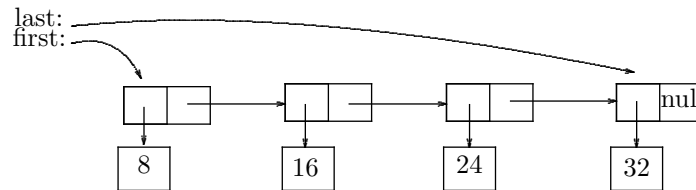


Algorithmen und Datenstrukturen

Übung 7 (AIN2)



1. Implementieren Sie eine Klasse **MyList**, die eine verkettete Liste von **Integer** Objekten darstellt. Dazu sollen Sie **keine Collections** verwenden, sondern sich z.B. an der Klasse **Stack** aus dem Skript (Seite 27) orientieren.

Ihre Klasse soll folgende Methoden bereitstellen:

`public MyList()` erzeugt eine leere Liste.

`public void add(Integer x)` hängt ein neues Element **x** hinten an die Liste an. Diese Operation soll unabhängig von der Länge der Liste immer gleich schnell arbeiten. ($T_{\text{add}} = O(1)$)

`public int size()` liefert die Länge der Liste

`public Integer get(int index)` liefert das Listenelement an der Position $\text{index} \geq 0$.

`public Integer remove((int index)` liefert das Listenelement an der Position $\text{index} \geq 0$ und entfernt es aus der Liste.

2. Testen Sie Ihre Implementierung mit folgendem Beispielprogramm

```

public static void main(String[] args) {
    MyList l= new MyList();
    for (int i=1; i<=10; ++i)
        l.add(8*i);
    l.remove(3);
    for (int i=l.size()-1; i>=0; --i)
        System.out.print(" " + l.get(i));
    System.out.println();
}

```

3. Erweitern Sie Ihre Klasse um einen Typparameter **T**, so dass man statt **Integer**-Objekten auch Objekte einer beliebigen Klasse **T** verwenden kann.
4. Testen Sie die neue typsichere Klasse mit irgendeinem anderen im Praktikum verwendeten Datentyp. (Z.B. **Zahlungsmittel** aus Übung 6 oder **Bruch** aus Übung 2)