

ALGORITHMS AND DATA STRUCTURES

C PROJECT

UNIVERSIDAD
AUSTRAL



Facultad de Ingeniería

Ariel Mirra
Franco Velardez

Contents

1	Appliance Shop	1
1.1	DataBase	1
1.2	Manufacturer	4
1.3	Provider	5
1.4	Appliance	5
1.5	ShoppingCart	6
1.6	ShoppingCartLine	8
1.7	Invoice	9
1.8	InvoiceLine	9
1.9	Label	10
2	Camera Shop	10
2.1	DataBase	10
2.2	Accessory	16
2.3	Camera	17
2.4	Cart	18
2.5	CartLine	19
2.6	Invoice	20
2.7	InvoiceLine	20
2.8	Manufacturer	21
2.9	Product	21
2.10	Provider	22
2.11	User	22
3	Library	24
3.1	Library	24
3.2	Loan	25
3.3	Material	26
3.4	Person	27
4	Hotel	29
4.1	Admin	29
4.2	Client	30
4.3	Hotel	31
4.4	Invoice	32
4.5	Receptionist	33
4.6	Reservation	33
4.7	Room	34
5	Movie Rental	35
5.1	Admin	35
5.2	Client	35
5.3	ClientsDatabase	36
5.4	Excess	37
5.5	License	38
5.6	Movie	39
5.7	MoviesDatabase	39

1 Appliance Shop

1.1 DataBase

Description: The DataBase holds the information of the hole system, is where manufacturers, providers and appliances are added, consulted or deleted.

createDataBase: $\text{---} \rightarrow \text{DataBase}$
Description: creates an empty DataBase
Precondition: ---
Postcondition: A DataBase is created
Classification: Constructor

growManufacturer: $\text{DataBase} \rightarrow \text{void}$
Description: expands the Manufacturer array inside the DataBase
Precondition: the DataBase must exist
Postcondition: the Manufacturer array is larger
Classification: Modifier

growProvider: $\text{DataBase} \rightarrow \text{void}$
Description: expands the Provider array inside the DataBase
Precondition: the DataBase must exist
Postcondition: the Provider array is larger
Classification: Modifier

growAppliance: $\text{DataBase} \rightarrow \text{void}$
Description: expands the Appliance array inside the DataBase
Precondition: the DataBase must exist
Postcondition: the Appliance array is larger
Classification: Modifier

addManufacturer: $\text{DataBase} \times \text{Manufacturer} \rightarrow \text{void}$
Description: adds a Manufacturer to the DataBase
Precondition:
the DataBase must exist
the Manufacturer must not be null
Postcondition: the Manufacturer is on the Database
Classification: Modifier

addProvider: DataBase x Provider \rightarrow void
Description: adds a Provider to the DataBase
Precondition:
the DataBase must exist
the Provider must not be null
Postcondition: the Provider is on the Database
Classification: Modifier

addAppliance: DataBase x Appliance x manufacturerName x providerName \rightarrow void
Description: adds an Appliance to the DataBase and connects it to a manufacturer and a provider currently on the DataBase
Precondition:
the DataBase must exist
the Appliance, manufacturerName and providerName must not be null
Postcondition: the Appliance is on the Database
Classification: Modifier

getManufacturer: DataBase x manufacturerName \rightarrow Manufacturer
Description: searches a Manufacturer on the DataBase and returns it if found
Precondition:
the DataBase and Manufacturer must exist
the manufacturerName must not be null
Postcondition: the Manufacturer is returned
Classification: analyzer

getProvider: DataBase x providerName \rightarrow Provider
Description: searches a Provider on the DataBase and returns it if found
Precondition:
the DataBase and Provider must exist
the providerName must not be null
Postcondition: the Provider is returned
Classification: analyzer

getAppliance: DataBase x applianceName \rightarrow Appliance
Description: searches a Appliance on the DataBase and returns it if found
Precondition:
the DataBase and Appliance must exist
the applianceName must not be null
Postcondition: the Appliance is returned
Classification: analyzer

removeManufacturer: DataBase x manufacturerName \rightarrow boolean
Description:
searches a Manufacturer on the DataBase and deletes it if found
Precondition:
the DataBase and Manufacturer must exist
the manufacturerName must not be null
Postcondition: if successfully removed returned true, false if not
Classification: Modifier

removeProvider: DataBase x providerName \rightarrow boolean
Description:
searches a Provider on the DataBase and deletes it if found
Precondition:
the DataBase and Provider must exist
the providerName must not be null
Postcondition: if successfully removed returned true, false if not
Classification: Modifier

removeAppliance: DataBase x applianceName \rightarrow boolean
Description:
searches an Appliance on the DataBase and deletes it if found
Precondition:
the DataBase and Appliance must exist
the applianceName must not be null
Postcondition: if successfully removed returned true, false if not
Classification: Modifier

manufacturerExist: DataBase x manufacturerName \rightarrow boolean

Description:

checks if a Manufacturer is in the DataBase

Precondition:

the DataBase and Manufacturer must exist

the manufacturerName must not be null

Postcondition: if successfully found returned true, false if not

Classification: analyzer

providerExist: DataBase x providerName \rightarrow boolean

Description:

checks if a Provider is in the DataBase

Precondition:

the DataBase and Provider must exist

the providerName must not be null

Postcondition: if successfully found returned true, false if not

Classification: analyzer

applianceExist: DataBase x applianceName \rightarrow boolean

Description:

checks if an Appliance is in the DataBase

Precondition:

the DataBase and Appliance must exist

the applianceName must not be null

Postcondition: if successfully found returned true, false if not

Classification: analyzer

destroyDataBase: DataBase \rightarrow void

Description: Deallocates memory assigned for a DataBase and all its components

Precondition: DataBase must not be null

Postcondition: Memory freed

Classification: Destroyer

1.2 Manufacturer

Description: The Manufacturer structure has the information of a manufacturer and it's details.

