

Der bestehende Iterator Ihres **ADS_set** ist um einen zusätzlichen „Modus“ zu erweitern. In der bisherigen Implementierung liefert der Iterator alle Elemente 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, bricht aber nach einer bestimmten Anzahl von Elementen ab. Diese Anzahl wird bei der Erzeugung des Iterators festgelegt. In beiden Modi erreicht der Iterator nach dem letzten gelieferten Element **end()**.

Details: Erweitern Sie Ihre Implementierung **ADS_set** um die Methode

```
const_iterator z(size_t m) const;
```

Diese soll einen Iterator im Modus „speziell“ erzeugen, der genau **m** Werte liefert. Wenn kein Element im **ADS_set** vorhanden ist, oder **m == 0**, dann gilt **z(m) == end()**. Andernfalls liefert **z(m)** einen Iterator, der auf das 1. Element zeigt. Der von **z(m)** gelieferte Iterator soll in weiterer Folge die nächsten **m-1** Elemente liefern, danach wird **end()** geliefert. Wenn **m** größer ist als die Anzahl der Elemente im **ADS_set**, dann wird der Iterator nach dem letzten Element im **ADS_set** wieder auf das 1. Element gesetzt usw.

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

Angenommen der von begin() retournierte Iterator liefert alle gespeicherten Elemente in der Reihenfolge	Dann liefert der von			
	z(0)	z(1)	z(3)	z(7)
	retournierte Iterator die folgenden Elemente in der Reihenfolge			
(7,4,1,5,3,6,0,8,10,2,9)	()	(7)	(7,4,1)	(7,4,1,5,3,6,0)
(7,8,9,4,1,5)	()	(7)	(7,8,9)	(7,8,9,4,1,5,7)
(7,8,9)	()	(7)	(7,8,9)	(7,8,9,7,8,9,7)
(7,9)	()	(7)	(7,9,7)	(7,9,7,9,7,9,7)
(7)	()	(7)	(7,7,7)	(7,7,7,7,7,7,7)
()	()	()	()	()

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 Instanzvariablen, um den Anfangszustand zu speichern.
- Passen Sie die Inkrement-Operationen (nur!) für den Modus „speziell“ an: nach dem **m**. Element wird der Iterator auf **end()** gesetzt bzw. bis **end()** weitergeschaltet. Sollte vor dem **m**. Element **end()** erreicht werden, dann wird der Iterator wieder in den Anfangszustand (1. Element) gesetzt.

Die Methode **ADS_set::z(size_t)** erzeugt einen Iterator im Modus „speziell“ und retourniert diesen. Die Methode **ADS_set::begin()** liefert wie bisher einen Iterator im Modus „normal“.