

Aufgabe 11.2: Konstruktoren bei abgeleiteten Klassen

11.2.1 Reihenfolge von Konstruktoraufrufen

Die folgende Klasse `TestKonstruktoren` dient zum Testen der Aufrufreihenfolge von Konstruktoren bei abgeleiteten Klassen:

```
// Datei: TestKonstruktoren
public class TestKonstruktoren
{
    public static void main (String[] args)
    {
        System.out.println ("Exemplar von A wird angelegt");
        A aRef = new A();

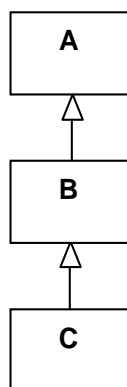
        System.out.println();
        System.out.println ("Exemplar von B wird angelegt");
        B bRef = new B();

        System.out.println();
        System.out.println ("Exemplar von C wird angelegt");
        C cRef = new C();
        System.out.println();
    }
}
```

Schreiben Sie die drei Klassen `A`, `B` und `C`, welche jeweils nur einen Konstruktor ohne Parameter enthalten. Im Konstruktor der Klasse `A` soll folgender Text ausgegeben werden:

```
System.out.println ("Klasse A - Konstruktor ohne Parameter");
```

Schreiben Sie entsprechende Konstruktoren für die Klassen `B` und `C`. Beachten Sie, dass `B` von `A` und `C` von `B` abgeleitet ist. Der Konstruktor der Klasse `C` soll den parameterlosen Konstruktor der Basisklasse `B` durch `super()` explizit aufrufen. Beim Ausführen der Klasse `TestKonstruktoren` wird deutlich, dass es in diesem Fall keinen Unterschied macht, ob der Aufruf `super()` explizit durch den Programmierer eingefügt wird oder implizit durch den Compiler. Das folgende Bild gibt die Klassenhierarchie wieder:



11.2.2 Vererbungshierarchie für Fertigungsgüter, Teil 2

Ausgangspunkt ist das Programm aus Aufgabe 11.1.2. Erweitern Sie alle Klassen der Vererbungshierarchie mit Konstruktoren, um eine einfache Initialisierung der Klassen zu ermöglichen.