createManufacturer: name x description x address x city x telephoneNumber x web \rightarrow Manufacturer
Description: creates a Manufacturer
Precondition: strings passed must not be empty
Postcondition: A Manufacturer is created
Classification: Constructor

destroyManufacturer: Manufacturer \rightarrow void
Description: Deallocates memory assigned for a Manufacturer and all its components
Precondition: Manufacturer must not be null
Postcondition: Memory freed
Classification: Destroyer

1.3 Provider

Description: The Provider structure has the information of a provider and it's details.

createProvider: name x description x address x city x telephoneNumber x web \rightarrow Provider
Description: creates a Provider
Precondition: strings passed must not be empty
Postcondition: A Provider is created
Classification: Constructor

destroyProvider: Provider \rightarrow void
Description: Deallocates memory assigned for a Provider and all its components
Precondition: Provider must not be null
Postcondition: Memory freed
Classification: Destroyer

1.4 Appliance

Description: The Appliance structure holds the information of an Appliance, its label name model manufacturer provider and price.

createAppliance: name x model x model x price x id \rightarrow Appliance
Description: creates an Appliance with the components passed
Precondition: strings passed can't be empty
Postcondition: An Appliance is created
Classification: Constructor

setManufacturer: Appliance x manufacturerName \rightarrow void
Description: sets the Manufacturer of an Appliance
Precondition:
the Appliance must exist
the manufacturerName must not be null
Postcondition: the Appliance has the Manufacturer set
Classification: Modifier

setProvider: Appliance x providerName \rightarrow void
Description: sets the Provider of an Appliance
Precondition:
the Appliance must exist
the providerName must not be null
Postcondition: the Appliance has the Provider set
Classification: Modifier

destroyAppliance: Appliance \rightarrow void
Description: Deallocates memory assigned for an Appliance and all its components
Precondition: Appliance must not be null
Postcondition: Memory freed
Classification: Destroyer

1.5 ShoppingCart

Description: The ShoppingCart structure has the information of a shopping cart and it's details, it's used for storing appliances before purchase.

createShoppingCart: $\text{id} \rightarrow \text{ShoppingCart}$
Description: creates a ShoppingCart
Precondition: strings passed must not be empty
Postcondition: A ShoppingCart is created
Classification: Constructor

addApplianceToCart: $\text{ShoppingCart} \times \text{Appliance} \times \text{amount} \rightarrow \text{void}$
Description: adds an Appliance to the cart
Precondition:
the ShoppingCart and the Appliance must exist
amount > 0
Postcondition: the Appliance is in the ShoppingCart
Classification: Modifier

removeApplianceFromCart: $\text{ShoppingCart} \times \text{applianceName} \rightarrow \text{void}$
Description: removes an Appliance from the cart
Precondition: the ShoppingCart and the Appliance must exist
Postcondition: the Appliance has been removed from the ShoppingCart
Classification: Modifier

isFull: $\text{ShoppingCart} \rightarrow \text{boolean}$
Description: checks if a cart is full
Precondition: the ShoppingCart must exist
Postcondition: returned true if the cart is full, false if it's not
Classification: analyzer

grow: $\text{ShoppingCart} \rightarrow \text{void}$
Description: expands the ShoppingCart capacity
Precondition: the ShoppingCart must exist
Postcondition: the ShoppingCart is now larger
Classification: Modifier

total: $\text{ShoppingCart} \rightarrow \text{int}$
Description: sums the amount of money of every appliance inside the cart
Precondition: the ShoppingCart must exist
Postcondition: returned total ≥ 0
Classification: analyzer

emptyShoppingCart: $\text{ShoppingCart} \rightarrow \text{void}$
Description: empties the cart, deletes all ShoppingCartLine inside
Precondition: the ShoppingCart must exist
Postcondition: the ShoppingCart is now empty, there are no appliances inside
Classification: Modifier

destroyShoppingCart: $\text{ShoppingCart} \rightarrow \text{void}$
Description: Deallocates memory assigned for a ShoppingCart and all its components
Precondition: ShoppingCart must not be null
Postcondition: Memory freed
Classification: Destroyer

1.6 ShoppingCartLine

Description: The ShoppingCartLine structure holds the information of a single appliance in a ShoppingCart, it has the appliance, and the amount of that appliance in the cart.

createShoppingCartLine: $\text{Appliance} \times \text{units} \rightarrow \text{ShoppingCartLine}$
Description: creates an ShoppingCartLine with an appliance and it's units
Precondition: Appliance passed must not be empty
Postcondition: A ShoppingCartLine is created
Classification: Constructor

destroyShoppingCartLine: $\text{ShoppingCartLine} \rightarrow \text{void}$
Description: Deallocates memory assigned for a ShoppingCartLine and all its components
Precondition: ShoppingCartLine must not be null
Postcondition: Memory freed
Classification: Destroyer

