

Inhaltsfeld Algorithmen

Kontrollstrukturen in Java: Anwendung von zählergesteuerten Schleifen

Ein Beitrag von Johann-Georg Vogelhuber



© RAABE 2021

© Sitthiphong/Stock/Getty Images Plus

Zählergesteuerte Schleifen sind eine grundlegende Kontrollstruktur in jeder imperativen Programmiersprache. Lassen Sie Ihre Schülerinnen und Schüler ausgehend von einer realitätsnahen Handlungssituation einen typischen Softwareentwicklungsprozess durchlaufen und so ein vertieftes Verständnis für die Funktionsweise und Verwendung von Schleifen entwickeln.

KOMPETENZPROFIL

Klassenstufe:	10
Dauer:	5–8 Unterrichtsstunden
Lernziele:	Die Lernenden ... 1. implementieren, indem sie eine bestehende Software mithilfe der for-Schleife um eine neue Funktionalität erweitern, 2. kommunizieren und kooperieren, indem sie sich gegenseitig konstruktives Feedback zu ihren Lösungsansätzen geben.
Kompetenzen:	Implementieren, Kommunizieren und Kooperieren
Themengebiete:	Kontrollstrukturen, <i>Java</i> , zählergesteuerte Schleifen

Auf einen Blick

Benötigte Materialien

- Für Programmieraufgaben: eine IDE wie *Visual Studio Code* oder *BlueJ*
- Für Programmieraufgaben: ausreichend PC-/Laptop-Arbeitsplätze
- Für optional einsetzbare interaktive *Kahoot!*- oder *LearningSnacks*-Übungen: mobiles Endgerät
- Quellcode-Dateien *BlueJ*: Ordner *BlueJ*
 - Unterordner *CardioTrainer – Einstieg*: *CardioTrainer.java*, *package.bluej*
 - Unterordner *CardioTrainer – MitMehrKommentaren*: *CardioTrainer.java*, *package.bluej*
 - Unterordner *CardioTrainer – Lösung – Niveau_Mittel/Schwierig*: *CardioTrainer.java*, *package.bluej*
 - Unterordner *SIRModell – Lösung*: *SIRModell.java*, *package.bluej*
 - Unterordner *SIRModell – Hilfematerial*: *SIRModell.java*, *package.bluej*
- Quellcode-Dateien *Visual Code Studios*: Ordner *VSCode*
 - Unterordner *CardioTrainer – Einstieg*: *CardioTrainer.java*, *package.bluej*
 - Unterordner *CardioTrainer – MitMehrKommentaren*: *CardioTrainer.java*, *package.bluej*
 - Unterordner *CardioTrainer – Lösung – Niveau_Mittel/Schwierig*: *CardioTrainer.java*, *package.bluej*
 - Unterordner *SIRModell*: *SIRModell.java*, *package.bluej*
 - Unterordner *SIRModell – Hilfematerial*: *SIRModell.java*, *package.bluej*

Einstieg

- M 1 **Ein Überblick für die Trainings-App – Analyse des vorhandenen Quelltextes**
- M 2 **Analyse: Bestimmung der Anforderung für die Klasse *CardioTrainer***
- Benötigt:** *CardioTrainer.java*, *CardioTrainer_kommentiert.java* (optional)

Erarbeitung

- M 3 **Analyse und Implementierung: Auftrag zur Erweiterung der Klasse *CardioTrainer***
- M 4 **Planung und Umsetzung der *User Story***
- M 5 **Infomaterial: Zählergesteuerte Schleifen in *Java***
- Benötigt:** *Erklärvideo zur for-Schleife in Java* https://raabe.click/Java-Tutorial_for-Schleife

