

Beispiel 1 (55 Punkte)

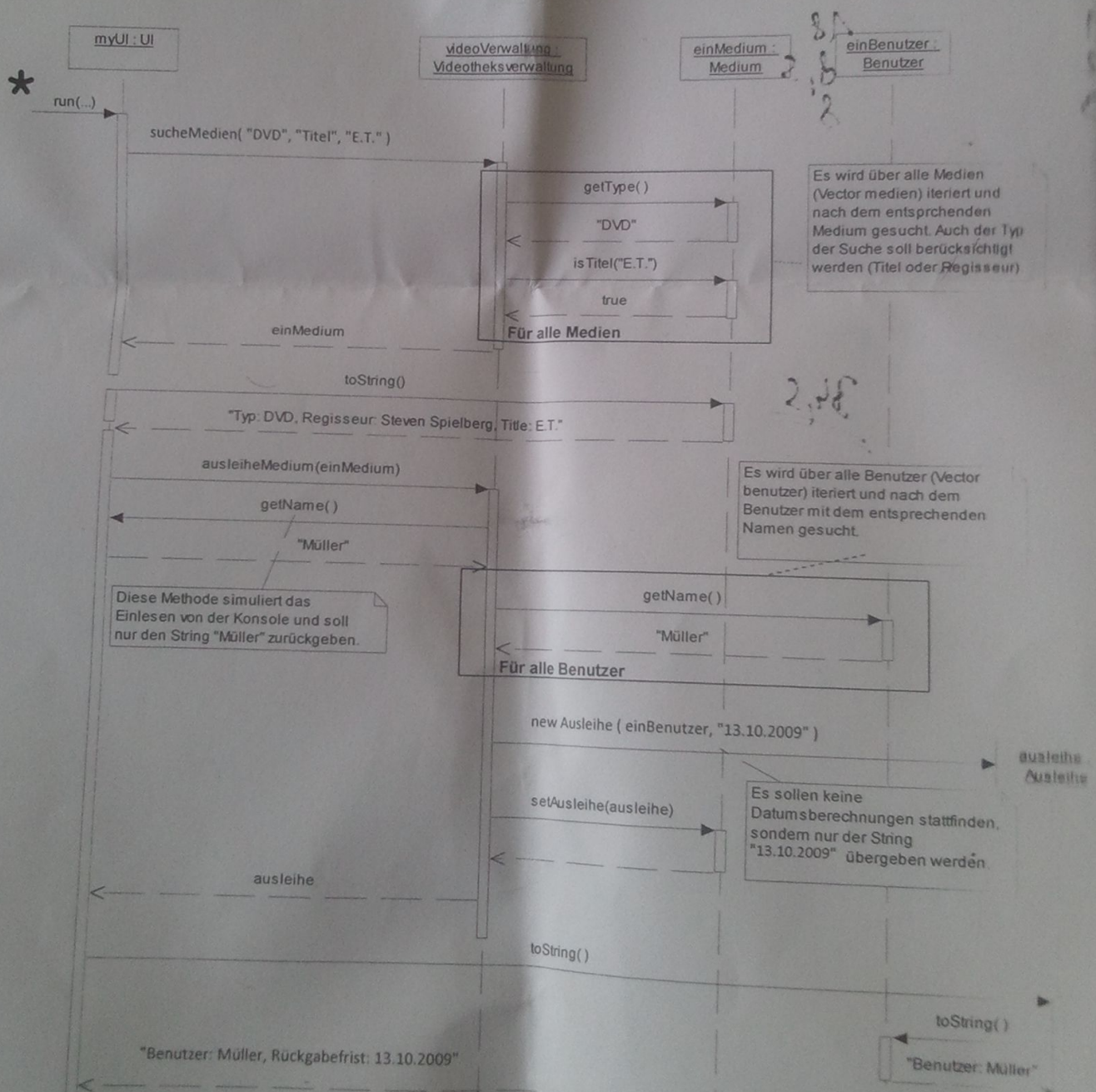
Gegeben sind ein Klassen- und ein Sequenzdiagramm eines Software-Entwurfs. Basierend auf diesen Diagrammen ist folgende Aufgabe zu lösen:

Implementieren Sie die Klassen (Methoden und Attribute) und das Interface entsprechend dem Sequenz-, dem Klassendiagramm und dem angegebenen Code der Klasse "Start" (in dieser Klasse wird das Programm initialisiert).

Der Ablauf im Sequenzdiagramm (markiert mit "*") beginnt in der Methode run der Klasse UI.

