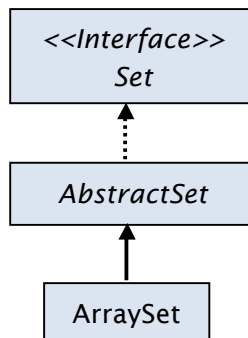


Aufgabenblatt 1

Es soll ein Datentyp erstellt werden, der das Arbeiten mit int-Mengen wie z.B. {5,7,1}, {2}, {}, {3,5,0} etc. unterstützt.

Auf der Web-Seite finden Sie das Interface Set mit einer Javadoc-Beschreibung. Mit der Klasse `ArraySet`, die das Interface `Set` implementiert, sollen Mengen als Felder realisiert werden. Dabei wird eine abstrakte Klasse `AbstractSet` als Zwischenschritt eingebaut.

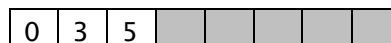


Klasse `AbstractSet`:

Realisieren Sie diese Klasse, indem Sie die Methode `containsAll` mit Hilfe der Methoden `get` und `contains` implementieren. Ebenso soll `addAll` bzw. `removeAll` mit Hilfe von `get` und `add` bzw. `remove` und `clear` implementiert werden. Beachten Sie, dass auch Aufrufe der Form `s.addAll(s)` und `s.removeAll(s)` fehlerfrei funktionieren.

Klasse `ArraySet`:

Realisieren Sie eine Menge als Feld, in dem die Elemente sortiert, linksbündig und lückenlos abgespeichert werden. Z.B. wird die Menge {0, 3, 5} im Feld wie folgt abgespeichert:



Hinweise:

Das Feld im Beispiel hat zwar die Größe 8, ist aber nur mit 3 Elementen belegt. Daher ist es notwendig, dass das Mengenobjekt noch zusätzlich die Anzahl der Elemente speichert. Ist das Feld gefüllt, dann muss bei einem weiteren Dazufügen eines Elements, das Feld vergrößert werden (siehe hierzu die Methode `ensureCapacity` in der Klasse `ArrayList` in der Vorlesung).

Beim Einfügen eines Elements muss zunächst die richtige Position bestimmt werden und dann das davon rechts liegende Teilfeld mit Hilfe von `arraycopy` um eine Position nach rechts verrückt werden.

Der Konstruktor legt eine leere Menge an.

Die Klasse `ArraySet` überschreibt außerdem die Methode `toString`.

Anwendungsklasse `SetApplication`

Zu Testzwecken ist eine Anwendungsklasse mit einer `main`-Methode auf der Web-Seite vorgegeben.

Abgabe:

Die `main`-Methode der Anwendungsklasse muss fehlerfrei laufen.
Der erstellte Code muss genau erklärt werden können.