Design Specification, Sorting part

Senni Tan, tans28@mcmcaster.ca

April 6, 2020

Comparator Module

Module

Comparable

Uses

Job

Syntax

Exported Access Programs

Routine name	In	Out	Exceptions
CompareString	Job, Job	$\mathbb{Z}$	
CompareOutlook	Job, Job	$\mathbb{Z}$	
CompareNOC	Job, Job	$\mathbb{Z}$	
CompareRegionS	Job, Job	$\mathbb{Z}$	

Semantics

Access Routine Semantics

CompareString(a, b):

- output: out := a.get_title.compareTo(b.get_title)
- exception: None

// compareTo is a build in method to compare String in lexgraphical order.

CompareOutlook(a, b):

- output: out := (a.get_outlook > b.get_outlook) ⇒ 1 | (a.get_outlook < b.get_outlook) ⇒ -1 | 0
- exception: None

CompareNOC(a, b):

- output: out := (a.get_noc(0) > b.get_noc(0)) ⇒ 1 | (a.get_noc(0) < b.get_noc(0)) ⇒ -1 | 0

- exception: None

CompareRegionS(a, b):

- output: *out* := a.get_regions.compareTo(b.get_regions)
- exception: None

// compareTo is a build in method to compare String in lexgraphical order.

Sorting Module

Module

Sorting

Uses

Comparable

Syntax

Exported Access Programs

Routine name	**In**	**Out**	**Exceptions**
sortString	Seq of Job		
sortOutlook	Seq of Job		
sortNOC	Seq of Job		
sortRegionS	Seq of Job		

Semantics

Access Routine Semantics

sortString(a):

- transition: sortString(a, 0, $|a|$-1)
- exception: None

sortOutlook(a):

- transition: sortOutlook(a, 0, $|a|$-1)
- exception: None

sortNOC(a):

- transition: sortNOC(a, 0, $|a|$-1)
- exception: None

sortRegionS(a):

- transition: sortRegionS(a, 0, $|a|$-1)
- exception: None

Local Functions

exch: Seq of Job × $\mathbb{Z} \times \mathbb{Z} \rightarrow$ None
exch$(a, i, j) \equiv$ exchange a[i] and a[j] in the array

sortString: Seq of Job × $\mathbb{Z} \times \mathbb{Z} \rightarrow$ None
sortString$(a, lo, hi) \equiv$ (hi <= lo) $\Rightarrow$ return | sortString(a, lo, j-1) && sortString(a, j+1, hi) where j = partitionString(a, lo, hi)

partitionString: Seq of Job × $\mathbb{Z} \times \mathbb{Z} \rightarrow$ None
partitionString$(a, lo, hi) \equiv$ partition on array a using ComapreString, see detail in code

sortOutlook: Seq of Job × $\mathbb{Z} \times \mathbb{Z} \rightarrow$ None
sortOutlook$(a, lo, hi) \equiv$ (hi <= lo) $\Rightarrow$ return | sortOutlook(a, lo, j-1) && sortOutlook(a, j+1, hi) where j = partitionOutlook(a, lo, hi)

partitionOutlook: Seq of Job × $\mathbb{Z} \times \mathbb{Z} \rightarrow$ None
partitionOutlook$(a, lo, hi) \equiv$ partition on array a using ComapreOutlook, see detail in code

sortNOC: Seq of Job × $\mathbb{Z} \times \mathbb{Z} \rightarrow$ None
sortNOC$(a, lo, hi) \equiv$ (hi <= lo) $\Rightarrow$ return | sortNOC(a, lo, j-1) && sortNOC(a, j+1, hi) where j = partitionNOC(a, lo, hi)

partitionNOC: Seq of Job × $\mathbb{Z} \times \mathbb{Z} \rightarrow$ None
partitionNOC$(a, lo, hi) \equiv$ partition on array a using ComapreNOC, see detail in code

sortRegionS: Seq of Job × $\mathbb{Z} \times \mathbb{Z} \rightarrow$ None
sortRegionS$(a, lo, hi) \equiv$ (hi <= lo) $\Rightarrow$ return | sortRegionS(a, lo, j-1) && sortRegionS(a, j+1, hi) where j = partitionRegionS(a, lo, hi)

partitionRegionS: Seq of Job × $\mathbb{Z} \times \mathbb{Z} \rightarrow$ None
partitionRegionS$(a, lo, hi) \equiv$ partition on array a using ComapreRegionS, see detail in code