



Instituto Tecnológico de Santo Domingo

Laboratorio de Tendencias en Desarrollo de Aplicaciones

Sección: INS 368L-02

Profesor: LORENZO SOLANO MARTINEZ

Plan de Pruebas

Jun Huo He Zhang 1088620

Santo Domingo, Distrito Nacional,

República Dominicana,

4 de Junio de 2020

Indice

Indice

1

Introduccion

Este plan de pruebas vamos a estar probando los métodos de la clase Range y verificar si cada una de ellas funciona cómo debe funcionar a través de las pruebas. Por eso, este documento servirá como guía para la elaboracion y ejecucion de las pruebas.

Estructura General del Programa

(C01)

```
Clase Range{  
  Int[] arreglo de numero = [ ];  
  Int valor_maximo, valor_minimo;
```

(C01-M01)

```
Constructor (string expression){  
  Convertir string a un arreglo de char  
  Sacar un intervalo del conjunto  
  Crear Objeto con los datos  
  Devolver el Objeto  
}
```

(C01-M02)

```
Contains (string expression){  
  Convertir string a un arreglo de char  
  Sacar un intervalo del conjunto  
  Crear Objeto con los datos  
  Comparar los límites inferiores y superiores de ambos objetos  
  Devolver bool dependiendo de lo comparado  
}
```

(C01-M03)

```
DoesNotContain(string expression){  
  Convertir string a un arreglo de char  
  Sacar un intervalo del conjunto  
  Crear Objeto con los datos  
  Comparar los límites inferiores y superiores de ambos objetos  
  Devolver bool dependiendo de lo comparado  
}
```

(C01-M04)

GetAllPoints(){

For loop, iterando desde el valor mínimo hasta el valor máximo y guardar los números en una lista.

Devolvemos esa lista

}

(C01-M05)

ContainsRange(Objeto Range){

Comparación entre los límites inferiores y superiores de los dos objetos

Devolver un bool dependiendo de lo comparado

}

(C01-M06)

DoesNotContain(Objeto Range){

Comparación entre los límites inferiores y superiores de los dos objetos

Devolver un bool dependiendo de lo comparado

}

(C01-M07)

EndPoints(){

Crear un string con el valor mínimo y valor máximo

Devolvemos string

}

(C01-M08)

OverlapsRange(Objeto Range){

Hacemos iteración y buscamos por elementos similares

Agregamos los elementos similares a una lista

Si la longitud de la lista es mayor que uno, devolver true

En caso contrario, devolver false

}

(C01-M09)

```
Equals(Objeto Range){  
  Comparamos los limites de ambos objetos  
  Si son iguales, devolver true  
  En caso contrario, devolver false  
}
```

(C01-M10)

```
NotEquals(Objeto Range){  
  Comparamos los limites de ambos objetos  
  Si son iguales, devolver false  
  En caso contrario, devolver true  
  
}  
  
}
```

Tabla de Pruebas

ID del metodo	Range (Objeto)	Input	Output
C01-M01		"[-8, 0]"	Objeto Rango ("[-8,0]")
C01-M01		"(-7,-4)"	Objeto Rango ("(-7,-4)")
C01-M01		"[2, 10]"	Objeto Rango ("[2, 10]")
C01-M01		"(4,8)"	Objeto Rango ("(4,8)")
C01-M02	Objeto Rango ("[-8,0]")	"[-4,-2]"	True
C01-M02	Objeto Rango ("(-7,-4)")	"(4, 10)"	False
C01-M02	Objeto Rango ("[2, 10]")	"(2, 10)"	True
C01-M02	Objeto Rango ("(4,8)")	"[-2,0]"	False
C01-M03	Objeto Rango ("[-8,0]")	"[-4,-2]"	False
C01-M03	Objeto Rango ("(-7,-4)")	"(4, 10)"	True
C01-M03	Objeto Rango ("[2, 10]")	"(2, 10)"	False

	10]”)		
C01-M03	Objeto Rango (“(4,8)”)	“[-2,0]”	True
C01-M04	Objeto Rango (“[-8,0]”)		“{-8,-7,-6,-5,-4,-3,-2,-1,0}”
C01-M04	Objeto Rango (“(-7,-4)”)		“{-6,-5}”
C01-M04	Objeto Rango (“[2, 10]”)		“{2,3,4,5,6,7,8,9,10}”
C01-M04	Objeto Rango (“(4,8)”)		“{5,6,7}”
C01-M05	Objeto Rango (“[-8,0]”)	Objeto Rango (“(-7,-4)”)	True
C01-M05	Objeto Rango (“[-8,0]”)	Objeto Rango (“(4,8)”)	False
C01-M05	Objeto Rango (“[2, 10]”)	Objeto Rango (“(4,8)”)	True
C01-M05	Objeto Rango (“[2, 10]”)	Objeto Rango (“(-7,-4)”)	False
C01-M06	Objeto Rango (“[-8,0]”)	Objeto Rango (“(-7,-4)”)	False

C01-M06	Objeto Rango ("[-8,0]")	Objeto Rango ("(4,8)")	True
C01-M06	Objeto Rango ("[2, 10]")	Objeto Rango ("(4,8)")	False
C01-M06	Objeto Rango ("[2, 10]")	Objeto Rango ("(-7,-4)")	True
C01-M07	Objeto Rango ("[-8,0]")		"{-8,0}"
C01-M07	Objeto Rango ("(-7,-4)")		"{-6, -5}"
C01-M07	Objeto Rango ("[2, 10]")		"{2,10}"
C01-M07	Objeto Rango ("(4,8)")		"{5,7}"
C01-M08	Objeto Rango ("[-8,0]")	Objeto Rango ("(-7,-4)")	True
C01-M08	Objeto Rango ("[-8,0]")	Objeto Rango ("[2, 10]")	False
C01-M08	Objeto Rango ("(4,8)")	Objeto Rango ("(-7,-4)")	False
C01-M08	Objeto Rango ("(4,8)")	Objeto Rango ("[2, 10]")	True

C01-M09	Objeto Rango ("[-8,0]")	Objeto Rango ("[-8,0]")	True
C01-M09	Objeto Rango ("[-8,0]")	Objeto Rango ("(-7,-4)")	False
C01-M09	Objeto Rango ("[2, 10]")	Objeto Rango ("(2, 10)")	True
C01-M09	Objeto Rango ("[2, 10]")	Objeto Rango ("[4,8]")	False
C01-M10	Objeto Rango ("[-8,0]")	Objeto Rango ("[-8,0]")	False
C01-M10	Objeto Rango ("[-8,0]")	Objeto Rango ("(-7,-4)")	True
C01-M10	Objeto Rango ("[2, 10]")	Objeto Rango ("(2, 10)")	False
C01-M10	Objeto Rango ("[2, 10]")	Objeto Rango ("[4,8]")	True