1.7 Invoice

Description: The Invoice structure holds the information of a purchase, all the appliances purchased, their price, and the total amount to pay.

createInvoice: ShoppingCart \rightarrow Invoice
Description: creates an Invoice with the components on the ShoppingCart
Precondition: ShoppingCart passed must not be empty
Postcondition: An Invoice is created
Classification: Constructor

destroyInvoice: Invoice \rightarrow void
Description: Deallocates memory assigned for an Invoice and all its components
Precondition: Invoice must not be null
Postcondition: Memory freed
Classification: Destroyer

1.8 InvoiceLine

Description: The InvoiceLine structure holds the information of a single type of appliance purchased and the amount purchased, an Invoice contains multiple InvoiceLines.

createInvoiceLine: ShoppingCartLine \rightarrow InvoiceLine
Description: creates an InvoiceLine with the components on the ShoppingCartLine
Precondition: ShoppingCartLine passed must not be empty
Postcondition: An InvoiceLine is created
Classification: Constructor

destroyInvoiceLine: InvoiceLine \rightarrow void
Description: Deallocates memory assigned for an InvoiceLine and all its components
Precondition: InvoiceLine must not be null
Postcondition: Memory freed
Classification: Destroyer

1.9 Label

Description: The Label structure has the information of an appliance, and is created only when an appliance is created.

createLabel: $\text{name} \times \text{id} \rightarrow \text{Label}$
Description: creates a Label
Precondition: strings passed must not be empty
Postcondition: A Label is created
Classification: Constructor

destroyLabel: $\text{Label} \rightarrow \text{void}$
Description: Deallocates memory assigned for a Label and all its components
Precondition: Label must not be null
Postcondition: Memory freed
Classification: Destroyer

2 Camera Shop

2.1 DataBase

Description: The DataBase holds the information of the hole system, is where manufacturers, providers and products are added, consulted or deleted.

createDataBase: $\text{—} \rightarrow \text{DataBase}$
Description: creates an empty DataBase
Precondition: —
Postcondition: A DataBase is created
Classification: Constructor

growManufacturer: $\text{DataBase} \rightarrow \text{void}$
Description: expands the Manufacturer array inside the DataBase
Precondition: the DataBase must exist
Postcondition: the Manufacturer array is larger
Classification: Modifier

growProvider: DataBase \rightarrow void
Description: expands the Provider array inside the DataBase
Precondition: the DataBase must exist
Postcondition: the Provider array is larger
Classification: Modifier

growProduct: DataBase \rightarrow void
Description: expands the Product array inside the DataBase
Precondition: the DataBase must exist
Postcondition: the Product array is larger
Classification: Modifier

growCamera: DataBase \rightarrow void
Description: expands the Camera array inside the DataBase
Precondition: the DataBase must exist
Postcondition: the Camera array is larger
Classification: Modifier

growUser: DataBase \rightarrow void
Description: expands the User array inside the DataBase
Precondition: the DataBase must exist
Postcondition: the User array is larger
Classification: Modifier

growAccessory: DataBase \rightarrow void
Description: expands the Accessory array inside the DataBase
Precondition: the DataBase must exist
Postcondition: the Accessory array is larger
Classification: Modifier

addManufacturer: DataBase x Manufacturer \rightarrow void

Description: adds a Manufacturer to the DataBase

Precondition:

the DataBase must exist

the Manufacturer must not be null

Postcondition: the Manufacturer is in the Database

Classification: Modifier

addProvider: DataBase x Provider \rightarrow void

Description: adds a Provider to the DataBase

Precondition:

the DataBase must exist

the Provider must not be null

Postcondition: the Provider is in the Database

Classification: Modifier

addProduct: DataBase x Product x manufacturerID x providerID \rightarrow void

Description: adds a Product to the DataBase and connects it to a manufacturer and a provider currently on the DataBase

Precondition:

the DataBase must exist

the Product, manufacturerID and providerID must not be null

Postcondition: the Product is in the Database

Classification: Modifier

addCamera: DataBase x Camera x manufacturerID x providerID \rightarrow void

Description: adds a Camera to the DataBase and connects it to a manufacturer and a provider currently on the DataBase

Precondition:

the DataBase must exist

the Camera, manufacturerID and providerID must not be null

Postcondition: the Camera is in the Database

Classification: Modifier

addAccessory: DataBase x Accessory x manufacturerID x providerID \rightarrow void
Description: adds a Accessory to the DataBase and connects it to a manufacturer and a provider currently on the DataBase
Precondition:
the DataBase must exist
the Accessory, manufacturerID and providerID must not be null
Postcondition: the Accessory is in the Database
Classification: Modifier

addUser: DataBase x User \rightarrow void
Description: adds a User to the DataBase
Precondition:
the DataBase must exist
the User must not be null
Postcondition: the User is in the Database
Classification: Modifier

getManufacturer: DataBase x manufacturerID \rightarrow Manufacturer
Description: searches a Manufacturer on the DataBase and returns it if found
Precondition:
the DataBase and Manufacturer must exist
the manufacturerID must not be null
Postcondition: the Manufacturer is returned
Classification: analyzer

