Aufgabe F1: Doppelt verkettete Liste

Im Lerntext haben wir einfach verkette Listen vorgestellt und dort bereits kurz erwähnt, dass es auch doppelt verkette Listen gibt.

In einer doppelt verketten Liste hat jedes Listenelement nicht nur einen Zeiger auf sein Nachfolgeelement, sondern zusätzlich auch einen Zeiger auf sein *Vorgängerelement*.

In dieser Aufgabe stellen wir Ihnen in Form des Verbund-Typs `doppeltVerketteteListe` die grundlegende Datenstruktur einer doppelt verketten Liste zu Verfügung, deren Elemente jeweils einen Integer-Wert repräsentieren. Mittels der Felder `anfang` und `ende` kann direkt auf den Anfang der Liste (also den Kopf) und auf das Ende (also das letzte Element) zugegriffen werden.

Die beiden Funktionen `listeAusgeben` und `listeAusgebenRückwärts` zur Vorwärts- bzw. Rückwärts-Ausgabe einer Liste sind bereits fertig implementiert.

Bitte beachten Sie, dass es sich bei der hier implementierten Datenstruktur *nicht* um eine sortierte Liste handelt, d.h. die Elemente der Liste sind *nicht* nach ihrem Integer-Wert geordnet. Somit ergibt sich die Reihenfolge der Elemente also prinzipiell aus der Reihenfolge ihres Einfügens in die Liste.

Sie sollen nun die folgenden **acht Funktionen implementieren**, welche u.a. das Einfügen, Entfernen oder Zurückgeben eines Elements bzw. Werts an einer bestimmten Position in der Liste realisieren:

```
func einfügenAmAnfang(liste *doppeltVerketteteListe, neuerWert int) {...}
func einfügenAmEnde(liste *doppeltVerketteteListe, neuerWert int) {...}
func einfügenAnPosition(liste *doppeltVerketteteListe, neuerWert int, pos
 ↪  int) {...}
func gibElementAnPosition(liste *doppeltVerketteteListe, pos int)
 ↪  *listenElement {...}
func entferneAnfang(liste *doppeltVerketteteListe) {...}
func entferneEnde(liste *doppeltVerketteteListe) {...}
func entferneAnPosition(liste *doppeltVerketteteListe, pos int) {...}
func gibPositionVonWert(liste *doppeltVerketteteListe, gesuchterWert int)
 ↪  int {...}
```

Um die Implementierung einiger dieser Funktionen möglichst effizient zu gestalten, sollten Sie ggf. von der Möglichkeit Gebrauch machen, die Liste von vorne oder von hinten durchlaufen zu können.

Drei weitere Funktionen, nämlich `gibWertAnPosition`, `gibElementMitWert` und `istWertInListe`, die direkt auf den obigen Funktionen aufbauen, sind bereits implementiert.

Ressourcen

Im Ordner dieser Aufgabe finden Sie eine Datei `doppeltVerketteteListe.go`, die Sie entsprechend der Aufgabenstellung abändern sollen.

Weiterhin stellen wir in der Datei `main.go` eine main-Funktion zur Verfügung, damit Sie Ihre Funktion in einem beispielhaften Kontext kompilieren und ausführen können. Nachdem Sie in den Aufgabenordner gewechselt sind, geben Sie dazu folgenden Befehl ein:

```
go run .
```

Die Datei `doppeltVerketteteListe_test.go` stellt Tests bereit, die Sie mit folgendem Befehl durchführen können:

```
go test
```

Im Unterordner ML finden Sie einen Lösungsvorschlag.