1. (2 Punkte) Liste ist unser selbstgebastelter Datentyp. Wir nummerieren die Einträge der Liste mit Zeiger `anf` an der Stelle 0.

   a. Welche Zahlenfolge findet sich in den Listenlementen?
   b. Gib den Index des Elements an, auf das der Zeiger `pos` zeigt.

```
a = Liste()
a.insert(1)
a.insert(2)
a.advance()
a.insert(3)
a.insert(4)
a.reset()
a.advance()
a.insert(6)
```

   Lösung: a. `2 6 4 3 1` b. `1`

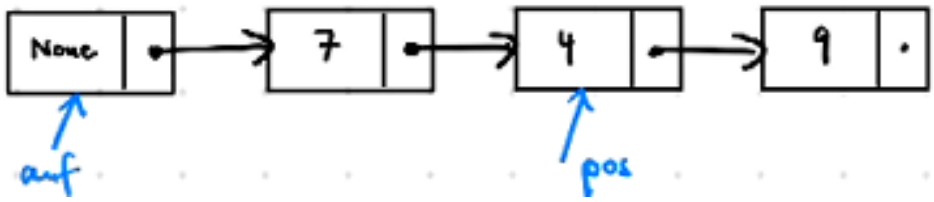
2. (2 Punkte) Liste ist unser selbstgebastelter Datentyp. Wir nummerieren die Einträge der Liste mit Zeiger `anf` an der Stelle 0.

   a. Welche Zahlenfolge findet sich in den Listenlementen?
   b. Gib den Index des Elements an, auf das der Zeiger `pos` zeigt.

```
a = Liste()
a.insert(1)
a.insert(2)
a.insert(3)
a.insert(4)
```

   Lösung: a. `4 3 2 1` b. `0`

3. (5 Punkte) Liste ist unser selbstgebastelter Datentyp. Das Bild zeigt die Situation der Liste a.



   a. Was liefern die folgenden Anweisungen:

```
print(a.elem())
print(a.endpos())
```

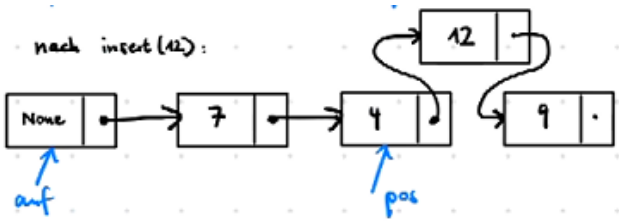
   b. Zeichne die Situation nach der Ausführung von a.insert(12)
   c. Was liefern jetzt die folgenden Anweisungen:

```
print(a.elem())
print(a.endpos())
```

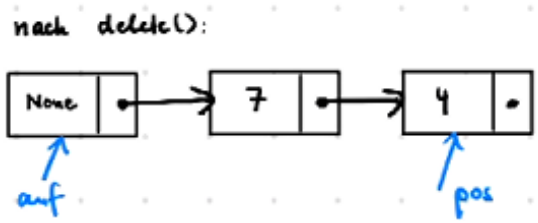
   d. Wir gehen nochmal von der Anfangssituation aus. Zeichne die Situation nach Ausführung von a.delete()
   e. Was liefern jetzt die folgenden Anweisungen:

```
print(a.elem())
print(a.endpos())
```

Lösung: a. 9, False
b.

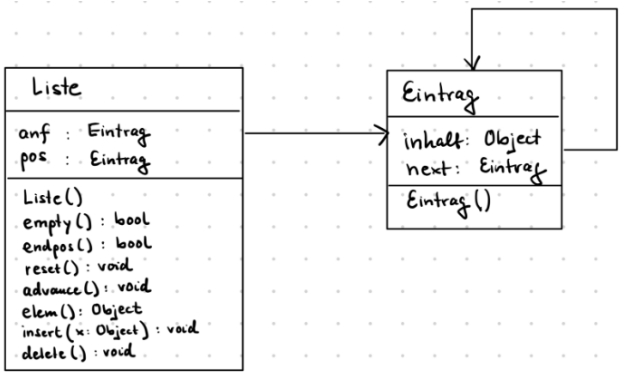


c. 12, False
d.



e. Fehler, True

4. (4 Punkte) Das Bild zeigt das UML-Diagramm unseres Datentyps Liste. Implementiere in Python die Klassen Eintrag und die Klasse Liste (nur Konstruktor und Methode delete). Achte dabei auf mögliche Fehlersituationen.



Lösung:

```
class Eintrag:
    def __init__(self):
        self.inhalt = None
        self.next = None

class Liste:
    def __init__(self):
        self.anf = Eintrag()
        self.pos = self.anf

    def delete(self):
        if self.endpos(): raise RuntimeError("Fehler: Liste am Ende")
        self.pos.next = self.pos.next.next
```