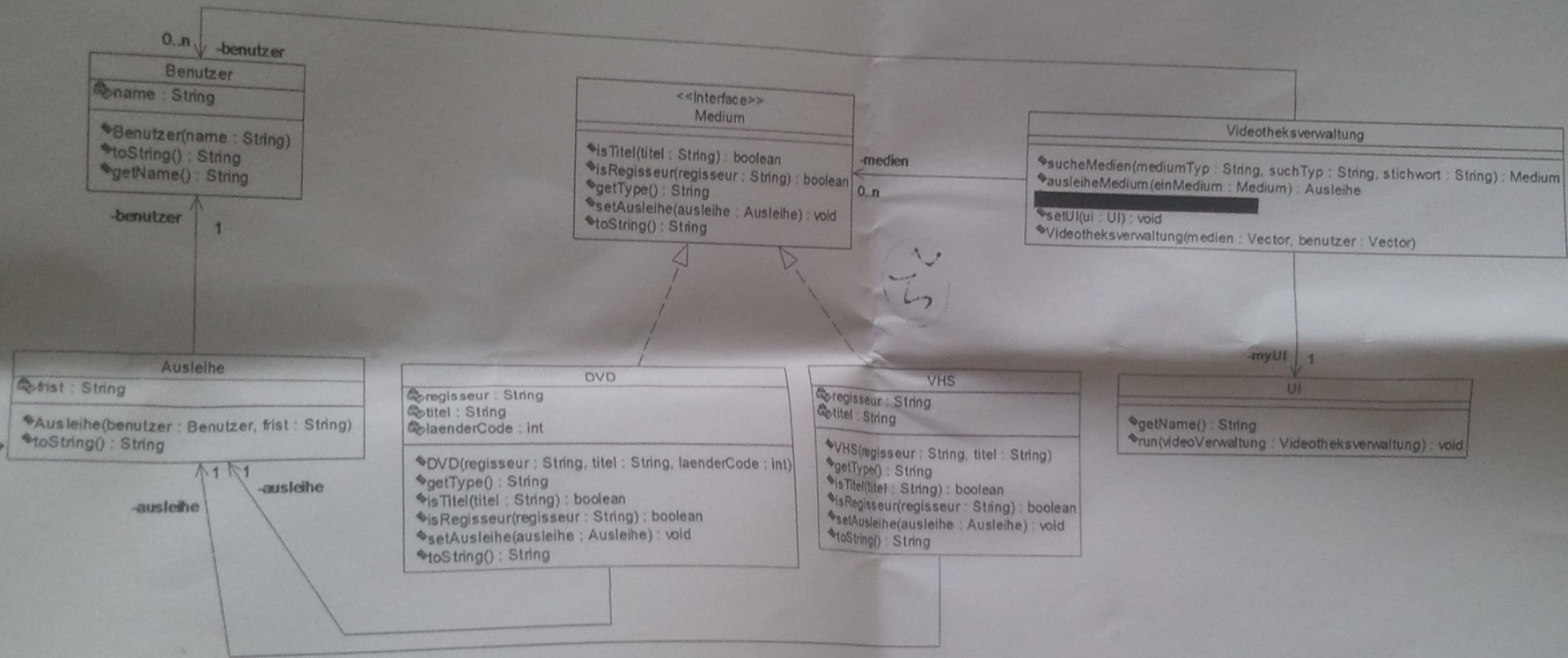
Das Sequenzdiagramm beschreibt den Vorgang einer Medienausleihe in einer Videothek. Das Objekt der Klasse UI (Userinterface) ist die Schnittstelle zum Anwender. Videotheksverwaltung verwaltet eine Liste mit Medien, DVDs oder VHS Kassetten (Vector medien), und sucht nach dem angeforderten Medium. Um ein Medium auszuleihen, muss der Benutzer zuerst nach dem Medium suchen, indem er den Typ des Mediums (VHS oder DVD), Suchtyp (Titel oder Regisseur) und den Suchstring (z.B. „E.T.“ für den Filmtitel) eingibt. Abhängig vom Suchtyp, soll im Programm die entsprechende Methode der Klasse Medium aufgerufen werden. Danach soll der Benutzer seinen Namen angeben. Nachdem der User seinen Namen angegeben hat, wird eine Ausleihe angelegt. Zuletzt werden die Daten der Ausleihe ausgelesen.

Sequenzdiagramm:



Klassendiagramm

Assoziationen mit einem Pfeil (gerichtete Assoziationen) im Klassendiagramm bedeuten, dass jene Klasse beim Pfeilursprung eine Referenz (gespeichert in einer Instanzvariable) zur Klasse am Pfeilende besitzt. Die Referenzen sind den Assoziationen zugeordnet – z.B. Assoziation "DVD – Ausleihe": Ein Objekt von Typ DVD speichert die Referenz zu einem Objekt vom Typ Ausleihe in der Variable `ausleihe`.



Beispiel 2 (25 Punkte)

a) (8 Punkte)

Gegeben sind drei Klassen `Flugobjekt`, `Flugzeug` und `SpaceShuttle` sowie die Schnittstelle `Ufo`.

```
public abstract class Flugobjekt { ... }
public class Flugzeug extends Flugobjekt { ... }
public interface Ufo { ... }
public class SpaceShuttle implements Ufo { ... }

public class Test {
    public static void main( String[] args ) {
        ...
        Flugobjekt fObj = new Flugobjekt(); // 1
        Flugobjekt fZeug = new Flugzeug(); // 2
        Ufo u = new Ufo(); // 3
        Ufo shuttle = new SpaceShuttle(); // 4
        ...
    }
}
```

Ist das Kreieren der vier Instanzen in der Klasse `Test` erlaubt?

