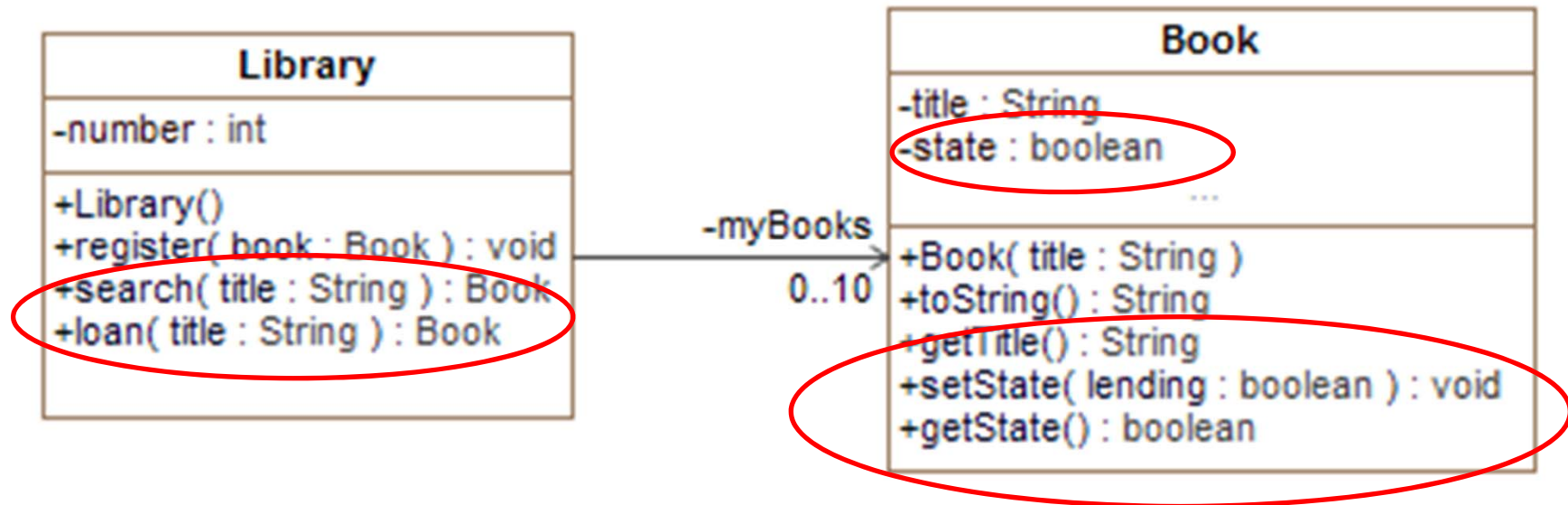


OOSE2

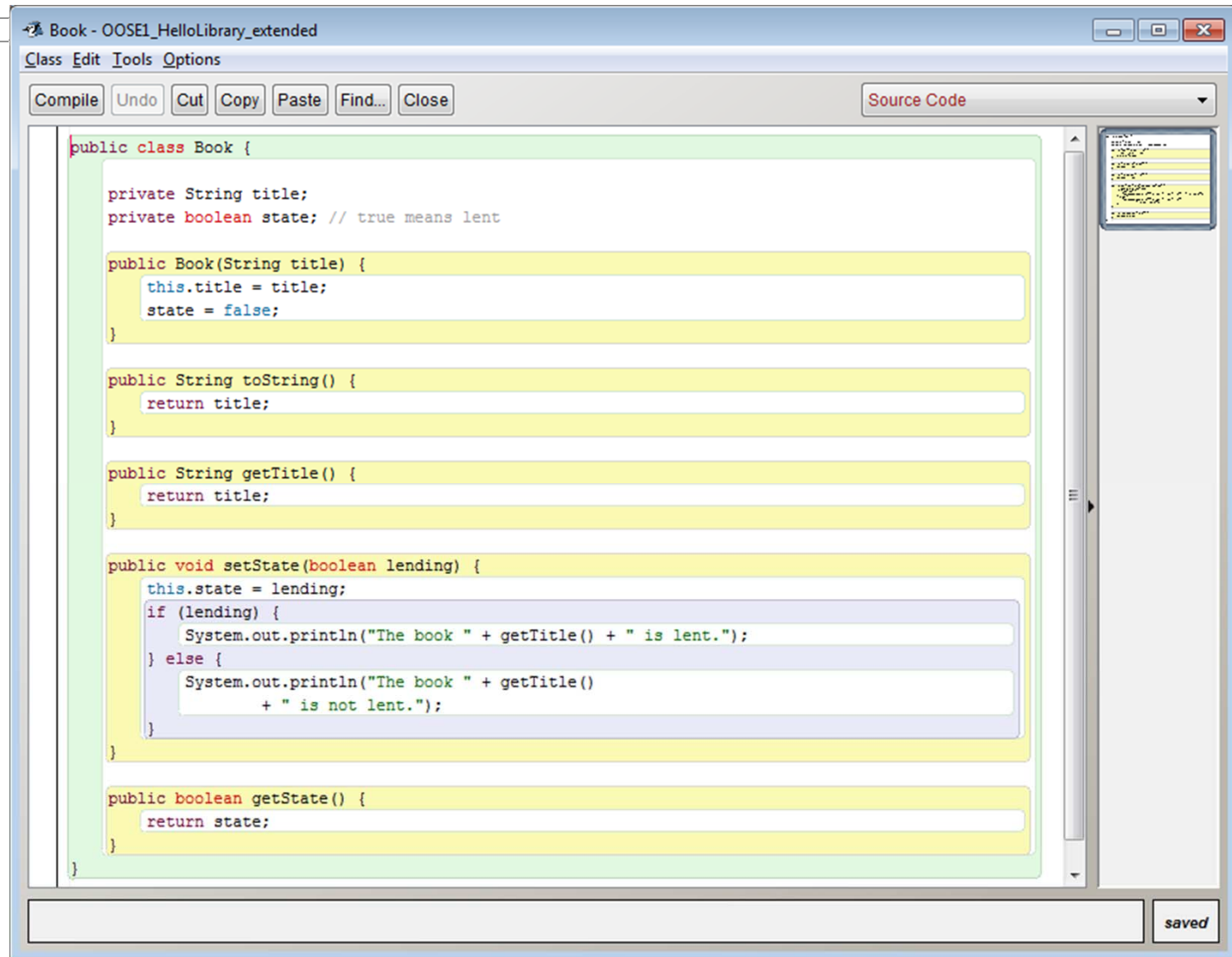
Vererbung und Polymorphie mit BlueJ

Lehrstuhl Softwaretechnologie, Dr. Birgit Demuth
Sommersemester 2016

Erweiterung von HelloLibrary (U02)