getProvider: DataBase x providerID \rightarrow Provider
Description: searches a Provider on the DataBase and returns it if found
Precondition:
the DataBase and Provider must exist
the providerID must not be null
Postcondition: the Provider is returned
Classification: analyzer

getProduct: DataBase x productID \rightarrow Product

Description: searches a Product on the DataBase and returns it if found

Precondition:

the DataBase and Product must exist

the productID must not be null

Postcondition: the product is returned

Classification: analyzer

removeManufacturer: DataBase x manufacturerName \rightarrow boolean

Description:

searches a Manufacturer on the DataBase and deletes it if found

Precondition:

the DataBase and Manufacturer must exist

the manufacturerName must not be null

Postcondition: if successfully removed returned true, false if not

Classification: Modifier

removeProvider: DataBase x providerName \rightarrow boolean

Description:

searches a Provider on the DataBase and deletes it if found

Precondition:

the DataBase and Provider must exist

the providerName must not be null

Postcondition: if successfully removed returned true, false if not

Classification: Modifier

removeProduct: DataBase x productID \rightarrow boolean

Description:

searches a product on the DataBase and deletes it if found

Precondition:

the DataBase and product must exist

the productID must not be null

Postcondition: if successfully removed returned true, false if not

Classification: Modifier

removeUser: DataBase x UserName \rightarrow boolean

Description:

searches a User in the DataBase and deletes it if found

Precondition:

the DataBase and User must exist

the UserName must not be null

Postcondition: if successfully removed returned true, false if not

Classification: Modifier

manufacturerExist: DataBase x manufacturerName \rightarrow boolean

Description:

checks if a Manufacturer is in the DataBase

Precondition:

the DataBase and Manufacturer must exist

the manufacturerName must not be null

Postcondition: if successfully found returned true, false if not

Classification: analyzer

providerExist: DataBase x providerName \rightarrow boolean

Description:

checks if a Provider is in the DataBase

Precondition:

the DataBase and Provider must exist

the providerName must not be null

Postcondition: if successfully found returned true, false if not

Classification: analyzer

productExist: DataBase x productID \rightarrow boolean

Description:

checks if a product is in the DataBase

Precondition:

the DataBase and product must exist

the productID must not be null

Postcondition: if successfully found returned true, false if not

Classification: analyzer

nextManufacturerID: DataBase \rightarrow int
Description: returns the next ID available for a manufacturer in a DataBase
Precondition: DataBase must not be null
Postcondition: ID returned is available
Classification: analyzer

nextProviderID: DataBase \rightarrow int
Description: returns the next ID available for a provider in a DataBase
Precondition: DataBase must not be null
Postcondition: ID returned is available
Classification: analyzer

nextProductID: DataBase \rightarrow int
Description: returns the next ID available for a product in a DataBase
Precondition: DataBase must not be null
Postcondition: ID returned is available
Classification: analyzer

destroyDataBase: DataBase \rightarrow void
Description: Deallocates memory assigned for a DataBase and all its components
Precondition: DataBase must not be null
Postcondition: Memory freed
Classification: Destroyer

2.2 Accessory

Description: represents an Accessory in the Camera Shop system.

createAccessory: name x description x price \rightarrow Accessory

Description: creates an Accessory

Precondition: details passed must not be null

Postcondition: an Accessory is created

Classification: Constructor

destroyAccessory: Accessory \rightarrow void

Description: Deallocates memory assigned for an Accessory and all its components

Precondition: Accessory must not be null

Postcondition: Memory freed

Classification: Destroyer

2.3 Camera

Description: Represents a camera in the Camera Shop system.

createCamera: name x megaPixels x zoom x lcdDislay x CameraType x price \rightarrow Camera

Description: creates a Camera

Precondition: details passed must not be null

Postcondition: a Camera is created

Classification: Constructor

addCameraAccessory: camera x accessoryID \rightarrow void

Description: adds an accessory id to the camera's accessory list, if it's not added already

Precondition: details passed must not be null

Postcondition: the Camera has the accessoryID in it's accessory list

Classification: modifier

removeCameraAccessory: camera x accessoryID \rightarrow void

Description: removes an accessory id from the camera's accessory list

Precondition: details passed must not be null

Postcondition: the Camera doesn't have the accessoryID in it's accessory list

Classification: modifier

destroyCamera: Camera \rightarrow void
Description: Deallocates memory assigned for a Camera and all its components
Precondition: Camera must not be null
Postcondition: Memory freed
Classification: Destroyer

2.4 Cart

Description: The Cart structure has the information of a cart and it's details, it's used for storing products before purchase.

