningless computations.
 */
public int loop(int count)
{
    int sum = 17;
    for (int i=0; i<count; i++) {
        sum = sum + i;
        sum = sum - 2;
    }
    return sum;
}

/**
 * Method for demonstrating single stepping with nested method call.
 */
public int carTest()
{
    int places;
    Car myCar = new Car(2, 3);
```
- BlueJ: Debugger**: The debugger window. A red arrow labeled '3' points from the 'Method Call' dialog to this window. The 'Options' tab is active, showing 'main (at breakpoint)' under 'Threads'. The 'Call Sequence' tab shows 'Demo.loop' selected. The 'Local variables' section displays:

```
int count = 4
int sum = 17
int i = 0
```

The debugger also features control buttons: Halt, Step, Step Into, Continue, and Terminate.

Schüttelrätsel mit Begriffen zur Objektorientierung (1)

	A	B	C	D	E	F	G	H	I	J	K	L
1	J	S	S	N	D	T	L	I	V	E	Ä	Ä
2	K	A	M	O	N	A	K	Z	I	R	D	U
3	T	D	E	I	T	R	A	A	A	F	I	G
4	O	N	A	T	T	I	S	T	R	R	T	N
5	E	U	G	A	Ä	H	E	N	I	E	E	U
6	B	Z	S	R	I	C	S	N	T	E	K	A
7		T	E	E	E	C		S	E	Z	I	S
8				P	T	H			I	N	U	P
9				O	I	N			N		Z	R
10									A		S	G
11											T	
12											N	
13											G	

	K9
	C2
	F4
	A4
	H1
	C1
	B6
	E1
	G4
	D2
	I5
	K8
	J7
	B1
	I6
	L4

Lösungswort an

bd1tud@gmail.com

Dem Teilnehmer am Rätsellösen mit dem ersten richtigen Lösungswort
Wird ein Bonuspunkt für die Klausur SS 2016 gutgeschrieben.

Ich höre und ich vergesse.
Ich sehe und ich erinnere mich.
Ich handle und ich verstehe.

Xun Zi (chinesischer Philosoph)
Vor mehr als 2000 Jahren