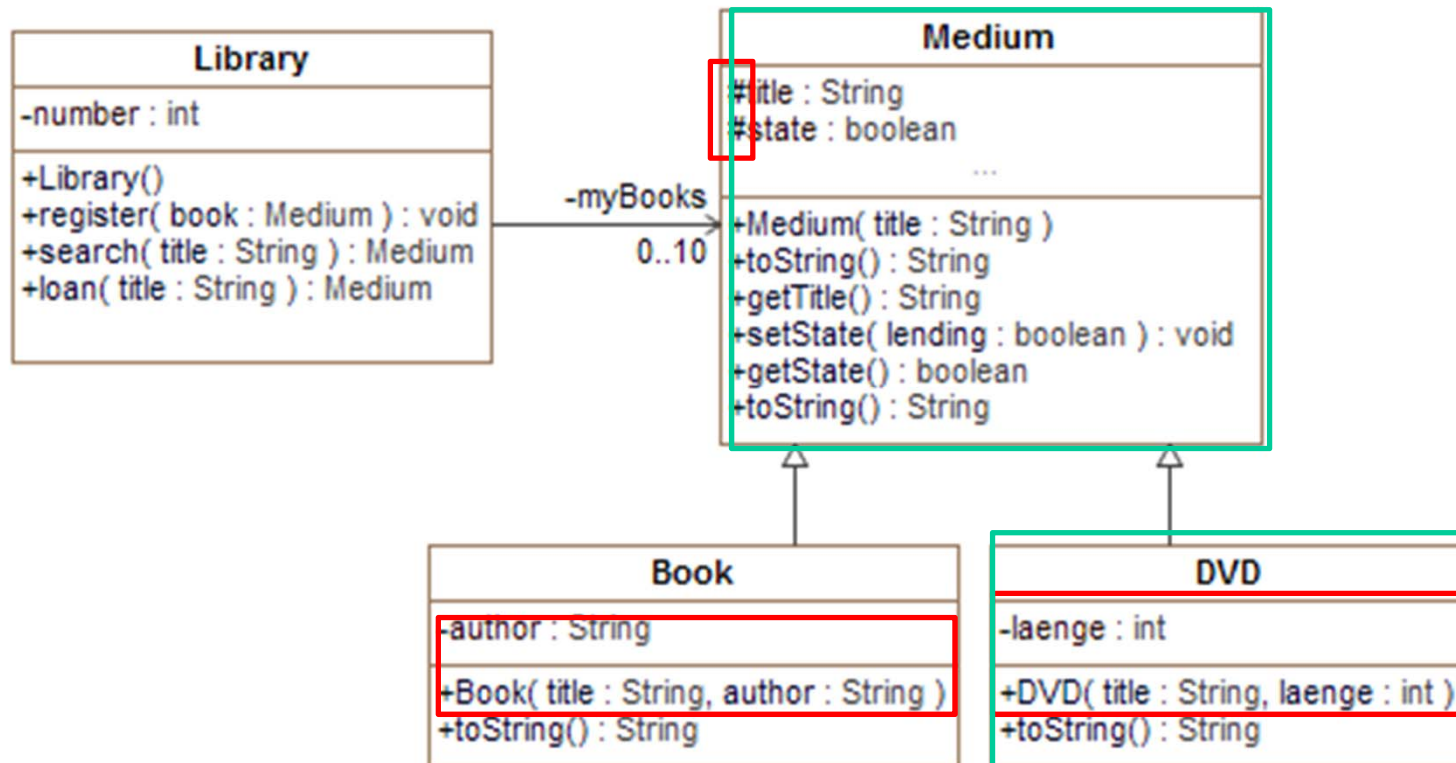


BlueJ

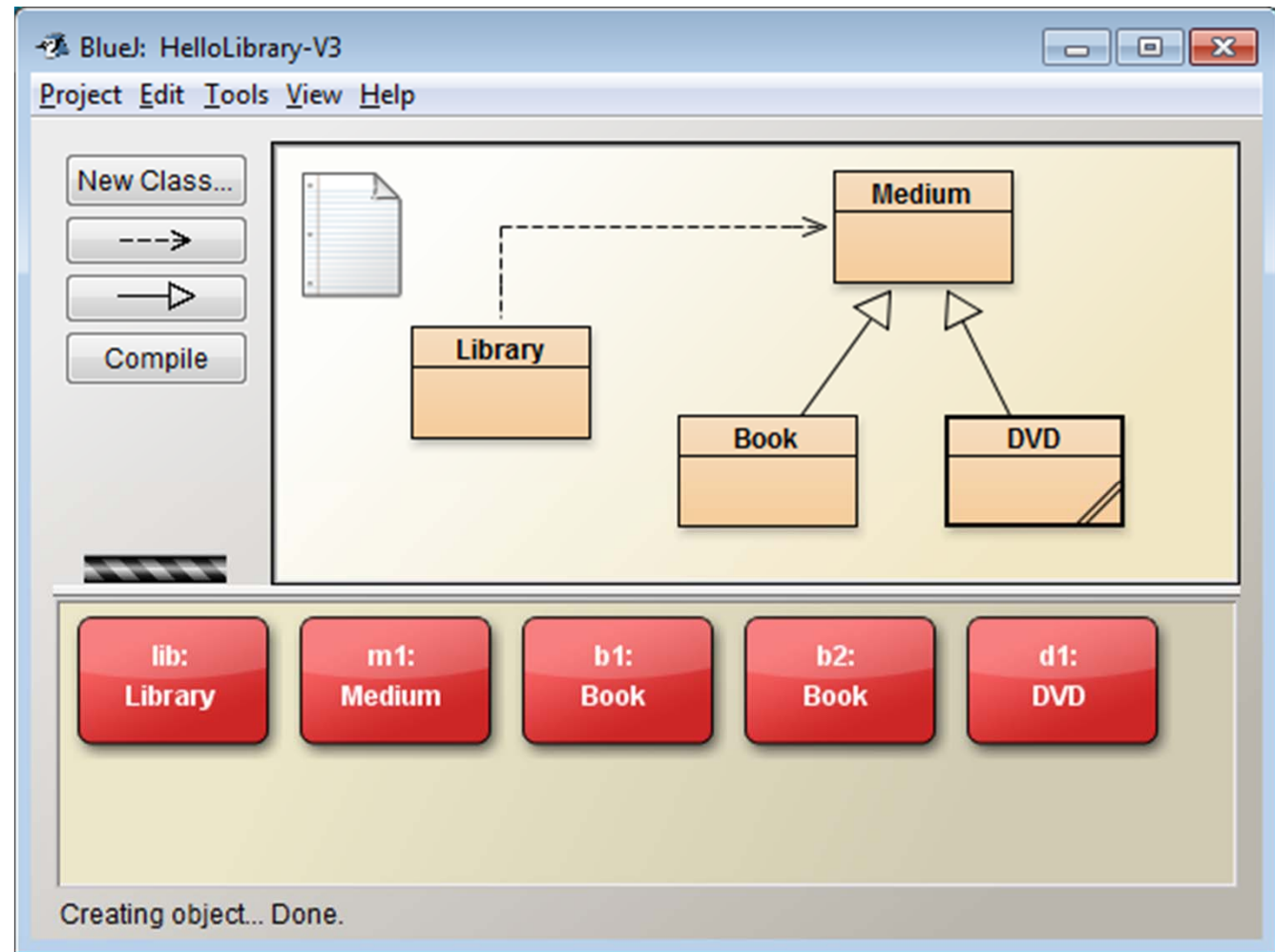


```
public class Book {  
    private String title;  
    private boolean state; // true means lent  
  
    public Book(String title) {  
        this.title = title;  
        state = false;  
    }  
  
    public String toString() {  
        return title;  
    }  
  
    public String getTitle() {  
        return title;  
    }  
  
    public void setState(boolean lending) {  
        this.state = lending;  
        if (lending) {  
            System.out.println("The book " + getTitle() + " is lent.");  
        } else {  
            System.out.println("The book " + getTitle()  
                + " is not lent.");  
        }  
    }  
  
    public boolean getState() {  
        return state;  
    }  
}
```

