```
/************************************************
UV Objektorientierte Programmierung - Aufgabe 3

Kevin Jahreiß, David Karasz, Melanie Urban
************************************************/
```

Teil A

Um etwa in einer generischen Klasse ein Array des generischen Typs E zu erstellen, kann die Anweisung

private E[] hh = new E[5];

verwendet werden. Dann meldet der Compiler jedoch den Fehler „generic Array creation“!
Finden Sie heraus, warum dies so ist und fassen Sie dies kurz zusammen.

Typlöschung (type erase)

Um zu verdeutlichen, wieso der Compiler diesen Fehler ausgibt, muss zuerst auf die Typlöschung bei Generics eingegangen werden.

Der Compiler wandelt jeden Typparameter in den Obertyp Object um und löscht jegliche Typinformationen. Dies geschieht aus Kompatibilitätsgründen, da vor Einführung von Typparametern in Java 5 die Klasse „Object“ zum Speichern eines Objektes mit beliebigem Typ verwendet wurde. So können alte Quellcodes weiterhin kompiliert werden und neue Klassen sind auch von alten Compilern lesbar.
Zur Laufzeit wird also mit sogenannten Raw-Types gearbeitet und erst zur Compile-Zeit findet eine Typprüfung durch den Compiler statt, um die Zuweisung verschiedener Typen zu unterbinden (Fehler „incompatible types“).

Dies kann jedoch zu mehreren Problemen führen. Es kann z.B. kein Objekt eines bestimmten Parametertyps erstellt werden:

new T;

Zum einen sind die Objektgröße und somit der reservierte Speicherplatz vom Datentyp abhängig und zum anderen wird in jedem Objekt bei Erstellung der gewünschte Typ notiert. Hier würde durch die Typlöschung nur ein „Object“ erstellt werden, aber kein Objekt mit einem Parametertyp.

Generics und Arrays

Man kann auch kein Array eines Parameters erzeugen. Wird einem Array ein nicht kompatibler Wert zugewiesen, wird ein Objekt der Klasse ArrayStoreException() erzeugt. Da im Array zur Laufzeit kein Elementtyp vermerkt ist, kann die Kompatibilität nicht überprüft werden. Um Typsicherheit zu gewähren, ist ein solches Konstrukt nicht möglich.
So führt die Erzeugung eines Arrays zum Compilerfehler, die Deklaration allein würde nur eine Warnung ausgeben.

Lösungsansätze

Eine Möglichkeit ist, stattdessen ein Array vom Typ Object zu erstellen und es nach Parametertyp E zu casten.

E[] hh = (E[]) new Object[5];

Dies führt zu einer Warnung („Type safety: Unchecked cast“), da der Objekttyp nicht überprüft werden kann, jedoch ist die Erzeugung trotzdem durchführbar.
Die Warnung kann mit dem vorangestellten Zusatz *@SuppressWarnings("unchecked")* unterdrückt werden.

Quellen

Brass, P. D. S., 2013. *Objektorientierte Programmierung.* [Online]
Available at: http://users.informatik.uni-halle.de/~brass/oop13/pl_gener.pdf
[Zugriff am 02 04 2020].

Dr. Steffen Jost, S. B., 2017. *Lehr- und Forschungseinheit für theoretische Informatik.* [Online]
Available at: https://www.tcs.ifi.lmu.de/lehre/ws-2017-18/sep/material/12_generics.pdf
[Zugriff am 02 04 2020].

Ullenboom, C., 2010. *Java ist auch eine Insel.* 9. Auflage Hrsg. s.l.:Rheinwerk Computing.