Raymundo Guzmán Mata A01284709

Para esta actividad, corrí los siguientes test cases para mis códigos de Stack, Queue y Dictionary.

Stack

Implemente los siguientes métodos:

- NewStack[T](): Constructor que crea una nueva pila vacía del tipo T
- Push(item int): Agrega un elemento al final de la pila
- Pop() int: Elimina y retorna el último elemento de la pila
- Peek() int: Retorna el último elemento sin eliminarlo
- IsEmpty() bool: Verifica si la pila está vacía
- Size() int: Retorna el número de elementos en la pila
- Print(): Muestra todos los elementos de la pila
- Clear(): Elimina todos los elementos de la pila

Prueba

```go
// Test Stack with integers
fmt.Println("=== Testing Stack Implementation with Integers ===")
stackInt := NewStack[int]()
stackInt.Push(1)
stackInt.Push(2)
stackInt.Push(3)
fmt.Print("Stack after pushing 1,2,3: ")
stackInt.Print()
fmt.Printf("Popped value: %d\n", stackInt.Pop())
fmt.Printf("Popped value: %d\n", stackInt.Pop())
fmt.Print("Stack after pop: ")
stackInt.Print()
fmt.Printf("Peek value: %d\n", stackInt.Peek())
fmt.Printf("Is stack empty?: %v\n", stackInt.IsEmpty())
fmt.Printf("Stack size: %d\n", stackInt.Size())
stackInt.Clear()
fmt.Printf("Is stack empty after clear?: %v\n", stackInt.IsEmpty())
```

```
=== Testing Stack Implementation with Integers ===
Stack after pushing 1,2,3: 1 2 3
Popped value: 3
Popped value: 2
Stack after pop: 1
Peek value: 1
Is stack empty?: false
Stack size: 1
Is stack empty after clear?: true
```

Raymundo Guzmán Mata A01284709

Queue

Implemente los siguientes métodos:

- NewQueue[T](): Constructor que crea una nueva cola vacía del tipo T
- Enqueue(item int): Agrega un elemento al final de la cola
- Dequeue() int: Elimina y retorna el primer elemento de la cola
- Front() int: Retorna el primer elemento sin eliminarlo
- IsEmpty() bool: Verifica si la cola está vacía
- Size() int: Retorna el número de elementos en la cola
- Print(): Muestra todos los elementos de la cola
- Clear(): Elimina todos los elementos de la cola

Prueba

```go
// Test Queue with integers
fmt.Println("\n=== Testing Queue Implementation with Integers ===")
queueInt := NewQueue[int]()
queueInt.Enqueue(1)
queueInt.Enqueue(2)
queueInt.Enqueue(3)
fmt.Print("Queue after adding 1,2,3: ")
queueInt.Print()
fmt.Printf("Dequeued value: %d\n", queueInt.Dequeue())
fmt.Printf("Dequeued value: %d\n", queueInt.Dequeue())
fmt.Print("Queue after dequeue: ")
queueInt.Print()
fmt.Printf("Front value: %d\n", queueInt.Front())
fmt.Printf("Is queue empty?: %v\n", queueInt.IsEmpty())
fmt.Printf("Queue size: %d\n", queueInt.Size())
queueInt.Clear()
fmt.Printf("Is queue empty?: %v\n", queueInt.IsEmpty())
```

```
=== Testing Queue Implementation with Integers ===
Queue after adding 1,2,3: 1 2 3
Dequeued value: 1
Dequeued value: 2
Queue after dequeue: 3
Front value: 3
Is queue empty?: false
Queue size: 1
Is queue empty?: true
```

Raymundo Guzmán Mata A01284709

Dictionary

Implemente los siguientes métodos:

- NewDictionary[K,V](): Crea un nuevo diccionario vacío con claves tipo K y valores tipo V
- Add(key string, value string): Agrega un par clave-valor al diccionario
- Remove(key string): Elimina un par clave-valor del diccionario
- Get(key string) string: Retorna el valor asociado a una clave
- Contains(key string) bool: Verifica si una clave existe en el diccionario
- Print(): Muestra todos los pares clave-valor del diccionario

Prueba

```go
// Test Dictionary with string keys and string values
fmt.Println("\n=== Testing Dictionary Implementation with String Keys and Values ===")
dictStrStr := NewDictionary[string, string]()
dictStrStr.Add("key1", "value1")
dictStrStr.Add("key2", "value2")
dictStrStr.Add("key3", "value3")
fmt.Print("Current dictionary contents:")
dictStrStr.Print()
dictStrStr.Remove("key2")
fmt.Print("Dictionary after removing key2:")
dictStrStr.Print()
fmt.Printf("Value for key1: %v\n", dictStrStr.Get("key1"))
fmt.Printf("key4 exists in dictionary: %v\n", dictStrStr.Contains("key4"))
```

```
=== Testing Dictionary Implementation with String Keys and Values ===
Current dictionary contents:{key1:value1, key2:value2, key3:value3}
Dictionary after removing key2:{key1:value1, key3:value3}
Value for key1: value1
key4 exists in dictionary: false
```

Git:
https://github.com/raxhacks/TC3002B/tree/main/Tarea1_A01284709