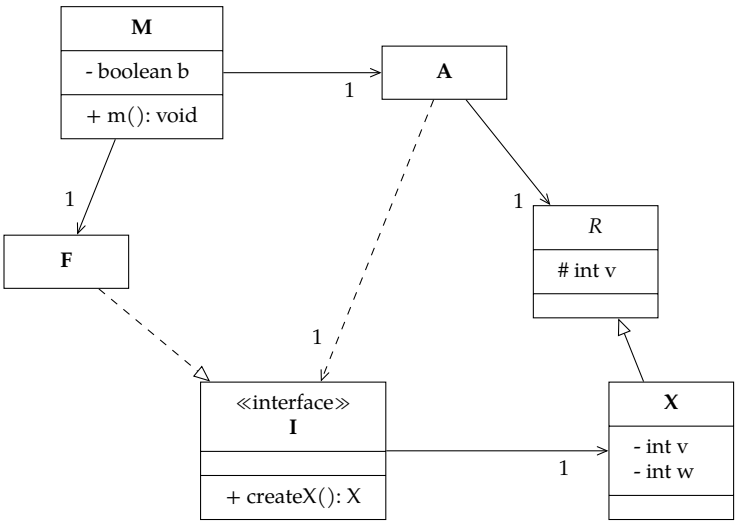
Aufgabe 1: (Objektorientierte Programme und Reverse Engineering)

Gegeben sei das folgende Java-Programm:

```
1  class M {
2    private boolean b;
3    private F f;
4    private A a;
5
6    public void m() {
7      f = new F();
8      a = new A(f);
9      b = true;
10   }
11 }
12
13 class A {
14   private R r;
15   public A(I i) {
16     r = i.createX();
17   }
18 }
19
20 interface I {
21   public X createX();
22 }
23
24 class F implements I {
25   public X createX() {
26     return new X(0, 0);
27   }
28 }
29
30 abstract class R {
31   protected int v;
32 }
33
34 class X extends R {
35   private int v, w;
36   public X(int v, int w) {
37     this.v = v;
38     this.w = w;
39   }
40 }
```

(a) Das Subtypprinzip der objektorientierten Programmierung wird in obigem Programmcode zweimal ausgenutzt. Erläutern Sie wo und wie dies geschieht.

(b) Zeichnen Sie ein UML-Klassendiagramm, das die statische Struktur des obigen Programms modelliert. Instanzvariablen mit einem Klassentyp sollen durch gerichtete Assoziationen mit Rollennamen und Multiplizität am gerichteten Assoziationsende modelliert werden. Alle aus dem Programmcode ersichtlichen statischen Informationen (insbesondere Interfaces, abstrakte Klassen, Zugriffsrechte, benutzerdefinierte Konstruktoren und Methoden) sollen in dem Klassendiagramm abgebildet werden.



(c) Es wird angenommen, dass ein Objekt der Klasse M existiert, für das die Methode `m()` aufgerufen wird. Geben Sie ein Instanzendiagramm (Objektdiagramm) an, das alle nach der Ausführung der Methode `m` existierenden Objekte und deren Verbindungen (Links) zeigt.

(d) Wie in Teil c) werde angenommen, dass ein Objekt der Klasse M existiert, für das die Methode `m()` aufgerufen wird. Diese Situation wird in Abb. 1 dargestellt. Zeichnen Sie ein Sequenzdiagramm, das Abb. 1 so ergänzt, dass alle auf den Aufruf der Methode `m()` folgenden Objekterzeugungen und Interaktionen gemäß der im Programmcode angegebenen Konstruktor- und Methodenrümpfe dargestellt werden. Aktivierungsphasen von Objekten sind durch längliche Rechtecke deutlich zu machen.