Name: Lê Bảo Phúc
ID: ITDSIU18033

OOAD – Assignment

Task 2.

1. Add a Branch

input:

branchNr?:String

address?:String

phoneNr?:String

output:

NONE

pre:

// branchNr? Is new

not exists b in branchList | b.branchNr = branchNr?

post:

// create new branch

let b = new Branch(branchNr?, address?, phoneNr?) |

b.branchNr = branchNr?

b.address = address?

b.phoneNr = phoneNr?

b.cars = new List<Car>()

// add new branch to list

add b to branchList

2. **Make a pair of branches neighbors to each other**

input:

branchNrA?:String

output:

NONE

pre:

// branchNrA? Exists

exists b in branchList | b.branchNr = branchNrA?

// branchNrB? Exists

exists b in branchList | b.branchNr = branchNrB?

// branchNrA? different from branchNrB?

let b = element branchList | branchNrA? <> branchNrB?

post:

// retrieve the branch

Let b = element in branchList | b.branchNr = branchNrA?

Let b = element in branchList | b.branchNr = branchNrB?

// create new pair of neighbor

let n = new Neighbor(BranchNrA?, branchNrB?) |

n.branchNrA = branchNrA?

n.branchNrB = branchNrB?

// add a pair to list

add n to neighborList

3. **Add a car rental group**

input:

code?:String

description?:String

dailyRentalRate?:Real
output:
NONE
pre:
// code? Is new
not exists rg in rentalgroupList | rg.code = code?
post:
// create new rental group
let rg = new RentalGroup(code?, description?, dailyRentalRate?) |
rg.code = code?
rg.description = description?
rg.dailyRentalRate = dailyRentalRate?
rg.model = new Set<Model>()
// add new rental group to list
add rg to rentalgroupList

4. **Add a model**

input:
modelNr?:String
description?:String
gearType?:String
petrolConsumption?:Real
output:
NONE
pre:
// modelNr? Is new

not exists m in modelList | m.modelNr = modelNr?

post:

// create new model

let m = new Model(modelNr?, description?, gearType?, petrolConsumption?) |

m.modelNr = modelNr?

m.description = description?

m.gearType = gearType?

m.petrolConsumption = petrolConsumption?

m.cars = new List<Car>()

// add new model to list

add m to modelList

5. Add a car

input:

regNr?:String

status?:String

output:

NONE

pre:

// regNr? Is new

not exists v in vehicleList | v.regNr = regNr?

post:

// create new car

Let v = new Car(regNr?, status?) |

v.regNr = regNr?

v.status = status?

v.rentals = new Set<Rental>()
// add new car to list
add v to vehicleList

6. **Add a customer**

input:
licenseNr?:String
name?:String
phoneNr?:String
discount?:Boolean
blacklist?:Boolean
output:
NONE
pre:
// licenseNr? Is new
not exists c in customerList | c.licenseNr = licenseNr?
post:
// create new customer
let c = new Customer(licenseNr?, name?, phoneNr?) |
c.licenseNr = licenseNr?
c.name = name?
c.phoneNr = phoneNr?
c.discount = discount?
c.blacklist = blacklist?
c.rentals = new Set<Rental> ()
// add customer to list

add c to customerList

7. **List cars that are available at a specified branch and belong to a specified rental group (do not include the cars at neighbor branches)**

input:

branchNr?:String

code?:String

status?:String

output:

regNr!

pre:

// branchNr? exists

exists b in branchList | b.branchNr = branchNr?

// code? exists

exists rg in rentalgroupList | rg.code = code?

// status is available

let v = element in vehicleList | v.status = status? then v.status = ‘available’

post:

// retrieve regNr

let v = element in vehicleList | v.regNr = regNr?

8. **List cars that are available at the neighbor branches of a specified branch and belong to a specified rental group (do not include the cars at the specified branch)**

input:

branchNrA?:String

code?:String

status?:String

output:

regNr!

pre:

// branchNrA? Exists

exists b in branchList | b.branchNr = branchNrA?

// branchNrB? Exists

exists b in branchList | b.branchNr = branchNrB?

// branchNrA? different from branchNrB?

let b = element branchList | branchNrA? <> branchNrB?

post:

// retrieve regNr

let v = element in vehicleList | v.regNr = regNr?

9. **Enter a walk-in rental**

input:

licenseNr?:String

rentalNr?:String

status?:String

output:

NONE

pre:

// licenseNr? exists

exists c in customerList | c.licenseNr = licenseNr?

// rentalNr? Is new

Not exist r in rentalList | r.rentalNr = rentalNr?

post:

// retrieve the customer

let c = element in customerList | c.licenseNr = licenseNr?

//create new rental

Let w = new WalkInRental(rentalNr?, status?, payment) |

w.rentalNr = rentalNr?

w.status = status?

w.payment = payment

// add rental to list

Add w to walkinrentalList

10. Record the return of a car

input:

rentalNr?:String

creationDate?:String

exRetDay?:Date

exRetTime?:Time

actRetDay?:Date

actRetTime?:Time

note?:String

output:

NONE

pre:

// rentalNr? Exists

exists w in walkinrentalList | w.rentalNr = rentalNr?

exists rsv in reservationrentalList | rsv.rentalNr = rentalNr?

// exRetDay? exists

exists w in walkinrentalList | w.exRetDay = exRetDay?

exists rsv in reservationrentalList | rsv.exRetDay = exRetDay?

//actRetDay? Is new

not exists r in rentalList | r.actRetDay = actRetDay?

post:

// create new the return

let r = new Rental(rentalNr?, creationDate?, exRetDay?, exRetTime?, actRetDay?, actRetTime?, note?) |

r.rentalNr = rentalNr?

r.creationDate = creationDate?

r.exRetDay = exRetDay?

r.exRetTime = exRetTime?

r.actRetDay = actRetDay?

r.actRetTime = actRetTime?

r.note = note?

r.reserRental = new List<ReservationRental>()

r.walkinRental = new List<WalkinRental>()

// add record to list

Add r to rentalList