**Zusammenfassung OCL Smells und Refactorings**

**OCL:**

Die Object Constraint Language, kurz OCL, ist eine an Smalltalk angelehnte Sprache, mit der Einschränkungen (Constraints) für UML-Modelle beschrieben werden können. Dabei unterscheidet man verschiedene Arten von Constraints, darunter Invariants, Preconditions, Postconditions und Guards. Dabei müssen Preconditions beim Start einer Operation erfüllt sein, während Postconditions am Ende einer Operation erfüllt sein müssen. Invarianten müssen jederzeit gelten, und Guards müssen am Anfang eines Zustandsüberganges gültig sein. Die OCL wird hauptsächlich bei Klassendiagrammen verwendet, um sinnvolle Einschränkungen für die Attribute der im Diagramm modellierten Klassen zu definieren, wie beispielsweise die Einschränkung, dass das Alter einer Person nicht negativ sein darf. Aufgrund dieser Eigenschaften wird die OCL nicht als eigenständige Sprache, sondern mehr als Ergänzung zu UML verwendet, da die durch die OCL definierten Einschränkungen nicht oder nur schwer in UML realisiert werden können. Für die OCL existieren natürlich auch Smells sowie Refactorings, welche diese verbessern können.

**OCL Smells:**

Implies Chain: Eine Verkettung von mindestens zwei implies-Operationen. Diese kann durch das dazugehörige Refactoring "Replace Implies Chain by a Single Implication" verbessert werden.

ForAll Chain: Eine Verkettung von mindestens zwei ForAll-Operationen. Diese kann durch das Refactoring "Replace ForAll-Chain by Navigations" verbessert werden.

Duplication: Doppelte und somit redundante Operationen, die häufig durch copy/paste entsteht. Dies können beispielsweise Berechnungen sein, die mehrmals ausgeführt werden. Kann mit "Add Operation Definition" und "Replace Expression by Operation Call" behoben werden.

Magic Literal: Magic Literal beschreibt einen im Code auftauchenden festen Wert, der nicht weiter erklärt wird. Dieser Smell kann durch die Refactorings "Add Variable Definition" und "Replace Expression by Variable" verbessert werden.

Verbose Expression: Dieser Smell besagt, dass ein längerer Ausdruck gefunden wurde, der einfacher und somit kürzer geschrieben werden kann. Refactorings, die diesen Smell beheben, sind "Change Context", "Simplify Operation Calls" und "Change Initial Navigation".

**Refactorings:**

Replace Implies Chain by a Single Implication: Bei diesem Refactoring wird eine Verkettung von implies-Operationen durch eine einzige, gleichwertige implication ersetzt.

Replace ForAll-Chain by Navigation: Bei diesem Refactoring wird eine Verkettung von ForAll-Operationen durch neue Navigationen ersetzt, bis am Ende nur noch ein ForAll-Operator benutzt wird.

Add Operation Definition: Eine neue Methode wird erstellt. Diese wird häufig aus bestehendem Code extrahiert, um die Redundanz zu verringern.

Replace Expression by Operation Call: Redundante Ausdrücke im Code werden durch einen Methodenaufruf ersetzt, der die gleiche Funktionalität erfüllt. Dieses Refactoring wird häufig zusammen mit "Add Operation Definition" benutzt, um Redundanz zu verringern.

Add Variable Definition: Bei diesem Refactoring werden im Code auftauchende, nicht erklärte Zahlenwerte in Variablen gespeichert.

Replace Expression by Variable: Ein im Code auftauchender, nicht erklärter Zahlenwert wird durch eine Variable mit dem gleichen Wert ersetzt, um die Lesbarkeit und Wartbarkeit des Codes zu verbessern.

Remove/Add Redundant Brackets: Bei diesen Refactorings werden Klammern um Operationen entfernt, falls diese nicht benötigt werden, und hinzugefügt, falls sie die Lesbarkeit verbessern, da sie die Präzedenzen besser darstellen.

Change Context: Bei diesem Refactoring benutzt man einen anderen Context, um einen längeren Ausdruck etwas abkürzen zu können.

Simplify Operation Calls: Längere Ausdrücke, die aus mehreren, nicht zwingend benötigten Methodenaufrufen bestehen, werden durch weniger Methodenaufrufe kürzer geschrieben.

Change Initial Navigation: Ein längerer Ausdruck wird verkürzt, indem man statt der ursprünglichen Navigation eine andere Navigation verwendet, für die der entsprechende Ausdruck kürzer ist.