Specification

COMPSCI 2XB3

April 10, 2020

This Module Interface Specification (MIS) document contains three main modules; Queue, RedBlackBST, EarthquakeT, and CSVreader. It also contains a small EarthquakeBag module.

EarthquakeBag Module

Template Module

EarthquakeBag is seq of EarthquakeT

Generic Queue Module

Generic Template Module inherits Iterable(T)

Queue(T)

Uses

None

Syntax

Exported Constants

None

Exported Types

Queue = ?

Internal Types

Node = ?

# Internal Node type has a link to next item in the queue.

Exported Access Programs

Routine name	**In**	**Out**	**Exceptions**
Queue		Queue	
isEmpty		$\mathbb{B}$	
enqueue	T		
toString		String	
start			
next		T	NoSuchElementException

Semantics

State Variables

first: Node
last: Node
$n : \mathbb{N}$
s: seq of T

# For simplification, the linked-node structure is represented by seq of T.
# s[1] is the first Node.
# s[n] is the last Node.

State Invariant

None

Assumptions

None

Access Routine Semantics

Queue():

- transition: first, last, n := null, null, 0
- output: $out :=$ self
- exception: none

isEmpty():

- output: $out := (n = 0) \Rightarrow True | True \Rightarrow False$
- exception: None

enqueue(item):

- output: $out := s||$item
- exception: None

toString():

- output: $out := out||(\forall\, i : \mathbb{N}|i \in [1..n] : s[i])$
- exception: None

<u>Iterator Methods</u>:

$i : \mathbb{N}$

start():

- transition: $i := 0$
- exception: none

next():

- transition-output: $i, out := i + 1, s[i]$
- exception: $(i > n) \Rightarrow$ NoSuchElementException

Considerations

When an instance of Queue is iterated in a loop, an iterator consisting of these two methods is returned, and the start() method is call initially, and for the successive iterations next() method is call.

Generic RedBlackBST Module

Generic Template Module

RedBlackBST(T with Comparable(T), V)

Uses

Queue

Syntax

Exported Types

RedBlackBST = ?

Internal Types

Node = ?
State Variables of Node:
key: Key, lst: seq of V, left: Node, right: Node, color: B, size: N

# Internal Node type was modified to store a seq of V.

Exported Access Programs

Routine name	**In**	**Out**	**Exceptions**
RedBlackBST		RedBlackBST	
size		$\mathbb{N}$	
isEmpty		$\mathbb{B}$	
get	T	seq of V	
put	T, V		
min		T	
max		T	
keys		Queue of T	
keys	T, T	Queue of T	
values	T, T	Queue of V	

Semantics

State Variables

root: Node
RED: $\mathbb{B}$
BLACK: $\mathbb{B}$
s: set of $\langle T, V \rangle$

# For simplification, the linked-node structure is represented by set of $\langle T, V \rangle$.

State Invariant

RED = True
BLACK = False

Assumptions

None

Access Routine Semantics

RedBlackBST():

- transition: None
- output: $out := self$
- exception: None

size():

- output: $out := root.size$
- exception: None

isEmpty():

- output: $out := (root = null) \Rightarrow True | True \Rightarrow False$
- exception: None

get(key):

- output: $out :=$ L where $\langle x, L\rangle \in s \wedge (x.key = key)$
- exception: None

put(key, val):

- transition: $s := \{\langle x, L\rangle : \langle T, V\rangle | \langle x, L\rangle \in s : (x.key = key \Rightarrow \langle x, L||[val]\rangle | \text{True} \Rightarrow \langle x, L\rangle)\}$
- exception: None

min():

- output: $out :=$ smallest key in s
- exception: None

max():

- output: $out :=$ largest key in s
- exception: None

keys():

- output: $out := out||(\forall\langle x, L\rangle : \langle T, V\rangle | \langle x, L\rangle \in s : x.key)$
- exception: None

keys(lo, hi):

- output: $out := out||(\forall\langle x, L\rangle : \langle T, V\rangle | \langle x, L\rangle \in s \wedge lo \leq x.key \leq hi : x.key)$
- exception: None

values(lo, hi):

- output: $out := out||(\forall\langle x, L\rangle : \langle T, V\rangle | \langle x, L\rangle \in s \wedge lo \leq x.key \leq hi : L)$
- exception: None

EarthquakeT Module

Template Module

EarthquakeT

Uses

LocalDateTime, PointT

Syntax

Exported Types

EarthquakeT = ?

ColorRating = { NOCOLOR, ZERO, PURPLE, BLUE, GREEN, YELLOW, ORANGE, RED }

MagType = { M5, mb, MB, Mb, MC, Mc, mc, ML, MLSn, MN, MS, MW, Ms, Mw, BLANK }

# EarthquakeT implements Comparable(EarthquakeT)

Exported Constants

None

Exported Access Programs

Routine name	In	Out	Exceptions
EarthquakeT	String, String, LocalDateTime, $\mathbb{R}, \mathbb{R}, \mathbb{R}, \mathbb{R}$, MagType, ColorRating	EarthquakeT	
getNameOfProv		String	
getPlace		String	
getPointT		PointT	
getMag		$\mathbb{R}$	
getDph		$\mathbb{R}$	
getMagitudeType		MagType	
getDate		LocalDateTime	
getColor		ColorRating	
compareTo	EarthquakeT	$\mathbb{Z}$	
equals	EarthquakeT	$\mathbb{B}$	

Semantics

State Variables

place: String
nameOfProv: String
date: LocalDateTime
lat: $\mathbb{R}$
lng: $\mathbb{R}$
dph: $\mathbb{R}$
mag: $\mathbb{R}$
magnitudeType: MagType
color: ColorRating

State Invariant

None

Assumptions

Two earthquakes are not the same if they happened to have two different dates or two different places recorded.

