

# Algorithmen und Datenstrukturen in Go

## Aufgabe: Verkettete Liste

Entwickle einen dynamischen Datentyp `LinkedList` in Go, bei dem es sich um eine einfache verkettete Liste handeln soll. Der Datentyp soll als **Stack**, **Liste** oder **Warteschlange** verwendet werden können und die folgenden Interfaces implementieren.

```
type Stack[T any] interface {
    // Push legt ein Element auf den Stack
    Push(value T)
    // Pop entfernt das oberste Element vom Stack
    Pop() T
}

type Queue[T any] interface {
    // Enqueue fügt ein Element am Ende der Schlange hinzu
    Enqueue(value T)
    // Deque entfernt das erste Element vom Anfang der Schlange
    Deque() T
}

type List[T any] interface {
    Stack[T]
    // Delete entfernt das Element bei "index"
    Delete(index int)
    // Get gibt das Element bei "index" zurück
    Get(index int) T
    // Insert fügt ein Element bei "index" ein
    Insert(index int, value T)
    // Len gibt die Anzahl der Elemente zurück
    Len() int
}

// aus "fmt" Package
type Stringer interface {
    String() string
}
```