

Übung 08: Vererbung

Aufgabe 1: Datenstrukturen

In dieser Aufgabe implementieren Sie eine Datenstruktur die es erlaubt Boolesche Formeln als Bäume darzustellen. Dazu werden im Folgenden verschiedene Klassen von der Oberklasse `Knoten` abgeleitet:

```
public abstract class Knoten {
    public abstract Boolean getWert();

    @Override
    public String toString(){
        if(getWert()){
            return "1";
        }else{
            return "0";
        }
    }
}
```

Wie sie sehen können, handelt es sich dabei um eine s.g. abstrakte Klasse. Diese können nicht instanziiert werden, das bedeutet es lassen sich keine Objekte mit `new` daraus erzeugen. Trotzdem können sie bereits Methoden und Attribute bereitstellen, die sich dann mittels Vererbung in anderen Klassen weiterverwenden lassen. Außerdem ist es möglich abstrakte Methodenrumpfe vorzudefinieren, die zwar hier keine Implementierung haben, aber von den Klassen implementiert werden müssen, die von dieser abstrakten Klasse erben. Hier ist das z.B. die Methode `getWert()`. Klassen die von `Knoten` ableiten sind entweder selber abstrakt oder müssen die Methode `getWert()` überschreiben.

Folgende Klasse `Startup` verwendet bereits die von Ihnen zu implementierende Datenstruktur. Hier wird eine einfache Boolesche Formel erzeugt `Baum = a OR (b AND c)`.

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

        Basisereignis a = new Basisereignis(true);
        Basisereignis b = new Basisereignis(false);
        Basisereignis c = new Basisereignis(true);

        Oder baum = new Oder();
        baum.addKnoten(a).addKnoten(new Und().addKnoten(b).addKnoten(c));
        System.out.println(baum);
    }
}
```

- Implementieren Sie zunächst die Klasse `Basisereignis` die von der Klasse `Knoten` erbt. Implementieren Sie zusätzliche Funktionalität so, dass ein `Basisereignis` den Wert `true` oder `false` annehmen kann.
- Implementieren Sie die abstrakte Klasse `ErweiterbarerKnoten` die als Attribut ein Array von `Knoten` hat, die Unterknoten, und eine Methode `addKnoten(Knoten k)` bereitstellt mit Hilfe derer sich ein neuer `Knoten k` zu den Unterknoten hinzufügen lässt. Diese Methode gibt ihr eigenes Objekt zurück, damit sie sich wie in oben dargestellter Weise aufrufen lässt.
- Implementieren Sie die Klasse `Oder`, die von der Klasse `ErweiterbarerKnoten` erbt. Überschreiben Sie die Methode `getWert` so, dass eine Veroderung der Werte der Unterknoten entsteht.
- Implementieren Sie die Klasse `Und` in ähnlicher Weise wie die Klasse `Oder` aus Aufgabe c.
- Implementieren Sie geeignete Testfälle für die Klassen `Basisereignis`, `Oder` und `Und`.