Ergebnissicherung

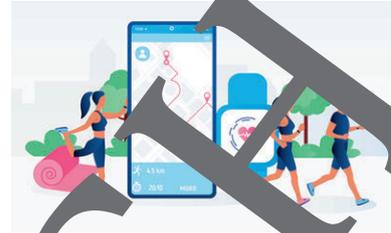
- M 6 **Arbeitsauftrag Code Review**
- M 7 **Feedbackbogen Code Review**
- M 8 **Zusammenfassung *for*-Schleife in *Java* – Ergebnissicherung**
- Benötigt:** *LearningSnack* (optional): <https://raabe.click/LearningSnack-for-Schleife>

M 1

Ein Update für die Trainings-App – Analyse des vorhandenen Quelltextes

Situationsbeschreibung

Apps zur Unterstützung des eigenen Trainings erfreuen sich momentan großer Beliebtheit. Lukas ist Auszubildender zum Fachinformatiker Anwendungsentwicklung und Teil des Entwicklerteams der App *CardioTrainer Ultimate*. Um weitere User zu gewinnen, wird die App regelmäßig um neue Funktionen ergänzt. Lukas bespricht mit seiner Ausbilderin, Frau Bruski, die anstehenden Aufgaben.



© Mdzhan Nagiev/iStock/Getty Images Plus

Frau Bruski: „Wie du weißt, können unsere Nutzer ihre Trainingsintensität mithilfe der Herzfrequenz überwachen.“

Lukas: „Ich erinnere mich. Die Berechnung der Herzfrequenzen war ziemlich kompliziert. Ist der Teil der App schon fertig?“

Frau Bruski: „Die Methode dafür ist zum Glück schon getestet. Allerdings wünschen sich viele Nutzer eine übersichtliche Tabelle mit unterschiedlichen Intensitätsstufen.“

Lukas: „Soweit ich weiß, wurde dafür schon etwas implementiert.“

Frau Bruski: „Okay. Dann schau dir nochmal den Quelltext dazu an und analysiere, welche Funktionen bereits umgesetzt sind. Danach schauen wir uns die *Story* zusammen an“

Aufgabe

Führe eine Analyse der Einstiegssituation durch, indem du die folgenden Fragen beantwortest.

Welche Aufgabe hat Lukas?

Welche Schritte sollte er zur Bearbeitung der Aufgabe durchführen?

Wie kann er sein Ergebnis dokumentieren?

Planung und Umsetzung der *User Story*

M 4

Aufgabe 1: Leitfragen zur Planung

Beantworte die folgenden Leitfragen, um die Umsetzung der *User Story* zu planen.

Welche Schritte müssen zur Umsetzung des Arbeitsauftrags M 3 durchgeführt werden?

Welche Technik muss zur Implementierung verwendet werden?

Wie wird diese Technik in Java umgesetzt? Recherchiere dazu ggf. kurz im Internet

An welcher Stelle im Programm müssen die Erweiterungen vorgenommen werden?

Was muss bei der Implementierung noch beachtet werden?

Aufgabe 2: Umsetzung der *User Story*

Implementiert eine der beiden *User Stories* 1 oder 2 aus M 3 entsprechend der obigen Leitfragen zur Planung.

Pair Programming

Bei dieser Methode wird zu zweit an einem Rechner gearbeitet. Ein Teammitglied erstellt den Quelltext und erläutert dem anderen Teammitglied seine bzw. ihre Überlegungen und Vorgehensweise. Das andere Teammitglied hört dabei aktiv zu und stellt Rückfragen. Nach einer festgelegten Zeitspanne tauschen die Teammitglieder ihre Rollen.

Bei Bedarf könnt ihr das Infomaterial M 5 als Hilfe verwenden.



Sie wollen mehr für Ihr Fach?

Bekommen Sie: Ganz einfach zum Download im RAABE Webshop.



Über 5.000 Unterrichtseinheiten
sofort zum Download verfügbar



Webinare und Videos
für Ihre fachliche und
persönliche Weiterbildung



Attraktive Vergünstigungen
für Referendar:innen
mit bis zu 15% Rabatt



Käuferschutz
mit Trusted Shops



Jetzt entdecken:
www.raabe.de