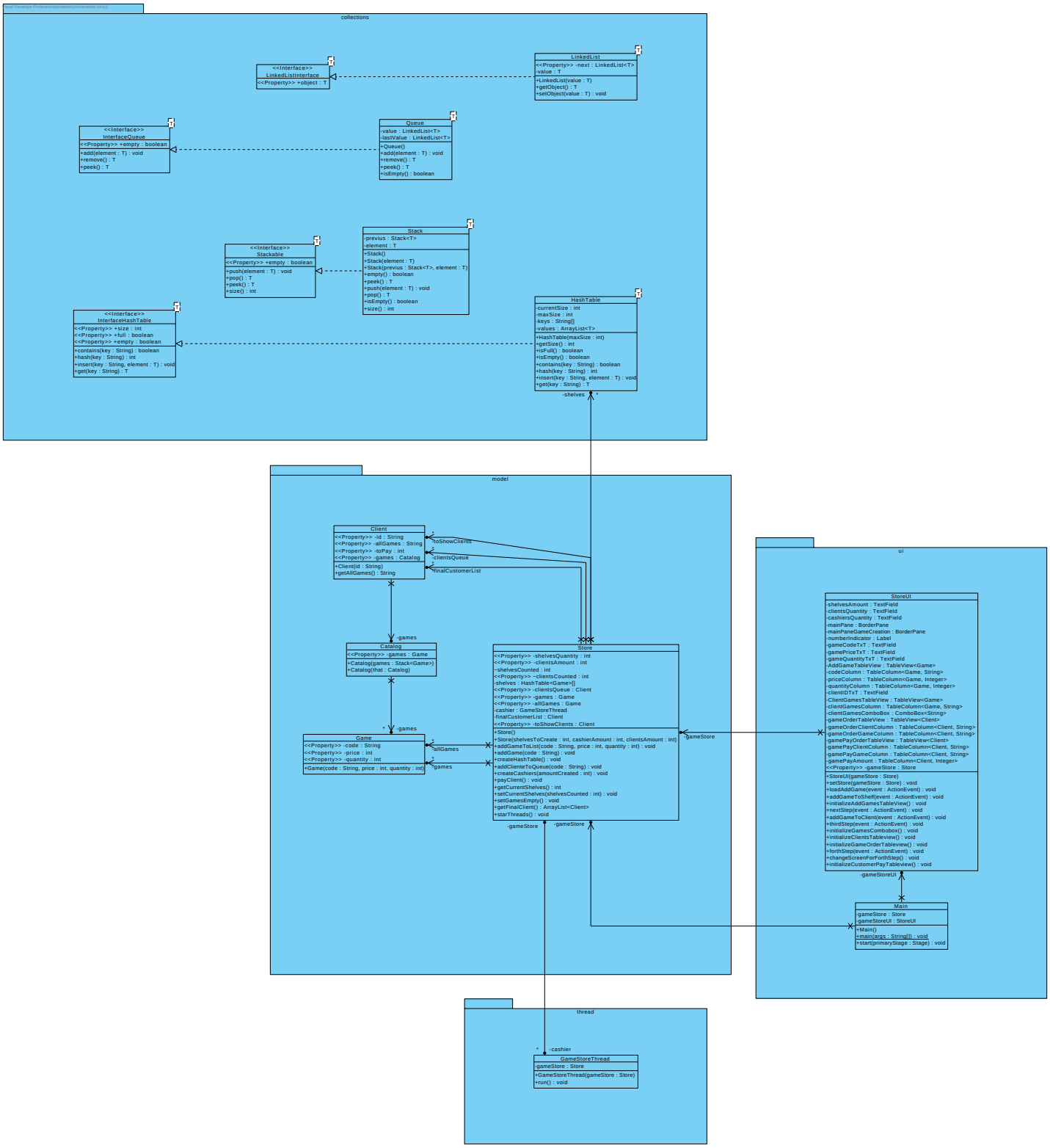
collections
<<Interface>>
LinkedListInterface
<<Property>> +object : T
LinkedList
<<Property>> -next : LinkedList<T>
-value : T
+LinkedList(value : T)
+getObject() : T
+setObject(value : T) : void
<<Interface>>
InterfaceQueue
<<Property>> +empty : boolean
+add(element : T) : void
+remove() : T
+peek() : T
Queue
-value : LinkedList<T>
-lastValue : LinkedList<T>
+Queue()
+add(element : T) : void
+remove() : T
+peek() : T
+isEmpty() : boolean
<<Interface>>
Stackable
<<Property>> +empty : boolean
+push(element : T) : void
+pop() : T
+peek() : T
+size() : int
Stack
-previus : Stack<T>
-element : T
+Stack()
+Stack(element : T)
+Stack(previus : Stack<T>, element : T)
+empty() : boolean
+peek() : T
+push(element : T) : void
+pop() : T
+isEmpty() : boolean
+size() : int
<<Interface>>
InterfaceHashTable
<<Property>> +size : int
<<Property>> +full : boolean
<<Property>> +empty : boolean
+contains(key : String) : boolean
+hash(key : String) : int
+insert(key : String, element : T) : void
+get(key : String) : T
HashTable
-currentSize : int
-maxSize : int
-keys : String[]
-values : ArrayList<T>
+HashTable(maxSize : int)
+getSize() : int
+isFull() : boolean
+isEmpty() : boolean
+contains(key : String) : boolean
+hash(key : String) : int
+insert(key : String, element : T) : void
+get(key : String) : T
-shelves
*
model
Client
<<Property>> -id : String
<<Property>> -allGames : String
<<Property>> -toPay : int
<<Property>> -games : Catalog
+Client(id : String)
+getAllGames() : String
-toShowClients
-clientsQueue
-finalCustomerList
-games
Catalog
<<Property>> -games : Game
+Catalog(games : Stack<Game>)
+Catalog(that : Catalog)
*
-games
Game
<<Property>> -code : String
<<Property>> -price : int
<<Property>> -quantity : int
+Game(code : String, price : int, quantity : int)
-allGames
-games
Store
<<Property>> -shelvesQuantity : int
<<Property>> -clientsAmount : int
~shelvesCounted : int
<<Property>> ~clientsCounted : int
-shelves : HashTable<Game>[]
<<Property>> -clientsQueue : Client
<<Property>> -games : Game
<<Property>> -allGames : Game
-cashier : GameStoreThread
-finalCustomerList : Client
<<Property>> -toShowClients : Client
+Store()
+Store(shelvesToCreate : int, cashierAmount : int, clientsAmount : int)
+addGameToList(code : String, price : int, quantity : int) : void
+addGame(code : String) : void
+createHashTable() : void
+addClienteToQueue(code : String) : void
+createCashiers(amountCreated : int) : void
+payClient() : void
+getCurrentShelves() : int
+setCurrentShelves(shelvesCounted : int) : void
+setGamesEmpty() : void
+getFinalClient() : ArrayList<Client>
+starThreads() : void
-gameStore
-gameStore
-gameStore
ui
StoreUI
-shelvesAmount : TextField
-clientsQuantity : TextField
-cashiersQuantity : TextField
-mainPane : BorderPane
-mainPaneGameCreation : BorderPane
-numberIndicator : Label
-gameCodeTxT : TextField
-gamePriceTxT : TextField
-gameQuantityTxT : TextField
-AddGameTableView : TableView<Game>
-codeColumn : TableColumn<Game, String>
-priceColumn : TableColumn<Game, Integer>
-quantityColumn : TableColumn<Game, Integer>
-clientIDTxT : TextField
-ClientGamesTableView : TableView<Game>
-clientGamesColumn : TableColumn<Game, String>
-clientGamesComboBox : ComboBox<String>
-gameOrderTableView : TableView<Client>
-gameOrderClientColumn : TableColumn<Client, String>
-gameOrderGameColumn : TableColumn<Client, String>
-gamePayOrderTableView : TableView<Client>
-gamePayClientColumn : TableColumn<Client, String>
-gamePayGameColumn : TableColumn<Client, String>
-gamePayAmount : TableColumn<Client, Integer>
<<Property>> -gameStore : Store
+StoreUI(gameStore : Store)
+setStore(gameStore : Store) : void
+loadAddGame(event : ActionEvent) : void
+addGameToShelf(event : ActionEvent) : void
+initializeAddGamesTableView() : void
+nextStep(event : ActionEvent) : void
+addGameToClient(event : ActionEvent) : void
+thirdStep(event : ActionEvent) : void
+initializeGamesComboBox() : void
+initializeClientsTableview() : void
+initializeGameOrderTableview() : void
+forthStep(event : ActionEvent) : void
+changeScreenForForthStep() : void
+initializeCustomerPayTableview() : void
-gameStore
-gameStoreUI
Main
-gameStore : Store
-gameStoreUI : StoreUI
+Main()
+main(args : String[]) : void
+start(primaryStage : Stage) : void
thread
*
-cashier
GameStoreThread
-gameStore : Store
+GameStoreThread(gameStore : Store)
+run() : void