TEST CASE DESIGN

Scenarios:

Name	Class	Scenary
setUpEscenary1	Game	Game with correctly attributes
setUpEscenary1	Shelve	Shelve with correctly attributes
setUpEscenary1	Costumer	Costumer with correctly attributes
setUpEscenary1	Hash	Empty
setUpEscenary2	Hash	Hash to insert in one position are occupied. Hash Table is full.
setUpEscenary1	Queue	Empty
setUpEscenary2	Queue	An object null
setUpEscenary1	Stack	Empty
setUpEscenary2	Stack	An object null

Test Objective: Verify that a game is correctly added to the Game store

Class	Method	Scenary	Input	Output
GameStore	addGame	setupScenary 1	Name: Call of Duty review: Weapons and Fun price: 30000 Shelve name: A Amount: 4	4 games with the name “Call of duty” and price 30000, to the shelve name “A” was added successfully

Test Objective: Verify that a shelve is correctly added to the Game store

Class	Method	Scenary	Input	Output
GameStore	addShelve	setupScenary 1	Shelve name: A Size Shelve slots: 4	The shelve “A” with 4 slots are added successfully to the game store.

Test Objective: Verify that a costumer is correctly added to the Game store

Class	Method	Scenary	Input	Output
GameStore	addCostumer	setupScenary 1	Id: 12345 Name: Pepito Code: 111 Wishlist: Call of duty, clash royale, clash of clans	The costumer “Pepito” with id 12345, code 111 and the wish list has entered to the game store.

Test Objective: Verify if one shelve doesn’t exist in the game store

Class	Method	Scenary	Input	Output
GameStore	repeatedVerifyShelve	setupScenary 1	Shelve name: A	The shelve name “A” already exist in the game store,

Test Objective: Check if a customer is repeat in the game store

Class	Method	Scenary	Input	Output
GameStore	repeatedVerifyCostumer	setupScenary 1	Name: Pepito	The costumer name “Pepito” already exist in the game store,

Test Objective: Get the amount of the cashiers in the game store

Class	Method	Scenary	Input	Output
GameStore	getCashiers	setupScenary 1	None	The amount of the cashiers in the game store are “0”.

Test Objective: Get the amount of the cashiers in the game store

Class	Method	Scenary	Input	Output
GameStore	searchCostumer	setupScenary 1	String: Pepito	The costumer “Pepito” was founded.

Test Objective: Get the values on the hash table

Class	Method	Scenary	Input	Output
Hash	Search()	setupScenary 1	Game game1= new Game(1,"Clash of clans ",”Fun”,40000,"A",10);	Return the values of the game1.
			Game game2 =new Game(2,"Clash Royale",”Fun”,S,"A",20);	Return values of the game2.

Test Objective: Get the values on the hash table

Class	Method	Scenary	Input	Output
Hash	Search()	setupScenary 2	Game game1= new Game(1,"Clash of clans ","Fun",40000,"A",10); Game game2 =new Game(2,"Clash Royale","Fun",S,"A",20);	Return the values of the game1. Return values of the game2.

Test Objective: Add element to the hash table

Class	Method	Scenary	Input	Output
Hash	addElement()	setupScenary 1	Game game1= new Game(1,"Clash of clans ",”Fun”,40000,"A",10); Game game2 =new Game(2,"Clash Royale",”Fun”,S,"A",20);	The game1 was added successfully. The game2 was added successfully.

Test Objective: Add element to the hash table

Class	Method	Scenary	Input	Output
Hash	addElement()	setupScenary 2	Game game1= new Game(1,"Clash of clans ",”Fun”,40000,"A",10); Game game2 =new Game(2,"Clash Royale",”Fun”,S,"A",20);	The position to insert the game was occupied. But with open addressing can found a new slot to insert. The game1 was insert successfully. Cause the hash table are full, you cannot add more elements, but if u use chaining, you can create a linked list to add more elements

Test Objective: Remove element to the hash table

Class	Method	Scenary	Input	Output
Hash	removeElement()	setupScenary 1	Game game1= new Game(1,"Clash of clans ","Fun",40000,"A",10); Game game2 =new Game(2,"Clash Royale","Fun",S,"A",20);	The game1 was deleted successfully. The game2 was deleted successfully.

Test Objective: Remove the values on the hash table

Class	Method	Scenary	Input	Output
Hash	removeElement()	setupScenary 2	Game game1= new Game(1,"Clash of clans ","Fun",40000,"A",10); Game game2 =new Game(2,"Clash Royale","Fun",S,"A",20);	The game1 was deleted successfully. If we have the method chaining to resolve collision. The game2 was deleted successfully.