Kreuzen Sie an und begründen Sie Ihre Antworten (Antworten ohne Begründung werden nicht gewertet!).

Codezeile:	Erlaubt?		Begründung:
	Ja	Nein	
1:	<input type="checkbox"/>	<input type="checkbox"/>	
2:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4:	<input type="checkbox"/>	<input type="checkbox"/>	

b) (5 Punkte)

Analysieren Sie die folgenden Klassen `Printer` und `NewPrinter`. Die `main`-Methode der Klasse `Test` erstellt Objekte und sendet den Objekten Nachrichten, die gewisse Ausgaben bewirken. Geben Sie die jeweiligen Ausgaben zu den gefragten Zeitpunkten (T1 bis T4, siehe Kommentare in `main`) aus, genauso wie es für den Zeitpunkt T0 gezeigt ist.

```
public class Printer {
    public void show( String val ) {
        System.out.println( "SIMPLE" + val );
    }

    public void show( boolean val ) {
        if ( val )
            System.out.println( "true" );
        else
            System.out.println( "false" );
    }
}
```

```
public class NewPrinter extends Printer {
    public void show( String val ) {
        System.out.println( "NEW" + val );
    }

    public void show( boolean val ) {
        if ( ! val )
            System.out.println( "falsch" );
        else
            System.out.println( "wahr" );
    }
}
```

```
public class Test {
    public static void main( String[] args ) {

        Printer p = new Printer();
        NewPrinter newP = new NewPrinter();

        System.out.println( "START" ); // T0
        p.show( true ); // T1
        p.show( "TRUE" ); // T2
        newP.show( "FALSE" ); // T3
        p = newP;
        p.show( false ); // T4
    }
}
```

Ausgaben:

T0: START

T1: ~~true~~ true

T2: ~~TRUE~~ FALSE

T3: ~~falsch~~ falsch

T4:

c) (6 Punkte)

3,5

Gegeben sind folgende Schnittstellen und Klassen:

```
public interface Intf_1 {
    ...
}

public interface Intf_2 {
    ...
}

public abstract class AbsClass {
    ...
}

public class Class_1 {
    ...
}

public class Class_2 {
    ...
}
```

Kreuzen Sie an, ob die jeweilige Deklaration in Java erlaubt ist?

	Ja	Nein
public class N extends Intf_2 implements Class_1 { ... }	<input type="checkbox"/>	<input checked="" type="checkbox"/>
public class Y extends AbsClass { ... }	<input checked="" type="checkbox"/>	<input type="checkbox"/>
public class X extends Class_1 { ... }	<input checked="" type="checkbox"/>	<input type="checkbox"/>
public class T implements AbsClass { ... }	<input type="checkbox"/>	<input checked="" type="checkbox"/>
public class U extends Class_2 implements Intf_1 { ... }	<input type="checkbox"/>	<input checked="" type="checkbox"/>
public class K implements Intf_1, Intf_2 { ... }	<input type="checkbox"/>	<input checked="" type="checkbox"/>
public class V extends AbsClass implements Intf_1 { ... }	<input type="checkbox"/>	<input checked="" type="checkbox"/>
public class S implements Class_1 { ... }	<input checked="" type="checkbox"/>	<input type="checkbox"/>
public class R extends Intf_1 { ... }	<input type="checkbox"/>	<input checked="" type="checkbox"/>
public class O extends Intf_1 implements AbsClass { ... }	<input type="checkbox"/>	<input checked="" type="checkbox"/>
public class Z implements Intf_1 { ... }	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
public class M extends Class_1, Class_2 { ... }	<input checked="" type="checkbox"/>	<input type="checkbox"/>

d) (6 Punkte)

4

```
public class Lebewesen {...}

public class Mehrzeller extends Lebewesen {...}

public class Saeugetier extends Mehrzeller {...}

public class Mensch extends Saeugetier {...}

public class User {
    public void doSomething( Mehrzeller mz ) {
        ...
    }
}
```

```
public class Test {
    public static void main( String[] args ) {
        Lebewesen lw = new Lebewesen();
        Saeugetier s = new Saeugetier();
        Mensch m = new Mensch();
        User u = new User();

        u.doSomething( lw ); // 1
        u.doSomething( s ); // 2
        u.doSomething( m ); // 3
    }
}
```

Welcher der Aufrufe der Methode User.doSomething() in der Klasse Test ist erlaubt?
Kreuzen Sie an und begründen Sie Ihre Antworten (Antworten ohne Begründung werden nicht gewertet!).

Codezeile:	Erlaubt?		Begründung:
	Ja	Nein	
1:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Die Klasse Mehrzeller ist abgeleitet von Lebewesen, daher hat Klasse Lebewesen nicht Attrib.
2:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Ist erlaubt da Attribute in Superklassen zu finden sind Mehr
3:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

pelganzh