

Nachdenkzettel: Clean Code

1. Gegeben sind folgende Klassendefinitionen:

```
class Formularfeld;  
class Textfeld extends Formularfeld;  
class Zahlfeld extends Formularfeld;  
class TextUndZahlFeld extends Formularfeld;  
class TextfeldOCR extends Textfeld;  
class ZahlfeldOCR extends Zahlfeld;  
class TextUndZahlFeldOCR extends TextUndZahlFeld;  
class TextfeldSonderZ extends TextUndZahlFeld;  
class TextfeldOCR_SonderZ extends TextUndZahlFeldOCR;  
//...
```

Jede weitere Eigenschaft oder Spezialisierung führt zu vielen neuen Klassen durch Kombination. Die Folge ist explosives Anwachsen der Zahl der Klassen mit identischem Code. Wie können Sie dies umgehen?

2. Korrekte Initialisierung und Updates von Objekten

Betrachten Sie die Klasse `Address`:

```
public class Address {  
    private String city;  
    private String zipCode;  
    private String streetName;  
    private String houseNumber;  
  
    public void setCity (String c) {  
        this.city = c;  
    }  
    public void setZipcode (String z) {  
        this.zipCode = z;  
    }  
  
    // weitere Setter für alle Attribute  
}
```

Was kann bei der initialisierung von `Address`-Objekten schiefgehen? Wie korrigieren Sie dies? Wie ist Ihre Meinung bzgl. der Parameterbenennung der aufgeführten `set`-Methoden?

1. Code sollte niemals doppelt vorkommen, dementsprechend könnten vermutlich mehrere Klassen zusammengefasst werden.

Im Falle eines Formularfeldes könnte man zum Beispiel mit Generics arbeiten um direkt anzugeben, welcher Datentyp eingegeben werden kann.

Außerdem sollten Interfaces verwendet werden.

2.

- a. Es kann passieren, dass nicht alle Attribute gesetzt werden. Es sollte ein Konstruktor implementiert werden, welcher die Attribute über gegebene Parameter setzt. Dann können auch die Setter private gesetzt werden und es kann nicht von außerhalb auf die Attribute zugegriffen werden.
- b. Die Parameter sollten genauso wie die Attribute heißen. Also anstatt c sollte es city sein und so weiter. Dadurch ist klarer was man als Parameter angeben muss.