createCart: id \rightarrow Cart
Description: creates a Cart
Precondition: strings passed must not be empty
Postcondition: A Cart is created
Classification: Constructor

addProductToCart: Cart x Product x amount \rightarrow void
Description: adds a Product to the cart
Precondition:
the Cart and the Product must exist
amount > 0
Postcondition: the Product is in the Cart
Classification: Modifier

removeApplianceFromCart: Cart x productID \rightarrow void
Description: removes a Product from the cart
Precondition: the Cart and the Product must exist
Postcondition: the Product has been removed from the Cart
Classification: Modifier

isFull: $\text{Cart} \rightarrow \text{boolean}$
Description: checks if a cart is full
Precondition: the Cart must exist
Postcondition: returned true if the cart is full, false if it's not
Classification: analyzer

grow: $\text{Cart} \rightarrow \text{void}$
Description: expands the Cart capacity
Precondition: the Cart must exist
Postcondition: the Cart is now larger
Classification: Modifier

total: $\text{Cart} \rightarrow \text{int}$
Description: sums the amount of money of every product inside the cart
Precondition: the Cart must exist
Postcondition: returned total ≥ 0
Classification: analyzer

emptyCart: $\text{Cart} \rightarrow \text{void}$
Description: empties the cart, deletes all CartLines inside
Precondition: the Cart must exist
Postcondition: the Cart is now empty, there are no products inside
Classification: Modifier

destroyCart: $\text{Cart} \rightarrow \text{void}$
Description: Deallocates memory assigned for a Cart and all its components
Precondition: Cart must not be null
Postcondition: Memory freed
Classification: Destroyer

2.5 CartLine

Description: The CartLine structure holds the information of a single product in a Cart, it has the product, and the amount of that product in the cart.

createCartLine: Product x units \rightarrow CartLine
Description: creates an CartLine with a product and it's units
Precondition: product passed must not be empty
Postcondition: A CartLine is created
Classification: Constructor

destroyCartLine: CartLine \rightarrow void
Description: Deallocates memory assigned for a CartLine and all its components
Precondition: CartLine must not be null
Postcondition: Memory freed
Classification: Destroyer

2.6 Invoice

Description: The Invoice structure holds the information of a purchase, all the products purchased, their price, and the total amount to pay.

createInvoice: Cart \rightarrow Invoice
Description: creates an Invoice with the components on the Cart
Precondition: Cart passed must not be empty
Postcondition: An Invoice is created
Classification: Constructor

destroyInvoice: Invoice \rightarrow void
Description: Deallocates memory assigned for an Invoice and all its components
Precondition: Invoice must not be null
Postcondition: Memory freed
Classification: Destroyer

2.7 InvoiceLine

Description: The InvoiceLine structure holds the information of a single type of product purchased and the amount purchased, an Invoice contains multiple InvoiceLines.

createInvoiceLine: $\text{CartLine} \rightarrow \text{InvoiceLine}$
Description: creates an InvoiceLine with the components on the CartLine
Precondition: CartLine passed must not be empty
Postcondition: An InvoiceLine is created
Classification: Constructor

destroyInvoiceLine: $\text{InvoiceLine} \rightarrow \text{void}$
Description: Deallocates memory assigned for an InvoiceLine and all its components
Precondition: InvoiceLine must not be null
Postcondition: Memory freed
Classification: Destroyer

2.8 Manufacturer

Description: The Manufacturer structure has the information of a manufacturer and it's details.

createManufacturer: name x description x address x city x telephoneNumber x country
 $\rightarrow \text{Manufacturer}$
Description: creates a Manufacturer
Precondition: strings passed must not be empty
Postcondition: A Manufacturer is created
Classification: Constructor

destroyManufacturer: $\text{Manufacturer} \rightarrow \text{void}$
Description: Deallocates memory assigned for a Manufacturer and all its components
Precondition: Manufacturer must not be null
Postcondition: Memory freed
Classification: Destroyer

2.9 Product

Description: Represents a product from the Camera Shop system in the DataBase. A product can be a camera or an accessory.

createProduct: name x ProductType x manufacturerID x providerID x productID x price
→ Product
Description: creates a Product
Precondition: strings passed must not be empty
Postcondition: A Product is created
Classification: Constructor

destroyProduct: Product → void
Description: Deallocates memory assigned for a Product and all its components
Precondition: Product must not be null
Postcondition: Memory freed
Classification: Destroyer

2.10 Provider

Description: The Provider structure has the information of a provider and it's details.

createProvider: name x description x address x city x telephoneNumber x country →
Provider
Description: creates a Provider
Precondition: strings passed must not be empty
Postcondition: A Provider is created
Classification: Constructor

destroyProvider: Provider → void
Description: Deallocates memory assigned for a Provider and all its components
Precondition: Provider must not be null
Postcondition: Memory freed
Classification: Destroyer

2.11 User

Description: The User structure has the information of a User and it's details.

createUser: name x dni x address x telephoneNumber x city x province x country x postalCode \rightarrow User

Description: creates a User

Precondition: strings passed must not be empty

Postcondition: A User is created

Classification: Constructor

destroyUser: User \rightarrow void

Description: Deallocates memory assigned for a User and all its components

Precondition: User must not be null

Postcondition: Memory freed

Classification: Destroyer

3 Library

3.1 Library

Description: The Library structure holds the information for a Library, the lists for its materials, people and loans

newLibrary: $\text{---} \rightarrow \text{Library}$
Description: Creates an empty library
Precondition: ---
Postcondition: A Library is created
Classification: Constructor

loadMaterial: $\text{library} \times \text{material} \rightarrow \text{---}$
Description: Loads materials to the library
Precondition: A library must exist
Postcondition: The material is loaded to the library
Classification: Modifier

loadPerson: $\text{library} \times \text{person} \rightarrow \text{---}$
Description: Loads a person to the library
Precondition: A library and the given person must exist
Postcondition: The person is added to the library
Classification: Modifier

loadLoan: $\text{library} \times \text{loan} \rightarrow \text{---}$
Description: Loads a loan to the Library
Precondition: A library and the given loan must exist
Postcondition: The loan is added to a list in the library
Classification: Modifier

printMaterialsList: $\text{library} \times \text{index} \rightarrow \text{---}$
Description: Prints the materials list of the library
Precondition: A library must exist
Postcondition: The list of materials is printed
Classification: Analyzer

showAllMaterials: library \rightarrow —
Description: Prints the list of all the materials
Precondition: A library must exist
Postcondition: The list of all the materials is printed
Classification: Analyzer

getAvailableMaterials: library \rightarrow —
Description: Calculates how many available materials are there
Precondition: A library must exist
Postcondition: The quantity of available is calculated
Classification: Analyzer

printPeopleList: library \rightarrow —
Description: Prints the people list of the library
Precondition: A library must exist
Postcondition: The list of people is printed
Classification: Analyzer

deleteMaterial: library x position \rightarrow —
Description: Deletes a material from the library
Precondition: A library must exist
Postcondition: The material in the position given is deleted from the library
Classification: Destroyer

freeLibrary: library \rightarrow —
Description: Frees the space of memory assigned for a Library and all its components
Precondition: A library must exist
Postcondition: The memory space is freed
Classification: Destroyer

3.2 Loan

Description: The Loan structure holds the information for a loan, the customer, the rented material, a random code, and the days it was rented

newLoan: person x material x rentDays \rightarrow Loan
Description: Creates a Loan
Precondition: The given person and material must exist
Postcondition: A Loan is created
Classification: Constructor

fee: loan \rightarrow —
Description: Fees the person who rented a material
Precondition: The loan must exist
Postcondition: A fee is emitted to the client
Classification: Modifier

showInvoice: loan \rightarrow —
Description: Prints the Invoice
Precondition: The loan must exist
Postcondition: The invoice is printed
Classification: Analyzer

freeLoan: loan \rightarrow —
Description: Frees the space of memory assigned for a loan and all its components
Precondition: The loan must exist
Postcondition: The memory space is freed
Classification: Destroyer

3.3 Material

Description: The Material structure holds the information for a material, its type, code, author, title, year, status, its cost per day, and, in case it is a book, its editorial

newMaterial: code x author x title x year x isBook x isAvailable x costPerDay \rightarrow Material
Description: Creates a Material
Precondition: —
Postcondition: A Material is created
Classification: Constructor

changeMaterialStatus: material \rightarrow —
Description: Changes the material availability status
Precondition: The given material must exist
Postcondition: The availability status of the given material is changed
Classification: Modifier

freeMaterial: material \rightarrow —
Description: Frees the space of memory related to a material and all its components
Precondition: The given material must exist
Postcondition: The space of memory is freed
Classification: Destroyer

3.4 Person

Description: The Person structure holds information about a person, such as its profession, name, surname, mail, phone number, debt, a list of materials in his possession, and, in case he is a student, its registration number, but if he is a teacher, its employee number

newPerson: name x surname x mail x phoneNumber x isStudent \rightarrow Person
Description: Creates a person
Precondition: —
Postcondition: A Person is created
Classification: Constructor

takeMaterial: person x material \rightarrow —
Description: Takes a material from the library
Precondition: The given material and person must exist
Postcondition: The person is in possession of the material
Classification: Modifier

showMyMaterials: person \rightarrow —
Description: Prints a list of materials in possession of a given person
Precondition: The given person must exist
Postcondition: The list of materials of the person is printed
Classification: Analyzer

freePerson: $\text{person} \rightarrow \text{—}$

Description: Frees the space of memory assigned to a person and all its components

Precondition: The given person must exist

Postcondition: The space of memory is freed

Classification: Destroyer

4 Hotel

4.1 Admin

Description: The Admin structure holds information about the management, such as lists of receptionists, clients, reservations, and invoices

newAdmin: $\text{---} \rightarrow \text{Admin}$
Description: Creates an Admin
Precondition: ---
Postcondition: An Admin is created
Classification: Constructor

addReceptionist: $\text{admin} \times \text{receptionist} \rightarrow \text{---}$
Description: Adds a receptionist
Precondition: The given admin and receptionist must exist
Postcondition: The receptionist is added to a list of the admin
Classification: Modifier

addClient: $\text{admin} \times \text{client} \rightarrow \text{---}$
Description: Adds a client to the system
Precondition: The given admin and client must exist
Postcondition: The client is added to a list of the admin
Classification: Modifier

addReservation: $\text{admin} \times \text{reservation} \rightarrow \text{---}$
Description: Adds a reservation to the system
Precondition: The given admin and reservation must exist
Postcondition: The reservation is added to a list of the admin
Classification: Modifier

addInvoice: $\text{admin} \times \text{invoice} \rightarrow \text{---}$
Description: Adds an invoice to the system
Precondition: The given admin and invoice must exist
Postcondition: The invoice is added to a list of the admin
Classification: Modifier

showReceptionistsList: admin \rightarrow —
Description: Prints the list of receptionists
Precondition: The given admin must exist
Postcondition: The list of receptionists is printed
Classification: Analyzer

showClientstsList: admin \rightarrow —
Description: Prints the list of clients
Precondition: The given admin must exist
Postcondition: The list of clients is printed
Classification: Analyzer

showReservationsList: admin \rightarrow —
Description: Prints the list of reservations
Precondition: The given admin must exist
Postcondition: The list of reservations is printed
Classification: Analyzer

freeAdmin: admin \rightarrow —
Description: Frees the space of memory assigned to an admin and all its components
Precondition: The given admin must exist
Postcondition: The space of memory is freed
Classification: Destroyer

