В своем репозитории:

1. git clone %repository\_name%.git (если еще не сделано)
2. Перейти на master (git checkout master)
3. Создать ветку (git checkout -b collection)
4. Создать в репозитории папку для задания
5. После выполнения (относительно репозитория, не внутренней папки) (git add .) => (git commit -m “message”) => (git push -u origin collection)
6. Создать пул реквест
7. Отправить ссылку на ваш PR

**Реализовать коллекцию, которая будет содержать в себе элементы определенного типа(для примеров выбран тип int)**

Требования:

1. Создать отдельный package для коллекции
2. Использовать массивы и слайсы запрещено
3. Коллекция должна быть структурой с методами описанными ниже

* **Add(element int)** – добавляет элемент в коллекцию
* **Get(index int)** – возвращает элемент с данным индексом из коллекции (первый элемент имеет index = 0) (пример сигнатуры метода:

**func (collection \*Collection) Get(index int) \*Element**)

* **Remove(index int)** – удаляет элемент с данным индексом из коллекции
* **First()** – возвращает первый элемент в коллекции.

пример сигнатуры метода: **func (collection \*Collection) First() \*Element**

* **Last()** – возвращает последний элемент в коллекции.
  + (пример сигнатуры метода:

**func (collection \*Collection) Last() \*Element**  
)

* **Length()** – возвращает размер коллекции (текущее кол-во элементов)
* **Print()** – выводит все элементы коллекции в консоль

1. Реализовать структуру узла коллекции.

Методы:

**Next()** – возвращает следующий элемент коллекции

**Prev()** – возвращает предыдущий элемент коллекции

**Value()** – возвращает значение элемента

В main.go:

1. заполнить коллекцию произвольными значениями
2. вызвать каждый метод коллекции (после Add и Remove вызывать Print)

Пример main.go:

func main() {

collection := list.Collection{}

for i := 0; i < 10; i++ {

collection.Add(i)

}

collection.Print()

collection.Remove(4)

collection.Print()

fmt.Println(collection.Get(3))

fmt.Println(collection.First())

fmt.Println(collection.Last())

fmt.Println(collection.Length())

}

