



Programmieren II  
Sommersemester 2016  
Übungsblatt 4 (2 Punkte)

### „Sandwich-Position“

In großen Softwareprojekten kommen Sie oft in eine Art Zwickmühle: Da gibt es einerseits den Kunden, der Ihren Code nutzen möchte, und andererseits sind auch Sie Kunde/Klient von einem oder mehreren Code-Zulieferern. Sie stecken also in verschiedenen Softwareverträgen gleichzeitig:

- Ein Vertrag mit dem Klienten, für den Sie Ihren Code schreiben
- Ein Vertrag mit dem/den Zulieferern, deren Code Sie nutzen möchten

In der vorliegenden Übungsaufgabe stecken Sie in einer solchen Sandwich-Position und müssen insbesondere die Behandlung von Exceptions auf die richtige Spur bringen.

### Aufgabe „exercise04“ (Abgabe vom 19.4. - 25.04.2016, bZv-relevant)

Sie sind in einem Softwarehaus für die Verbindung eines zugekauften GPS-Moduls mit der hausintern entwickelten Software für ein mobiles Navigationssystem verantwortlich. Somit haben Sie zum einen eine Schnittstelle zum Entwicklerteam, das die Software für das Navigationssystem implementiert (Ihr Klient), als auch eine Schnittstelle zum zugekauften GPS-Modul.

Der Leiter des Entwicklungsteams hat Ihnen per Mail mitgeteilt:

*Wir benötigen von Ihnen eine Klasse `NaviCon`, die das Java Interface `NaviConInterface` erfüllen muss. Wir werden diese Klasse unmittelbar nach dem Einschalten des Navigationsgeräts mit dem Standard-Konstruktor (ohne Parameter) instanziiieren. In der Klasse erwarten wir zwei Methoden, die wir anschließend aufrufen werden:*

- `void setGeoLocationManager(GeoLocationManager geoManager)`  
*Mit dieser Methode geben wir Ihnen direkt nach dem Instanziiieren Ihrer Implementierungsklasse eine Referenz auf das Interface zum verbauten GPS-Modul, auf das Sie ja zugreifen müssen. Sie können über diese Referenz Methoden aufrufen.*
- `Point getCurrentPosition()`  
*Mit dieser Methode möchten wir anschließend mit Hilfe Ihrer Klasse jederzeit die aktuelle Position abfragen können. Wie Sie diese Info aus dem GPS-Modul herausholen, ist Ihre Sache.*

*Sie können sich natürlich darauf verlassen, dass wir Ihnen vor der ersten Positionsabfrage in jedem Fall die Referenz auf das Interface zum GPS-Modul mitgeteilt haben, das wir ebenfalls für Sie anlegen. Viele Grüße, XYZ*

Im Handbuch zum GPS-Modul haben Sie folgenden Text gefunden:

*Herzlichen Glückwunsch zum Erwerb Ihres neuen GPS-Moduls. Sie haben eine gute Wahl getroffen. Die Programmierschnittstelle zu unserem Modul ist sehr einfach und durch das mitgelieferte Java Interface `GeoLocationManager` beschrieben. Es besteht im Wesentlichen aus einer Methode:*

- *`Point currentLocation()`*

*Mit dieser Methode können Sie jederzeit Ihre aktuelle Position vom GPS-Modul erfahren.*

*Hinweis: In der aktuellen Softwareversion kann es gelegentlich zu folgenden Exceptions kommen:*

- *`IllegalAccessException`*

*Es gab eine kurzzeitige Störung, bitte wiederholen Sie den Aufruf einfach noch einmal.*

- *`InstantiationException`*

*Es ist ein Reset des Moduls notwendig. Dazu bitte kurz die Spannung unterbrechen.*

In einer Rücksprache mit dem Entwicklerteam sind Sie bzgl. der Exceptions so verblieben:

- Falls es doch einmal vorkommen sollte, dass `getCurrentPosition` aufgerufen wird, bevor Sie via `setGeoLocationManager` eine Referenz auf das GPS-Modul erhalten haben, signalisieren Sie dies mit einer `NullPointerException` in der Methode `getCurrentPosition`, die als Exception-Text „NoGeoManager“ enthalten soll.
- Die `IllegalAccessException` des GPS-Moduls wird von Ihnen entschärft. Das Team bekommt davon nichts mit.
- Die `InstantiationException` des GPS-Moduls müssen Sie propagieren, da nur das Team die Chance hat, programmgesteuert die Spannungszufuhr kurz zu unterbrechen. Ihnen wird anschließend per `setGeoLocationManager` eine neue gültige Referenz auf das GPS-Modul mitgeteilt.

Außerdem haben Sie glücklicherweise noch bemerkt, dass sich der Koordinatenursprung des GPS-Moduls von dem des Navigationssystems unterscheidet. Sie müssen daher alle Punkte des GPS-Moduls um den Vektor (1;-1) verschieben, ehe Sie sie an die Navi-Software weitergeben können.

#### **Ihre Aufgabe:**

Implementieren Sie im Package `exercise04` eine Klasse `NaviCon`, die das Geforderte leistet. Die Interfaces `GeoLocationManager` und `NaviConInterface`, finden Sie im Verzeichnis `Übungsaufgabe_4`. Außerdem ist die Hilfsklasse `Point` und eine Klasse `TestProgramm` enthalten, deren `main`-Methode einen groben Test durchführt. Bitte belassen Sie die Interfaces, die Klasse `Point` und die Klasse `TestProgramm` unverändert.

**Hinweis:** Aufgrund der noch fehlenden Klasse `NaviCon` ist das bereitgestellte Projekt zunächst nicht übersetzbar. Erst nach einer entsprechenden Ergänzung kann mit Hilfe der `main`-Methode getestet werden.