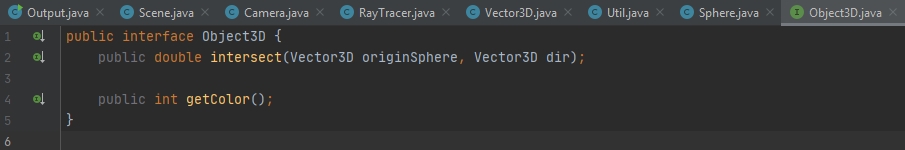
# Metriken am Beispiel eines Java Raytracers

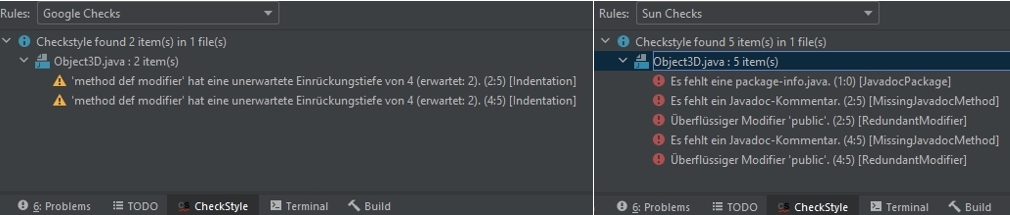
Ich habe mich im Zuge der Abgabe mit dem Werkzeug **Checkstyle** beschäftigt und meine ersten Gehversuche mit einem Java Raytracer auf Auffälligkeiten und Fehler im Programmcode überprüft.



**Abbildung 1:** Codebeispiel des Interface Object3D im Java Raytracer

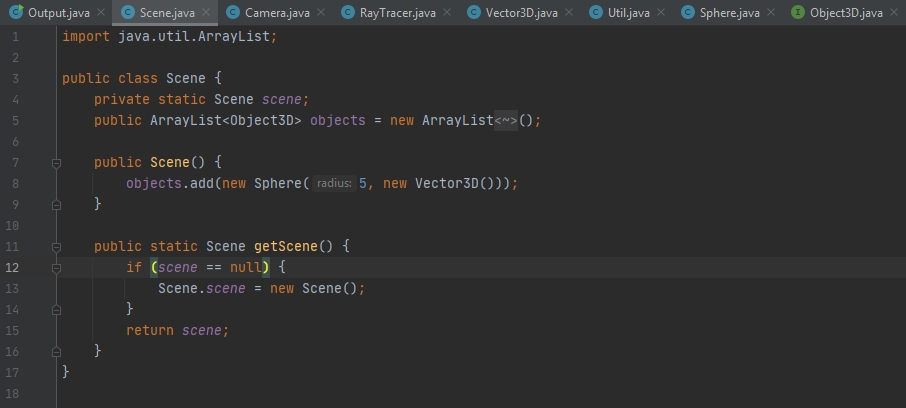
Zunächst ist mir aufgefallen, dass Checkstyle nach zwei verschiedenen **Rules** überprüfen kann. Zur Auswahl stehen *Sun Checks* und *Google Checks*, welche unterschiedliche Ergebnisse nach der Überprüfung liefern.

Während *Sun Checks* keine Warnung bezüglich der Einrücktiefe ausweist, fordert *Google Checks* eine Einrücktiefe von **zwei** Zeichen und weist eine Warnung aus. *Sun Checks* bemängelt im verwendeten Programmbeispiel hingegen die fehlenden **Javadoc-Kommentare** und weist diese Auffälligkeit als **Error** aus. **Abbildung 2** zeigt den direkten Vergleich der beiden **Rules** zum überprüften Interface Object3D.

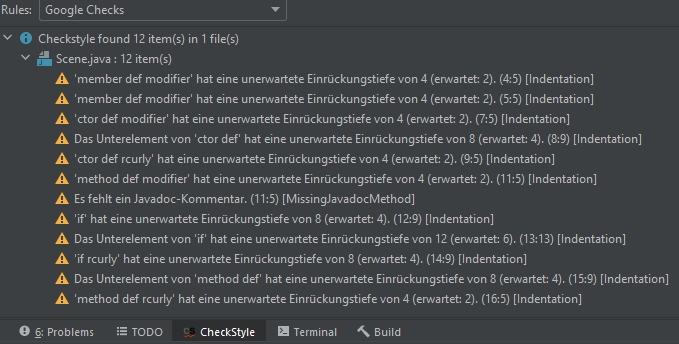


**Abbildung 2:** Direkter Vergleich des Sun Checks und Google Checks im Hinblick auf die erwartete Einrückungstiefe und die fehlenden Kommentare

Während *Sun Checks* in der Auswertung vor allem inhaltliche Punkte, wie beispielsweise die logische Verknüpfung und die korrekte Typ-Bezeichnung beurteilt, setzt Google Checks den Schwerpunkt auf die Leserlichkeit und Konsistenz des Codes und bemängelt zum Großteil fehlende Leerzeichen und Abstände in meinem Programm. **Abbildung 4** zeigt die Warnungen des *Google Checks* am Beispiel der Scene Klasse des Raytracers.



**Abbildung 3:** Scene Klasse des Raytracers



**Abbildung 4:** Google Check der Scene Klasse