Lernweg Rekursion

Verwendung: Du kannst die Informationen zu den Lernwegabschnitten in dein Notizprogramm übernehmen oder ausdrucken und in dein Heft kleben, so dass nach jedem Schritt Raum zur Selbstreflexion und für eigene Notizen bleibt.

1: Annäherung ans Thema

Inhalte: Was ist Rekursion?

- Einstieg: Schachtelsuche
- Zwei Möglichkeiten, den Schlüssel zu finden, die rekursive Variante ist hier beschrieben

Übungen/Aufgaben:

- Programmieraufgabe "Countdown"

Kontrollfragen:

- Erläutere, warum man bei rekursiven Fuktionsaufrufen stets eine Fallunterscheidung benötigt.

Checkliste:

- Erledigt:
- Selbsteinschätzung:

2: Der Call-Stack

Inhalte: Der Programmaufrufstack

- Selbsterarbeitung: Der Programmaufrufstack bei Funktionsaufrufen
- Selbsterarbeitung: Der Programmaufrufstack bei rekursiven Funtionsaufrufen

Übungen/Aufgaben:

- Aufgaben auf beiden Wiki-Seiten.
- Erstelle eine kleine Präsentation, mit der du deinen Mitschülern den Call-Stack erklären kannst. Du kannst ein eigenes Beispiel überlegen oder die gegebenen Beispiele verwenden.

Kontrollfragen:

- Kann man sich mit dem Call-Stack die Sichtbarkeit von Variablen (lokal/global) erklären?
- Erstelle das Grundgerüst einer rekursiven Funktion.
- Was passiert bei einem "Stack Overfow Error?" - wann tritt dieser auf?

Checkliste:

- Erledigt:
- Selbsteinschätzung:

3: Anwendung und Übung

Methode

Pair Programming in Zufallsteams, Wechsel alle ~5 Minuten

Übungen/Aufgaben:

- Du weißt, was beim Pair-Programming zu tun ist - anderfalls liest du es nach.
- Es gibt drei Seiten mit Aufgaben:
  - Textuelle Knobeleien, die sich für eine rekursive Lösung anbieten. Zu jeder Aufgabe findet ihr Hinweise und Lösungvorschläge, die zu spicken dienen können, wenn ihr festhängt.
  - Grafische Knobeleien. Hier kommt eine Bibliothek zum Einsatz, mit Hilfe derer man eine "Schildkröte" laufen lassen kann, die dann eine Zeichenspur hinterlässt. Ihr müsst zunächst ausprobieren, wie man mit dieser Turtle-Grafik zeichnet. Anschließend könnt ihr die einzelnen Aufgaben bearbeiten, der Phytagorasbaum ist als Bonusaufgabe gedacht.
  - Mehrfache Selbstaufrufe. Hier finden sich zwei Beispiele, bei denen sich die Funktion in jedem Schritt mehrfach selbst aufruft. Wenn man eine solche Aufrufkaskade veranschaulichen möchte, muss man einen Baum zeichnen, wie das geht ist dort erklärt. Die dynamisch-rekursive Variante ist als Bonus für starke Schülerinnen gedacht.

Die Türme von Hanoi sollten als Hausaufgabe programmiert werden.

Kontrollfragen:

- Kannst ein allgemeines Vorgehen formulieren, wie du vorgehst, um ein Problem rekursiv zu lösen?

Checkliste:

- Erledigt:
- Selbsteinschätzung:

4: Backtracking

Inhalte: Das Funktionsprinzip bei "Backtracking"

- Input: Lehrervortrag
- Gemeinsame Besprechung und Programmierung des 8 Damen-Problems
- Lösung des 8 Damen-Problems in Pair-Programming nachvollziehen
- Eines der verbleibenden Beispiele (Magisches Quadrat/Sudoku) lösen

Kontrollfragen:

- Kannst du das allgemeine Vorgehen beim Backtracking erläutern? Worin besteht die Stärke der Methode?

Checkliste:

- Erledigt:
- Selbsteinschätzung: