

FireAware, Specification

FireAware team

March 15, 2019

This Module Interface Specification (MIS) document contains modules, types and methods for implementing the state of FireAware project.

Records Type Module

Library (exporting types)

Uses

None

Syntax

Exported Constants

None

Exported Types

CoordinatesT = tuple of (longitude: float, latitude: float)

State = visited, visiting, unvisited

Exported Access Programs

None

Semantics

State Variables

None

State Invariant

None

Building ADT

Template Module

Building

Uses

None

Syntax

Exported Constants

None

Exported Types

BuildingT = ?

Exported Access Programs

Routine name	In	Out	Exceptions
new SeqADT	sequence of T	SeqADT	
init			
getCoordinates		T	
getBourough		T	
getTotalInspection		T	
getDate		T	
setDate		T	
incrementTotalInspection		T	
nextInspectionDate		B	

Semantics

Need to be complete...

State Variables

date: ? idk what to put here

coordinates: coordinatesT

bourough: String
buildingId: String
totalInspection: Int
isSafe: \mathbb{B}

State Invariant

None

Generic Node Module

Generic Template Module

Node(T)

Uses

State

Syntax

Exported Access Programs

Routine name	In	Out	Exceptions
new Stack	seq of T	Stack	none
push	T	Stack	none
pop		Stack	out_of_range
top		T	out_of_range
size		N	
toSeq		seq of T	

Semantics

State Variables

visitState: State

connectedTo: a sequence of tuples of (node: Node, weight: double)

State Invariant

None

Access Routine Semantics

need to be completed...

Generic Graph Module

Generic Template Module

Graph(T)

Uses

Node(T)

Syntax

Exported Types

Stack = ?

[\[What should be written here? —SS\]](#)

Exported Constants

None

Exported Access Programs

Routine name	In	Out	Exceptions
new Stack	seq of T	Stack	none
push	T	Stack	none
pop		Stack	out_of_range
top		T	out_of_range
size		N	
toSeq		seq of T	

Semantics

State Variables

S: sequence of T

State Invariant

None

Assumptions & Design Decisions

- 1...
- 2...

Access Routine Semantics

new Stack(s):

- transition: $S := s$
- output: $out := self$
- exception: none

push(e):

- output: $out := new\ Stack(S \parallel \langle e \rangle)$
- exception: none

pop():

- output: $out := new\ Stack(new\ Stack(S[0..|S| - 2]))$
- exception: $(0 \geq (|S|) \Rightarrow out_of_range)$

top():

- output: $out := S[|S| - 1]$
- exception: $(0 \geq (|S|) \Rightarrow out_of_range)$

BuildingGraph Module

Template Module

BuildingGraph is Graph(Building)

Critique of Design