**Refactoring-Übungen**

**Aufgabe 1**

**Vorgehen**

1. Alle Klassen grob anschauen
2. Was sind die Probleme in Address und Person?
3. Wie geht man schrittweise vor? Welche Refactorings sind nützlich?
4. Klasse AddressPrinterApp ausführen
5. Wie finde ich Referenzierungen im Project (Cmd, Shift + G)?

Klasse Address

* Remove unused attribute street
* ***Encapsulate Field*** für city und country
  + Attribute private
  + Setter / Getter einführen
* Attribute können final sein, da nur im Konstruktor zugewiesen
* final für jedes Attribut
* ***Generate toString()***

Klasse Person

* Zuweisungsfehler in personInit() bereinigen
* Extrahieren der Methode printAddress(Address) aus printAddresses()
* Methode printAddress(Address) verschieben: ***Move***
* Methode setFirstAddress(Address) umbenennen: ***Rename***
* ***Encapsulate Field*** für name und age
  + Attribute private
  + Setter / Getter einführen
  + 2 x Inline in personInit()
* private final für jedes Attribut addresses
* ***Generate constructor using fields***
* Konstruktor mit dort personInit() aufrufen
* Dann ***Inline***
* final für Attribut name

**Optional**

* Design-Problem: Wie kann man älter werden? Modellierung als Geburtstag

**Aufgabe 2**

**public** **class** TimeStampUtils

{

**public** **static** String createTimeStampString(**final** MyTimePeriod currentPeriod,

**final** MyFrequency frequency)

{

**final** LocalDateTime start = currentPeriod.getDateTime();

**final** **int** divisor = frequency == MyFrequency.***P1M*** ? 1 : 3;

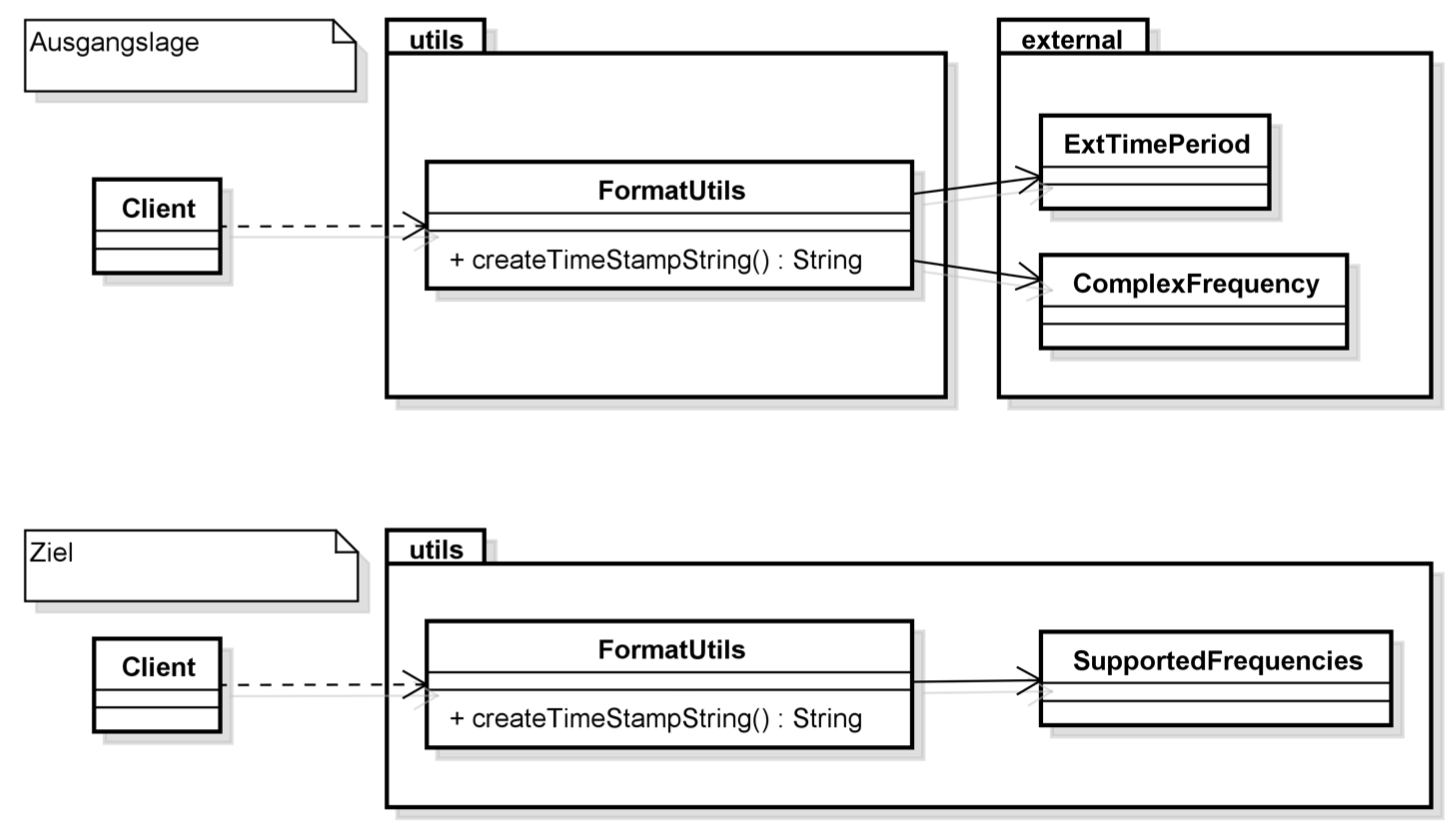
**final** String addition = frequency == MyFrequency.***P1M*** ? "" : "Q";

**final** **int** value = ((start.getMonthValue() - 1) / divisor + 1);

**return** start.getYear() + "-" + addition + value;

}

}



Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**Tipps zu möglichen Refactoring-Schritten**

**Auflösen der Abhängigkeiten**

**Schritt 1: Hilfsvariable einführen (EXTRACT LOCAL VARIABLE)**

**Schritt 2: Abhängigkeit zur Frequenz entfernen (EXTRACT METHOD)**

**Schritt 3: Inlining des Methodenaufrufs (INLINE)**

**Schritt 4: Abhängigkeit zur Klasse ExtTimePeriod entfernen**

**Schritt 5: Für konsistente Parameterreihenfolge sorgen**

**Schritte 6: Inlining des Methodenaufrufs**

**Tests ergänzen**

**Unzulänglichkeit: Boolescher Parameter**

**Vereinfachungen**

**Siehe Text**

**Verlagern von Funktionalität**

**Siehe Text**