

InClass-Exercise 2

EProg25 - Repetierenden Gruppen

October 2025

Dies ist kein offizielles EProg-Dokument und wurde daher nicht vom EProg-Team auf Korrektheit überprüft. Es dient dazu, in der Übungsstunde Rekursion zu üben.

Kurze Aufgaben

Gegeben ist die Klasse `RecTasks`, die mehrere Methoden zur Übung rekursiven Denkens enthält. Alle Methoden sollen **rekursiv** implementiert werden. Schleifen sind nicht erlaubt. Wo nötig, dürfen Sie eine *Hilfsmethode* definieren, um Zwischenzustände zu verfolgen (z.B. den aktuellen Index oder Teilergebnisse). Verwenden Sie bei String-Verarbeitung konsequent **substring** anstelle von Indexarithmetik.

Sie dürfen annehmen, dass keine Eingabe `null` ist. Alle anderen Eigenschaften (z.B. leere Strings oder Arrays) müssen geprüft werden. Eine Methode ist nur korrekt, wenn alle beschriebenen Bedingungen erfüllt sind.

Aufgabe 1 — Summe der Elemente

Schreiben Sie eine Methode

```
int sum(int[] xs)
```

die die Summe aller Elemente eines Arrays berechnet.

Beispiele: `[]` $\rightarrow 0$

`[7]` $\rightarrow 7$

`[2, 5, 1]` $\rightarrow 8$

Hinweis: Verwenden Sie eine Hilfsmethode, die den aktuellen Index mitführt. Das erlaubt, ohne Schleife schrittweise die Summe aufzubauen.

Aufgabe 2 — Zeichen zählen

Implementieren Sie

```
int count(char c, String s)
```

welche zählt, wie oft das Zeichen `c` im String `s` vorkommt. Nutzen Sie `s.substring(1)` zur Rekursion über den Rest des Strings.

Beispiele:

`count('a', "banane") → 2`

`count('x', "") → 0`

Aufgabe 3 — Sortiertes Array

Implementieren Sie

```
boolean isSorted(int[] xs)
```

welche überprüft, ob das Array `xs` in **nicht abnehmender Reihenfolge** sortiert ist.

Beispiele:

`[] → true`

`[5] → true`

`[1,2,2,5] → true`

`[2,1] → false`

Hinweis: Verwenden Sie eine Hilfsmethode mit einem Indexparameter, um das nächste Paar von Elementen zu vergleichen.

Aufgabe 4 — Wiederholungen entfernen

Schreiben Sie eine Methode

```
String dedupRuns(String s)
```

die aufeinanderfolgende gleiche Zeichen auf ein einziges Zeichen reduziert.

Beispiele:

`"aabbcca" → "abca"`

`"mississippi" → "misisipi"`

Hinweis: Verwenden Sie eine Hilfsmethode, die das letzte Zeichen speichert, um Wiederholungen zu erkennen. Arbeiten Sie mit `substring`, nicht mit Indizes.

Aufgabe 5 — Klammern prüfen (balanciert?)

Schreiben Sie eine Methode

```
boolean isBalanced(String s)
```

die prüft, ob der String aus geschweiften `{ }` Klammern korrekt balanciert ist.

Beachten Sie:

- Jede geschweifte Klammer `{ }` muss geschlossen sein.
- Andere Klammern `[ ( , < , > ]` dürfen nicht auftreten.

- Andere Zeichen dürfen ignoriert werden.

Beispiele:

`"{}"` → `true`

`"{{}}"` → `true`

`"({a})"` → `false`

`"{a{} }"` → `true`

Hinweis: Verwenden Sie eine Hilfsmethode.