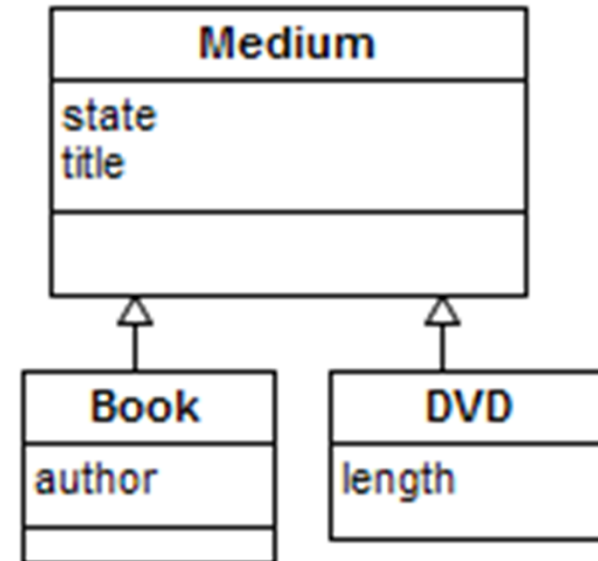
HelloLibrary (U02): Erweiterung um Vererbung



BlueJ: Vererbung



Using inheritance (Wiederholung)



- define one **superclass** : **Medium**
- define **subclasses** for **Book** and **DVD**
- the superclass defines common attributes: **title, state**
- the subclasses **inherit** the superclass attributes
- the subclasses add own attributes: **author** bzw. **length**

Quelle der englischsprachigen Folien:

David J. Barnes & Michael Kölling.

Objects First with Java. A Practical Introduction using BlueJ.

Fifth edition, Prentice Hall / Pearson Education, 2012

<http://www.bluej.org/objects-first/>

Inheritance in Java

```
public class Medium  
{  
    ...  
}
```

change here

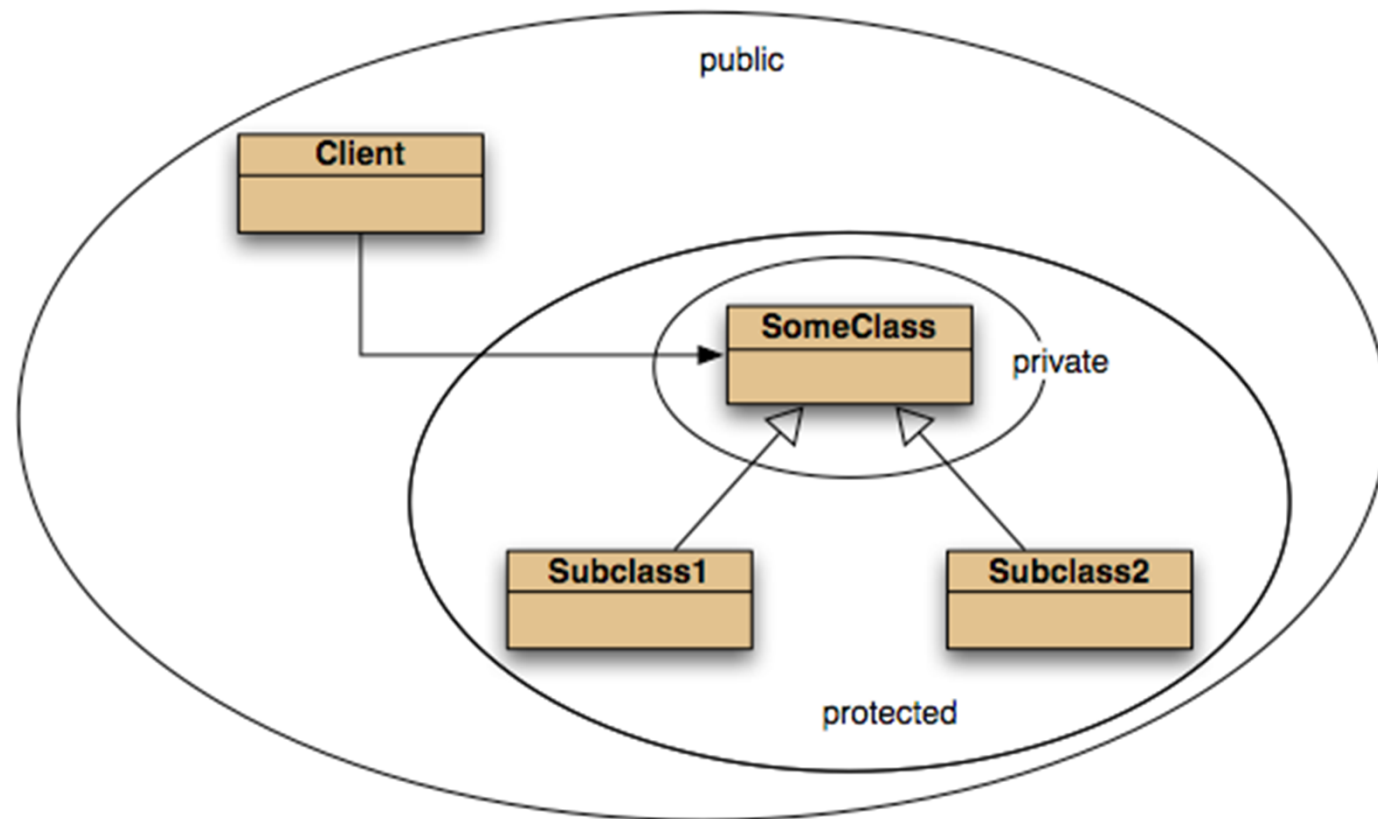
```
public class Book extends Medium  
{  
    ...  
}
```

```
public class DVD extends Medium  
{  
    ...  
}
```

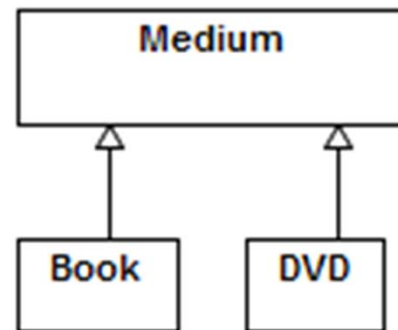
Template für eine Klassendefinition

```
[<visibility>] [abstract/final] class <className>
    [extends <superClassName>]
    [implements <interfaceNames>] {
    // variables
    // constructors
    // methods
        // application specific methods
        // setter/getter
        // helper as toString(), equals(), ...
    }
```


Access levels



Subtyping and assignment



subclass objects may
be assigned to
superclass variables

```
Medium m1 = new Medium(...);
Medium m2 = new Book(...);
Medium m3 = new DVD(...);
```

Static and dynamic type

What is the type of b1?

```
Book b1 = new Book(...);
```

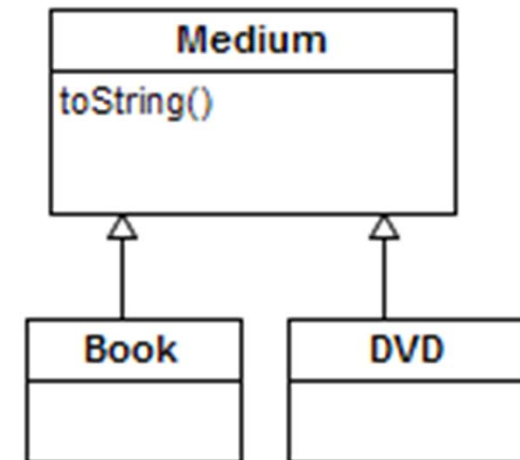
What is the type of m1?

```
Medium m1 = new Book(...);
```