Access Routine Semantics

EarthquakeT(place, prov, date, lat, lng, dph, mag, mgT, color):

- transition:
  lat, lng, place, nameOfProv, date, dph, mag, magnitudeType, color :=
  lat, lng, place, prov, date, dph, mag, mgT, color
- output: *out* := *self*
- exception: None

getNameOfProv():

- output: *out* := nameOfProv
- exception: None

getPlace():

- output: *out* := place
- exception: None

getPointT():

- output: *out* := PointT(lat, lng)
- exception: None

getMag():

- output: *out* := mag
- exception: None

getDph():

- output: *out* := dph
- exception: None

getMagitudeType():

- output: *out* := magnitudeType
- exception: None

getDate():

- output: *out* := date
- exception: None

getColor():

- output: *out* := color
- exception: None

compareTo(eq):

- output: *out* := an integer value according to the following table.

	out :=
$this$.mag < eq.mag	-1
$this$.mag > eq.mag	1
$this$.mag = eq.mag	0

- exception: None

equals(that):

- output: *out* := (sameDate ∧ samePoint ∧ samePlace ∧ sameDepth ∧ sameMagValue ∧ sameMagType ∧ sameEqClass) $\Rightarrow True | True \Rightarrow False$
- exception: None

Local Functions

sameDate: EarthquakeT $\rightarrow \mathbb{B}$
sameDate$(d) \equiv (d.\text{date}) = (\mathit{this}.\text{date})$
[Returns true if the given EarthquakeT object has the same date as the current. —SS]

samePoint: EarthquakeT $\rightarrow \mathbb{B}$
samePoint$(d) \equiv (d.\text{Point}) = (\mathit{this}.\text{Point})$
[Returns true if the given EarthquakeT object has the same Point as the current. —SS]

samePlace: EarthquakeT $\rightarrow \mathbb{B}$
samePlace$(d) \equiv (d.\text{place}) = (\mathit{this}.\text{place})$
[Returns true if the given EarthquakeT object has the same place as the current. —SS]

sameDepth: EarthquakeT $\rightarrow \mathbb{B}$
sameDepth$(d) \equiv |d.\text{dph} - \mathit{this}.\text{dph}| < 0.0000001$
[returns true if the given EarthquakeT object has the same depth value as the current within the tolerance. —SS]

sameMagValue: EarthquakeT $\rightarrow \mathbb{B}$
sameMagValue$(d) \equiv |d.\text{mag} - \mathit{this}.\text{mag}| < 0.0000001$
[returns true if the given EarthquakeT object has the same magnitude value as the current within the tolerance. —SS]

sameMagType: EarthquakeT $\rightarrow \mathbb{B}$
sameMagType$(d) \equiv (d.\text{magnitudeType}) = (\mathit{this}.\text{magnitudeType})$
[Returns true if the given EarthquakeT object has the same magnitude type as the current. —SS]

sameEqClass: EarthquakeT $\rightarrow \mathbb{B}$
sameEqClass$(d) \equiv (d.\text{color}) = (\mathit{this}.\text{color})$
[Returns true if the given EarthquakeT object has the same class as the current. —SS]

CSVreader Module

Module

CSVreader

Uses

CityPostT, CityT, EarthquakeT,
EarthquakeT.ColorRating, EarthquakeT.MagType,
EarthquakeBag, GeoCollection, RedBlackBST

Syntax

Exported Constants

None

Exported Access Programs

Routine name	In	Out	Exceptions
readEarthquakes	String, EarthquakeBag		
readEarthquakesBST	String, RedBlackBST		
readPopulation	String, GeoCollection		
readCityPosition	String, seq of CityPostT		
rmFirstLastQuote	String	String	
generateColorRating	$\mathbb{R}$	ColorRating	
fullProvName	String	String	

Semantics

Environment Variables

None

State Variables

None

State Invariant

None

Assumptions

None

Access Routine Semantics

readEarthquakes(filename, bag):

- transition: Read each line of the earthquake csv file and convert to EarthquakeT object, which is stored in a EarthquakeBag.
- exception: None

readEarthquakesBST(filename, bst):

- transition: Read each line of the earthquake csv file and convert to EarthquakeT object, which is stored in a RedBlackBST.
- exception: None

readPopulation(filename, geoCollec):

- transition: Read each line of the population csv file and convert to CityT object, which is stored in a GeoCollection HashMap.
- exception: None

readCityPosition(filename, cityPostList):

- transition: Read each line of the city coordinates csv file and convert to CityPostT object, which is stored in a list of cities.
- exception: None

rmFirstLastQuote(cell):

- transition: Remove first and last double quotations from a string.
- exception: None

generateColorRating(cell4):

- transition: Generate an enum ColorRating type based on the magnitude of earthquake.
- exception: None

fullProvName(nameP):

- output: a new province name similar to the following table.

	nameP =	*out* :=
$\lvert\text{nameP}\rvert = 2$	ON	Ontario
	QC, PQ	Quebec
	NS	Nova Scotia
	NB	New Brunswick
	MB	Manitoba
	BC	British Columbia
	PE	Prince Edward Island
	SK	Saskatchewan
	AB	Alberta
	NL	Newfoundland and Labrador
	NU	Nunavut
	NT	Northwest Territories
	YT	Yukon
	AK	Alaska
	WA	Washington
	default	UNLOCATED
$\lvert\text{nameP}\rvert \neq 2$	VANCOUVER ISLAND	British Columbia
	SOUTHERN QUEBEC	Quebec
	default	UNLOCATED

- exception: None

Considerations

There are a number of different variations of geolocation names in the earthquake csv file, for these an appropriate province name should be assigned. For any that could not be matched to a province name, UNLOCATED should be assigned.