# **AA AirLock Software Specs**

## **PressureSensor**

PressureSensor(double initialPressure) throws PressureException {

if initial pressure is negative

throws PressureException with informative message

otherwise sets the pressure reading to initialPressure

double getPressure()

returns the pressure reading currently set

void setPressure(double newPressure)throws PressureException {

if newPressure is negative then

throws PressureException with informative message

otherwise sets the pressure reading to newPressure

## **Door**

Door(IPressureSensor externalSensor, IPressureSensor internalSensor,

DoorState initialState)

**Note: external is the away opening side. (I.e. towards the outside environment of the spacecraft for the external door, and towards inside cabin of the spacecraft for the inner door – both AWAY from airlock)**

if sensors supplied are not valid instances of IPressureSensor then

Throws a DoorException with informative message

Sets external and internal pressure sensors.

Sets initial door state (OPEN, CLOSED)

if Door is open and initial pressures are unequal (diff > TOLERANCE) then

Throws a DoorException with informative message

void open() throws DoorException;

if Door is already open then

throws a DoorException with informative message

if Door is closed and pressures are unequal (diff > TOLERANCE) then

throws a DoorException with informative message

otherwise opens door (Door state becomes OPEN) i

void close() throws DoorException;

if Door is already closed then

throws a DoorException with informative message

otherwise closes door (Door state becomes CLOSED)

double getExternalPressure();

returns the pressure reading from the external sensor

double getInternalPressure();

returns the pressure reading from the internal sensor

boolean isOpen();

if door state is OPEN then

returns TRUE,

otherwise returns FALSE

boolean isClosed();

if door state is CLOSED then

returns TRUE,

otherwise returns FALSE

## **AirLock**

AirLock(IDoor externalDoor, IDoor internalDoor, IPressureSensor lockSensor)

Sets door and sensor dependencies,

Sets the initial operation state to MANUAL

if both doors are CLOSED then

Sets initial airlock state to SEALED,

otherwise

Sets initial airlock state to UNSEALED,

void openOuterDoor() throws AirLockException;

**(Note: pseudocode indentation indicates nesting)**

if outer door is already open then

throw an AirLockException reporting door is open

otherwise

try

if operation mode is AUTO then

if inner Door is open then

close inner door

equalise lock pressure with external environment pressure

open outer door

set airlock state to UNSEALED

catch any DoorException or PressureException

throw an AirLockException chaining any previous exception

void openInnerDoor() throws AirLockException, DoorException;

**(Note: pseudocode indentation indicates nesting)**

if inner door is already open then

throw AirLockException reporting door is open

otherwise

try

if operation mode is AUTO then

if outer Door is open then

close outer door

equalise lock pressure with internal cabin pressure

open inner door

set airlock state to UNSEALED

catch any DoorException or PressureException

throw an AirLockException chaining any previous exception

void closeOuterDoor() throws AirLockException;

**(Note: pseudocode indentation indicates nesting)**

try

close external door

if internal door is also closed then

set airlock state to SEALED

catch DoorException

throw an AirLockException chaining the DoorException

void closeInnerDoor() throws AirLockException;

**(Note: pseudocode indentation indicates nesting)**

try

close internal door

if external door is also closed then

set airlock state to SEALED

catch DoorException

throw an AirLockException chaining the DoorException

void equaliseWithCabinPressure() throws AirLockException;

if airlock state is not SEALED then

throws AirLockException

otherwise equalises lock pressure with spacecraft cabin pressure

void equaliseWithEnvironmentPressure() throws AirLockException;

if airlock state is not SEALED then

throws AirLockException

otherwise equalises lock pressure with spacecraft exterior environment pressure

void toggleOperationMode() throws AirLockException;

if airlock state is not SEALED then

throws AirLockException

otherwise toggles operationState between MANUAL and AUTO

boolean isInManualMode();

returns true if operationState is MANUAL otherwise false

boolean isInAutoMode();

returns true if operationState is AUTO otherwise false

boolean isInManualMode();

returns true if operationState is AUTO otherwise false

boolean isOuterDoorClosed();

returns true if outer door is CLOSED otherwise false

boolean isOuterDoorOpen();

returns true if outer door is OPEN otherwise false

boolean isInnerDoorClosed();

returns true if inner door is CLOSED otherwise false

boolean isInnerDoorOpen();

returns true if inner door is OPEN otherwise false

boolean isSealed();

returns true if airlock is SEALED

boolean isUnsealed();

returns true if airlock is UNSEALED