### **Algoritmo (passo a passo)**

Algoritmo makeRange(lower, upper):

criar vetor range com tamanho (upper - lower + 1)

para i de 0 até range.length - 1 faça:

range[i] ← lower

lower ← lower + 1

retorna range

Algoritmo principal:

declarar vetor range

criar objeto lister do tipo RangeLister

range ← lister.makeRange(1, 10)

imprimir "The array: [ "

para i de 0 até range.length - 1 faça:

imprimir range[i] + " "

imprimir "]"

**Teste de Mesa**

### **Criação de uma classe chamada RangeLister:** Essa classe é como uma caixinha onde guardamos um método que sabe criar uma lista de números em sequência, a partir de dois valores: o número inicial e o número final.

### **Método makeRange(lower, upper):** Esse método recebe dois números inteiros:

### Um número de início (lower)

### Um número de fim (upper)

### **Com esses dois números, ele cria uma lista de números inteiros, começando do número inicial e indo até o número final, incluindo os dois.** Internamente, o método faz o seguinte:

### Calcula o tamanho da lista que precisa criar. Por exemplo, se for do 1 até o 10, ele precisa de 10 espaços.

### Depois, ele preenche essa lista, colocando um número em cada posição. Começa com o número inicial, e vai somando 1 a cada vez, até preencher tudo.

### **Dentro da main (função principal):**

### O programa cria uma variável onde ele vai guardar essa lista de números.

### Depois, ele cria um objeto da classe RangeLister, como se fosse apertar um botão dizendo “ok, quero usar essa ferramenta que gera listas”.

### Em seguida, ele chama o método makeRange, pedindo uma lista de números de 1 até 10. O resultado é salvo na variável.

### Por fim, o programa imprime essa lista na tela. Ele passa por cada número da lista, um por um, e mostra esse número com um espaço ao lado, formando a sequência “1 2 3 4 5 6 7 8 9 10”.

### **O que aparece no final:** A frase “The array:[ 1 2 3 4 5 6 7 8 9 10 ]” aparece no console, mostrando que a lista foi criada certinho.

### 

### **Tipo de Algoritmo**

Esse é um algoritmo do tipo **TAD (Tipo Abstrato de Dado)**