Past Paper 2

Question # 2

[People, Subject]

University

students : ~~Peo~~ ℙ People

subjects : ℙ Subjects

enrollement : People ⟷ Subject

dom enrollement ⊆ students

ran enrollement ⊆ subjects

InitUniversity

University'

subjects = ϕ

students = ϕ

enrollement = ϕ

i)

newStudent

Δ University

std? : Person

---

std? ∉ Person

students' = students ∪ {std?}

subjects' = subjects

enrollement' = enrollement

ii)

Student Leaves

Δ University

s? : Person

---

s? ∈ students

s? ∉ dom enrollement

students' = students \ {s?}

enrollement' = enrollement

subject' = subject

iii)

PersonSubjects

Ξ University

std? : Person

reg! : Subjects

---

reg! = enrollement(|{std?}|)

Question #1

[MSG]

max : N

Buffer

items : seq MSG

#items ≤ max

i)

initBuffer

Buffer

items = <>

ii)

Add

ΔBuffer

msg? : MSG

items ⌢ ⟨msg?⟩ = <>
#items < max

items = ⟨msg?⟩ ⌢ items

iii)

Remove

ΔBuffer

msg? : MSG

msg? : items ↾{msg?} ≠ <>

items' = items - {{msg?}}

Q#3

capacity : ℤ

CarPark

cars : ℤ

cars ≥ 0

cars ≤ capacity

initCarPark

CarPark

cars = 0

Enter

Δ CarPark

cars < capacity

cars' = cars + 1

Depart

Δ CarPark

cars > 0

cars' = cars − 1

QuerySpace

Ξ CarPark

count! : ℤ

count! = capacity − cars

Question #4

types
values
MIN : ℤ ∧ MAX : ℤ
state IncubatorMon of
temp : [ℤ]

inv mk-IncubatorMonitor(t) ≜ ~~temp~~ MIN ≤ temp ≤ MA
init mk-IncubatorMonitor(t) ≜ temp = nil

end

operations

increment()
ext wr temp : [ℤ]
pre temp < MAX
post temp = $\overline{temp}$ + 1

decrement()
ext wr temp : [ℤ]
pre temp > MIN
post temp = $\overline{temp}$ - 1

getTemp() output : ℤ

ext rd temp : [ℤ]

pre true

post output = temp