4.2 Client

Description: The Client structure contains information about the client, his name, surname, ID, current debt, and, once effectuated, his invoice

newClient: name x surname x id \rightarrow Client
Description: Creates a client
Precondition: —
Postcondition: A Client is created
Classification: Constructor

payForRoom: invoice, client \rightarrow —
Description: Pays for the hotel room
Precondition: The given invoice and client must exist
Postcondition: The client debt is increased
Classification: Modifier

freeClient: client \rightarrow —
Description: Frees the space of memory assigned to a client and all its components
Precondition: The given client must exist
Postcondition: The space of memory is freed
Classification: Destroyer

4.3 Hotel

Description: The Hotel structure contains the information about the hotel, such as its name, a list of all its rooms, and both the price for the Basic and the Deluxe rooms

newHotel: — \rightarrow Hotel
Description: Creates a hotel
Precondition: —
Postcondition: A Hotel is created
Classification: Constructor

setHotelName: hotel x name \rightarrow —
Description: Sets the name for the hotel
Precondition: The given hotel must exist
Postcondition: The name of the hotel is set
Classification: Modifier

bookBasicRoom: client x hotel \rightarrow Reservation
Description: Books a basic room
Precondition: The given client and hotel must exist
Postcondition: A reservation in the hotel for a basic room is made for the client
Classification: Modifier

bookDeluxeRoom: client x hotel \rightarrow Reservation
Description: Books a deluxe room
Precondition: The given client and hotel must exist
Postcondition: A reservation in the hotel for a deluxe room is made for the client
Classification: Modifier

freeHotel: hotel \rightarrow —
Description: Frees the space of memory assigned to a hotel and all its components
Precondition: The given hotel must exist
Postcondition: The space of memory is freed
Classification: Destroyer

4.4 Invoice

Description: The Invoice structure contains information about the invoice, like its number, the name of the hotel, the name of the client, his ID, and the price to pay

newInvoice: hotelName x clientName x clientId x priceToPay \rightarrow Invoice
Description: Creates an Invoice
Precondition: —
Postcondition: An Invoice is created
Classification: Constructor

printInvoice: invoice \rightarrow —
Description: Prints an invoice
Precondition: The given invoice must exist
Postcondition: The invoice is printed
Classification: Analyzer

freeInvoice: invoice \rightarrow —
Description: Frees the space of memory assigned to an invoice and all its components
Precondition: The given invoice must exist
Postcondition: The space of memory is freed
Classification: Destroyer

4.5 Receptionist

Description: The Receptionist structure contains the information about the receptionist, such as his name, surname, and ID

newReceptionist: name x surname x id \rightarrow Receptionist
Description: Creates a Receptionist
Precondition: —
Postcondition: A Receptionist is created
Classification: Constructor

giveInfo: hotel \rightarrow —
Description: Gives information about the room availability status and its price per night
Precondition: The given hotel must exist
Postcondition: The information about the room availability status and its price per night is printed
Classification: Analyzer

freeReceptionist: receptionist \rightarrow —
Description: Frees the space of memory assigned to a receptionist and all its components
Precondition: The given receptionist must exist
Postcondition: The space of memory is freed
Classification: Destroyer

4.6 Reservation

Description: The Reservation structure contains information about the reservation, specifically, the surname of the client, his ID, and the booked room number

newReservation: clientSurname x clientId x roomNumber \rightarrow Reservation
Description: Creates a Reservation
Precondition: —
Postcondition: A Reservation is created
Classification: Constructor

freeReservation: reservation \rightarrow —
Description: Frees the space of memory assigned to a reservation and all its components
Precondition: The given reservation must exist
Postcondition: The space of memory is freed
Classification: Destroyer

4.7 Room

Description: The Room structure contains information about the room, its number, type, its availability status, price, and, if it is occupied, the Client in it

newRoom: number x type \rightarrow Room
Description: Creates a Room
Precondition: —
Postcondition: A Room is created
Classification: Constructor

accommodateClient: room, client \rightarrow —
Description: Accommodates a client in the room he previously booked
Precondition: The given room and client must exist
Postcondition: The client is in the room he had booked
Classification: Modifier

freeRoom: room \rightarrow —
Description: Frees the space of memory assigned to a room and all its components
Precondition: The given room must exist
Postcondition: The space of memory is freed
Classification: Destroyer

5 Movie Rental

5.1 Admin

Description: The Admin structure contains information related to an admin, like his name, surname, ID, and he also has a list of all the licenses

newAdmin: name x surname x id \rightarrow Admin

Description: Creates an Admin

Precondition: —

Postcondition: An Admin is created

Classification: Constructor

registerClient: admin x client x clientsDatabase \rightarrow —

Description: Registers a new client, and creates a license for him

Precondition: The admin, client, and clientsDatabase, must exist

Postcondition: The client is registered, and a license is created for him

Classification: Modifier

registerMovie: movie x moviesDatabase \rightarrow —

Description: Registers a new movie

Precondition: The movie and moviesDatabase must exist

Postcondition: The movie is registered

Classification: Modifier

freeAdmin: admin \rightarrow —

Description: Frees the space of memory assigned to an admin and all its components

