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` – gibt eine Menge zurück, die genau die Elemente enthält, die in dieser Menge und in `s` enthalten sind.
- `vereinigungsmenge(s: Set): Set` – gibt eine Menge zurück, die genau die Element enthält, die in dieser Menge oder in `s` oder in beiden enthalten sind.
- `differenz(s: Set): Set` – gibt eine Menge zurück, die genau die Elemente enthält, die in dieser Menge, aber nicht in `s` enthalten sind.
- `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 Versionen

In der Vorlage sind zwei verschiedene Versionen zu finden. 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 und nutze sie für diese Aufgabe.

- 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).
- 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.

Tipp1

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

Tipp2

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

Tipp3

Was passiert wenn `Set1` und `Set2` identisch sind?

- Vergleiche die Operationen des ADT Sets mit anderen Datenstrukturen (Liste, Array, ...). Welche Gemeinsamkeiten und v.a. welche Unterschiede fallen dir auf?

Tipp

Was passiert bei `einfuegen(wert)`?

- Ist die Reihenfolge der eingefügten Elemente relevant? Prüfe mithilfe passender Beispielmengen.

Tipp

Was passiert bei `gleich(s)`?

(A3) Verschiedene Varianten schneiden

- Erstelle je ein Set beider Varianten. Schneide diese beiden Sets miteinander. Beschreibe was passiert. Was passiert, wenn du sie anders herum schneidest?

Tipp

In welcher Variante liegt das Ergebnis vor?

- Warum ist das so? Begründe deine Beobachtung.

Tipp

Abstraktion

adt_set1.odp	45.2 KiB	30.11.2021 15:28
schnittdifferenzvereinigung.png	535.6 KiB	05.11.2021 10:24
schnittvereinigungdifferenz.drawio.png	568.7 KiB	05.11.2021 10:17
set01.odp	402.8 KiB	18.10.2021 18:27
set01.pdf	209.5 KiB	18.10.2021 18:27

From:
https://info-bw.de/ -

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:adt:set:start?rev=1635937376

Last update: **03.11.2021 11:02**