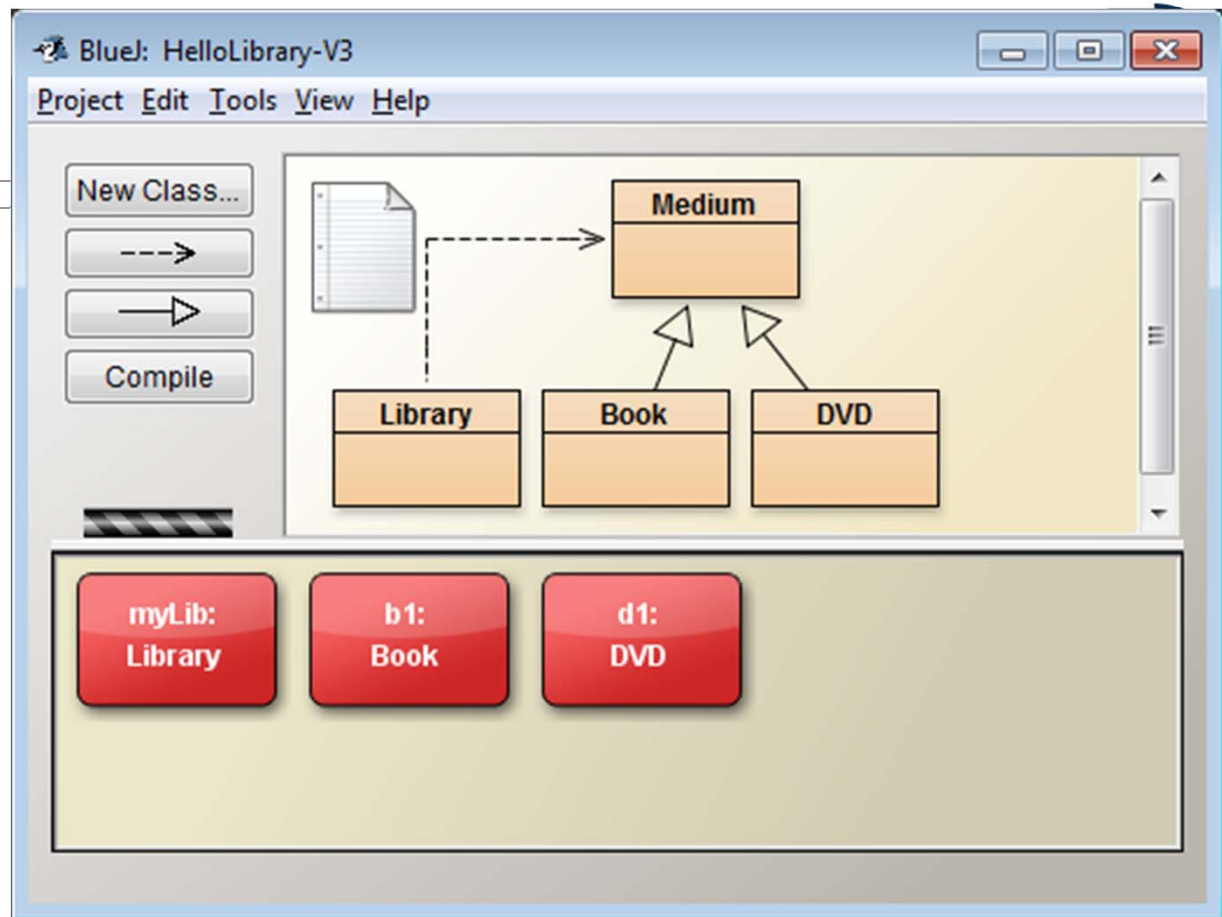
- The declared type of a variable is its **static type**.
- The type of the object a variable refers to is its **dynamic type**.

The problem

- The `toString()` method in `Medium` only prints the common fields (`title`).
- Inheritance is a one-way street:
 - A subclass inherits the superclass fields.
 - The superclass knows nothing about its subclass's fields.



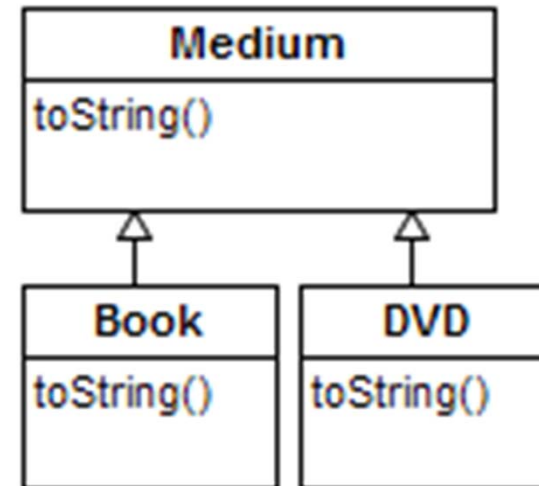
BlueJ: Vererbung



Ersetzung (**Substitution**)
der Methode `toString()`
in Book und DVD

```
BlueJ: Terminal Window - HelloLibrary-V3
Options
Hello, I am a small library for at most 10 media.
A new medium is registered: Titel: Shades of Grey
A new medium is registered: Titel: Shaun das Schaf - der Film
```

Overriding: the solution



- Superclass and subclass define methods with the same signature.
- Each has access to the fields of its class.
- Superclass satisfies static type check.
- Subclass method is called at runtime – it *overrides* the superclass version.

