

Der bestehende Iterator Ihres **ADS\_set** ist um einen zusätzlichen „Modus“ zu erweitern. In der bisherigen Implementierung liefert der Iterator alle  $n$  Elemente des **ADS\_set** in einer beliebigen Reihenfolge, wobei die Reihenfolge immer dieselbe sein muss, solange das **ADS\_set** nicht geändert wird (Modus „normal“). Im Modus „speziell“ liefert der Iterator die Elemente in derselben Reihenfolge, es wird aber nur jedes dritte Element geliefert, die übrigen Elemente werden übersprungen. Der Iterator im Modus „speziell“ liefert somit die  $\left\lfloor \frac{n}{3} \right\rfloor$  Elemente an den Positionen 3, 6, 9, ... usw., allgemein alle Elemente an den Positionen  $3i$  mit  $i = 1.. \left\lfloor \frac{n}{3} \right\rfloor$ . In beiden Modi erreicht der Iterator nach dem letzten gelieferten Element **end()**.

**Details:** Erweitern Sie Ihre Implementierung **ADS\_set** um die Methode

**const\_iterator w() const;**

**w()** erzeugt einen Iterator im Modus „speziell“. Wenn keine zu liefernden Elemente im **ADS\_set** vorhanden sind, dann gilt **w() == end()**.

Die Zeitkomplexität und Speicherkomplexität der Operatorfunktionen müssen unverändert bleiben. So sind z. B. zusätzliche Felder mit nicht konstanter Größe unzulässig.

Beispiele:

Angenommen der von <b>begin()</b> retournierte Iterator liefert alle $n$ gespeicherten Elemente in der Reihenfolge	Dann liefert der von <b>w()</b> retournierte Iterator die folgenden Elemente in der folgenden Reihenfolge
(1,2,3,4,5,6,7)	(3,6)
(1,2,3,4,5,6)	(3,6)
(1,2,3,4,5)	(3)
(4,3,6,1)	(6)
(9,7,8)	(8)
(7,9)	()
(7)	()
()	()

**Anleitung:** Schreiben Sie **keine** neue Iteratorklasse! Erweitern Sie die bestehende Iterator-Klasse wie folgt (dies ist nur einer der möglichen Lösungsansätze, abweichende korrekte Lösungen sind natürlich zulässig):

- Es muss ein Iterator im Modus „speziell“ erzeugt werden können. Dazu ist ein neuer Konstruktor zu schreiben und/oder bestehende zu erweitern, um die Instanzvariablen entsprechend zu initialisieren. Eventuell benötigen Sie zusätzliche Konstruktorparameter und/oder Instanzvariablen. Es kann hilfreich sein, wenn im Iterator bekannt ist, wie viele Elemente im **ADS\_set** vorhanden sind und an welcher Position sich der Iterator gerade befindet.
- Passen Sie die Inkrement-Operationen (nur!) für den Modus „speziell“ an: nur jedes dritte Element wird geliefert, die übrigen werden übersprungen. Nach dem letzten gelieferten Element wird der Iterator (wie immer) auf **end()** gesetzt.

Die Methode **ADS\_set::w()** erzeugt einen Iterator im Modus „speziell“ und retourniert diesen.

Die Methode **ADS\_set::begin()** liefert wie bisher einen Iterator im Modus „normal“.