Precondition: The given admin must exist

Postcondition: The space of memory is freed

Classification: Destroyer

5.2 Client

Description: The Client structure contains information about a client, specifically, his name, surname, ID, current debt, and a list of movies in his possession

newClient: name x surname x id \rightarrow Client

Description: Creates a Client

Precondition: —

Postcondition: A Client is created

Classification: Constructor

rentMovie: client x days x moviesDatabase x position x excess \rightarrow —

Description: Rents a movie

Precondition: The given client, moviesDatabase, and excess, must exist

Postcondition: The client is in possession of a new movie

Classification: Modifier

showMyMovies: client \rightarrow —

Description: Prints a list of movies in the clients possession

Precondition: The given client must exist

Postcondition: The list of movies in the clients possession is printed

Classification: Analyzer

showTotalDebt: client \rightarrow —

Description: Shows the total debt of a client

Precondition: The given client must exist

Postcondition: The total debt of a given client is shown

Classification: Analyzer

freeClient: client \rightarrow —

Description: Frees the space of memory assigned to a client and all its components

Precondition: The given client must exist

Postcondition: The space of memory is freed

Classification: Destroyer

5.3 ClientsDatabase

Description: The ClientsDatabase struct contains a list of registered clients

newClientsDatabase: $\text{---} \rightarrow \text{ClientsDatabase}$
Description: Creates a ClientsDatabase
Precondition: ---
Postcondition: A ClientsDatabase is created
Classification: Constructor

addClient: $\text{clientsDatabase} \times \text{client} \rightarrow \text{---}$
Description: Adds a client to the database
Precondition: The given clientsDatabase and client must exist
Postcondition: The client is in the database
Classification: Modifier

showClientsList: $\text{clientsDatabase} \rightarrow \text{---}$
Description: Prints a list of clients that are in the database
Precondition: The given clientsDatabase must exist
Postcondition: The list of clients that are in the database is printed
Classification: Analyzer

freeClientsDatabase: $\text{clientsDatabase} \rightarrow \text{---}$
Description: Frees the space of memory assigned to a clientsDatabase and all its components
Precondition: The given clientsDatabase must exist
Postcondition: The space of memory is freed
Classification: Destroyer

5.4 Excess

Description: The Excess struct contains a list of the rented movies, and the total income

newExcess: $\text{---} \rightarrow \text{Excess}$
Description: Creates an Excess
Precondition: ---
Postcondition: An Excess is created
Classification: Constructor

showRentedMoviesList: excess \rightarrow —

Description: Prints a list of all the rented movies

Precondition: The given excess must exist

Postcondition: The list of all the rented movies is printed

Classification: Analyzer

showTotalIncome: excess \rightarrow —

Description: Shows the total income

Precondition: The given excess must exist

Postcondition: The total income is shown

Classification: Analyzer

freeExcess: excess \rightarrow —

Description: Frees the space of memory assigned to an excess and all its components

Precondition: The given excess must exist

Postcondition: The space of memory is freed

Classification: Destroyer

5.5 License

Description: The License struct contains an ID

newLicense: id \rightarrow License

Description: Creates a License

Precondition: —

Postcondition: A License is created

Classification: Constructor

freeLicense: license \rightarrow —

Description: Frees the space of memory assigned to a license and all its components

Precondition: The given license must exist

Postcondition: The space of memory is freed

Classification: Destroyer

5.6 Movie

Description: The Movie struct contains information about a movie, its name and its price

newMovie: name x price \rightarrow Movie
Description: Creates a Movie with a name and a price
Precondition: —
Postcondition: A Movie is created
Classification: Constructor

freeMovie: movie \rightarrow —
Description: Frees the space of memory assigned to a movie and all its components
Precondition: The given movie must exist
Postcondition: The space of memory is freed
Classification: Destroyer

5.7 MoviesDatabase

Description: The MoviesDatabase struct contains a list of registered movies

newMoviesDatabase: — \rightarrow MoviesDatabase
Description: Creates a MoviesDatabase
Precondition: —
Postcondition: A MoviesDatabase is created
Classification: Constructor

addMovie: moviesDatabase x movie \rightarrow —
Description: Adds a movie to the database
Precondition: The given moviesDatabase and movie must exist
Postcondition: The movie is in the database
Classification: Modifier

showMoviesList: moviesDatabase \rightarrow —
Description: Prints the list of all the movies available in the system
Precondition: The given moviesDatabase must exist
Postcondition: The list of all the movies available in the system is printed
Classification: Analyzer

removeMovie: moviesDatabase x position \rightarrow —

Description: Removes the movie in the given position from the database

Precondition: The given moviesDatabase and position must exist

Postcondition: The movie is deleted from the database

Classification: Destroyer

freeMoviesDatabase: moviesDatabase \rightarrow —

Description: Frees the space of memory assigned to a MoviesDatabase and all its components

Precondition: The given MoviesDatabase must exist

Postcondition: The space of memory is freed

Classification: Destroyer