Method lookup

- The variable is accessed.
- The object stored in the variable is found.
- The class of the object is found.
- The class is searched for a method match.
- If no match is found, the superclass is searched.
- This is repeated until a match is found, or the class hierarchy is exhausted.
- Overriding methods take precedence.

Super call in methods

- Overridden methods are hidden ...
- ... but we often still want to be able to call them.
- An overridden method *can* be called from the method that overrides it.
 - `super.method(...)`
 - Compare with the use of `super` in constructors.

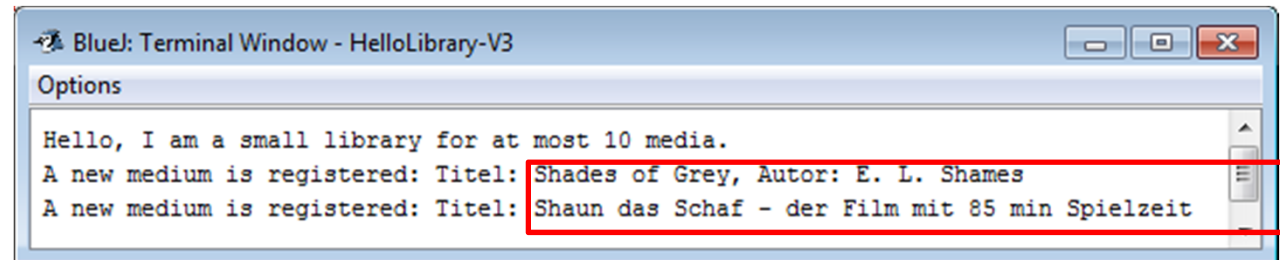
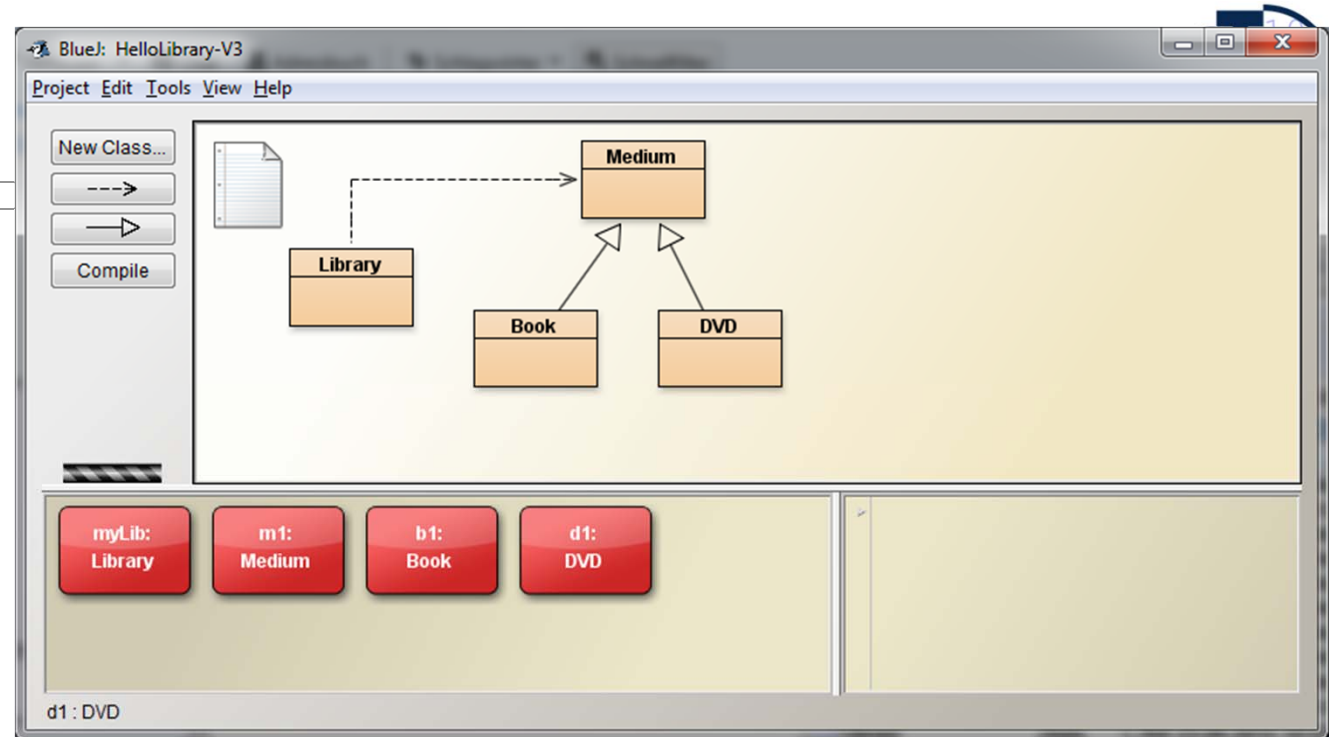
Calling an overridden method

```
public class DVD
{
    ...
    public void toString(){
        return super.toString() +
            " mit " + laenge + " min Spielzeit";
    }
}
```

