

Der ADT "Set" (Menge)

Definition: Set

Der abstrakte Datentyp Set repräsentiert im mathematischen Sinne eine Menge. Eine Menge hat folgende Eigenschaften:

- Eine Menge kann beliebig viele Elemente enthalten.
- Jedes Element kann höchstens einmal vorhanden sein.
- Es kommt nicht auf die Reihenfolge der Elemente an.

Und folgende Methoden:

- Konstruktor `Set()` - erzeugt eine leere Menge
- `einfuegen(wert: int)` - fügt den Wert `wert` in die Menge ein, falls er noch nicht vorhanden ist
- `entfernen(wert: int)` - entfernt den Wert `wert` aus der Menge, falls er vorhanden ist; andernfalls wird die Menge nicht verändert.
- `enthaelt(wert: int): boolean` - gibt `true` zurück, wenn `wert` in der Menge enthalten ist, sonst `false`.
- `anzahl(): int` - gibt die Anzahl der Elemente in der Menge zurück
- `istLeer(): boolean` - gibt `true` zurück, wenn die Menge leer ist, sonst `false`.
- `schnittmenge(s: Set): Set - ???`
- `vereinigungsmenge(s: Set): Set - ???`
- `differenz(s: Set): Set - ???`
- `untermenge(s: Set): boolean` - gibt `true` zurück, wenn jedes Element dieser Menge in `s` enthalten ist
- `gleich(s: Set): boolean` - gibt `true` zurück, wenn diese Menge und `s` die gleichen Elemente enthält; sonst `false`.

Erarbeitung

Arbeite mit der Vorlage von **LINK** und bearbeite nachfolgende Aufgaben.



(A1) Funktionalität der Varianten

In der Vorlage sind zwei verschiedene Varianten implementiert. Der Quellcode ist nicht einsehbar. Finde heraus, ob beide Versionen die gleiche Funktionalität aufweisen. Erstelle dazu Beispielobjekte.



(A2) Operationen mit Mengen

Wähle nun eine der beiden Varianten aus der Vorlage aus und löse damit die folgenden Teilaufgaben.

1. Erstelle ein Set $M = \{19, 23, 1, 11, 10, 33, 9, 42, 17\}$ und ein Set $N = \{10, 7, 11, 19, 2, 23, 42, 37\}$. Finde heraus was die Methoden `schnittmenge(s)`, `vereinigungsmenge(s)` und `differenz(s)` machen. Beschreibe die Funktionalität für das gegebene Beispiel sowie den allgemeinen Fall (schriftlich).
2. Die Operation `untermenge(s)` gibt `true` zurück, wenn jedes Element dieser Menge in `s` enthalten ist. Überprüfe diese Funktionalität, indem du die Operation mit verschiedenen Beispielmengen testest. Achte darauf, dass alle möglichen Fälle abgedeckt sind.
3. Vergleiche die Operationen des ADT Sets mit anderen Datenstrukturen (Liste, Array, ...). Welche Gemeinsamkeiten und v.a. welche Unterschiede fallen dir auf?
4. Ist die Reihenfolge der eingefügten Elemente relevant? Prüfe mithilfe passender Beispielmengen.



(A3) Verschiedene Varianten schneiden

1. Erstelle je ein Set beider Varianten. Schneide diese beiden Sets miteinander. Beschreibe was passiert. Was passiert, wenn du sie anders herum schneidest?
2. Warum ist das so? Begründe deine Beobachtung.

[n/a: Keine Treffer]

Hier findest du Tipps zu den einzelnen Aufgaben, sofern du sie benötigst.

Tipp A2.1

Mengen können auch in dieser Form dargestellt werden. Übertrage eine Skizze in dein Heft und ordne die Elemente der Mengen M und N den entsprechenden Stellen zu. Welche Abbildung passt zu welcher Operation?



Tipp1 A2.2

Wann gibt `untermenge(s)` `true` und wann `false` zurück?

Tipp2 A2.2

Was passiert wenn du statt `Set1. untermenge(Set2)` die beiden Sets miteinander vertauscht?

Tipp3 A2.2

Was passiert wenn Set1 und Set2 identisch sind?

Tipp A2.3

Was passiert bei einfuegen(wert)?

Tipp A2.4

Was passiert bei gleich(s)?

Tipp A3.1

In welcher Variante liegt das Ergebnis vor?

Tipp A3.2

Abstraktion

From:
<https://wiki.qg-moessingen.de/> - **QG Wiki**

Permanent link:
<https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:adt:set:start?rev=1636563697>

Last update: **10.11.2021 18:01**