Polymorphism (Polymorphie)

- We have been discussing *polymorphic method dispatch* (*dynamischer Aufruf*).
- A polymorphic variable can store objects of varying types.
- Method calls are polymorphic.
 - The actual method called depends on the dynamic object type.

BlueJ: Vererbung



```
Options
Hello, I am a small library for at most 10 media.
A new medium is registered: Titel: Shades of Grey, Autor: E. L. Shames
A new medium is registered: Titel: Shaun das Schaf - der Film mit 85 min Spielzeit
```

Überschreiben (**Overriding**) der Methode `toString()` in Book und DVD

Lessons learned

- Inheritance allows the definition of classes as extensions of other classes.
- Inheritance
 - avoids code duplication
 - allows code reuse
 - simplifies the code
 - simplifies maintenance and extending
- Variables can hold subtype objects.
- Subtypes can be used wherever supertype objects are expected (substitution).

Lessons learned

- The declared type of a variable is its static type.
 - Compilers check static types.
- The type of an object is its dynamic type.
 - Dynamic types are used at runtime.
- Methods may be overridden in a subclass.
- Method lookup starts with the dynamic type.
- Protected access supports inheritance.

Hilfe am Rechner gesucht?


- Lernraum für Java
 - ab 2. Mai 2016 jeweils montags 4. DS, E065
- Auditorium
- Java-Kurse des IFSR

Lösung des OOSE1-Rätsels

Lösungswort: **GEHEIMNISPRINZIP**

	A	B	C	D	E	F	G	H	I	J	K	L
1	J	S (B1)	S (C1)	N	D (E1)	T	L	I (H1)	V	E	Ä	Ä
2	K	A	M (C2)	O (D2)	N	A	K	Z	I	R	D	U
3	T	D	E	I	T	R	A	A	A	F	I	G
4	O (A4)	N	A	T	T	I (F4)	S (G4)	T	R	R	T	N (L4)
5	E	U	G	A	Ä	H	E	N	I (I5)	E	E	U
6	B	Z (B6)	S	R	I	C	S	N	T (I6)	E	K	A
7		T	E	E	E	C		S	E	Z (J7)	I	S
8				P	T	H			I	N	U (K8)	P
9				O	I	N			N		Z (K9)	R
10									A		S	G
11											T	
12											N	
13											G	
14												
15	OBJEKT	ZUSTAND	MESSAGE	OPERATION	IDENTITÄT	NACHRICHT	KLASSE	INSTANZ	INVARIANTE	REFERENZ	ZUSTÄNDIGKEIT	AUSPRÄGUNG

	A7
	B2
	C9
	D1
	E11
	F12
	G7
	H8
	I1
	J6
	K7
	L6
	M1
	N5
	O9
	P7
	Q6
	R9
	S11



Schüttelrätsel mit Begriffen zur Objektorientierung (2)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	B	M	S	X	C	G	G	S	U	Y	D	T	R	U	K	S	B	N	U
2	S	R	G	M	H	W	D	N	E	R	N	N	U	P	K	S	C	O	T
3	N	T	N	O	L	D	R	T	H	H	Ü	E	F	Z	S	E	E	I	K
4	Y	U	E	O	K	V	V	L	K	E	E	I	T	A	F	E	E	T	N
5	L	T	A	E	A	R	O	C	A	P	L	H	E	S	T	A	E	E	T
6	K	O	I	I	E	W	R	I	M	L	B	R	E	R	A	B	H	L	R
7	L	A	E	A	S	N	N	T	M	O	A	N	A	G	I	O	I	R	O
8	S		R	N	B	U	I	L	L	M	R	A		S	L	L	J	U	K
9	U		E	T	N	I	I	T	R	P	E	C		I	S	K	K	O	R
10	E		N		O	E	E	H	S	I		I		E	O	R	O	S	O
11	Z		L		L	R		S	E	O		E		L	N		P		S
12	E		G		S	E		E	C					E	A		R		
13			U		S	E		E	S					I	I		S		
14			I		E	E								I	I		T		
15			R		A	D								N					
16					N	N													
17																			