Test Objective: Validate if the hast table are full.

Class	Method	Scenary	Input	Output
Hash	isFull()	setupScenary 2	None	The hash table is not full. The hash table is full.

Test Objective: Validate if the hast table are empty.

Class	Method	Scenary	Input	Output
Hash	isEmpty()	setupScenary 1	None	The hash table is empty.

Test Objective: Obtain the size of the hash table.

Class	Method	Scenary	Input	Output
Hash	getSize()	setupScenary 1	None	The size of the hash table is 0. The hast is empty.

Test Objective: Obtain the size of the hash table.

Class	Method	Scenary	Input	Output
Hash	getSize()	setupScenary 2	Game game1= new Game(1,"Clash of clans ",”Fun”,40000,"A",10); Game game2 =new Game(2,"Clash Royale",”Fun”,S,"A",20);	The size of the hash table is 1. The size of the hash table is 2.

Test Objective: Add one value to the queue.

Class	Method	Scenary	Input	Output
Queue	enqueue()	setupScenary 1	Costumer costumer1 = new Costumer(1,"Juanita",”game2, game3”);	The costumer1 was added successfully.
			Costumer costumer2= new Costumer (2,"Pedro",”game1”);	The costumer2 was added successfully.

Test Objective: Add one value to the queue.

Class	Method	Scenary	Input	Output
Queue	enqueue()	setupScenary 2	Costumer costumer1 = null;	The costumer1 could not been added.

Test Objective: Delete one value to the queue

Class	Method	Scenary	Input	Output
Queue	dequeue()	setupScenary 1	costumer1 costumer2	The costumer1 was deleted successfully. Return costumer1. The costumer2 was deleted successfully. Return costumer2.

Test Objective: Delete one value to the queue

Class	Method	Scenary	Input	Output
Queue	dequeue()	setupScenary 2	Null	The object is null, the costumer couldn't be deleted. Return null.

Test Objective: Validate if the queue is empty.

Class	Method	Scenary	Input	Output
Queue	isEmpty()	setupScenary 1	Costumer costumer1 = new Costumer(1,"Juanita",”game2, game3”);	The queue is not empty.

Test Objective: Validate if the queue is empty.

Class	Method	Scenary	Input	Output
Queue	isEmpty()	setupScenary 2	None	The queue is empty.

Test Objective: Obtain the data of the element in one queue

Class	Method	Scenary	Input	Output
Queue	front()	setupScenary 1	Costumer costumer1 = new Costumer(1,"Juanita",”game2, game3”);	Return the value of the costumer1.

Test Objective: Obtain the data of the element in one queue

Class	Method	Scenary	Input	Output
Queue	front()	setupScenary 2	null	Not it´s possible return one object null.

Test Objective: Obtain the last element in the queue

Class	Method	Scenary	Input	Output
Queue	getLast()	setupScenary 1	Costumer costumer1 = new Costumer(1,"Juanita",”game2, game3”); null	The last element in the queue is null

Test Objective: Return the size of the queue

Class	Method	Scenary	Input	Output
Queue	size()	setupScenary 1	Costumer costumer1 = new Costumer(1,"Juanita",”game2, game3”); Costumer costumer2= new Costumer (2,"Pedro",”game1”);	The size of the queue is 1. The size of the queue is 2.

Test Objective: Return the size of the queue

Class	Method	Scenary	Input	Output
Queue	size()	setupScenary 2	null Costumer costumer2= new Costumer (2,"Pedro",”game1”);	Cause the first object is null, the size is 1.

Test Objective: Search one element in the queue.

Class	Method	Scenary	Input	Output
Queue	search()	setupScenary 1	Costumer costumer1 = new Costumer(1,"Juanita",”game2 , game3”); Costumer costumer2= new Costumer (2,"Pedro",”game1”);	Return the values of the game1. Return values of the game2.

Test Objective: Search one element in the queue.

Class	Method	Scenary	Input	Output
Queue	search()	setupScenary 2	null	Not it´s possible return one object null.
			Costumer costumer2= new Costumer (2,"Pedro",”game1”);	Return values of the game2.

Test Objective: Clear all the elements in the queue

.

Class	Method	Scenary	Input	Output
Queue	clear()	setupScenary 1	Costumer costumer1 = new Costumer(1,"Juanita",”game2, game3”); Costumer costumer2= new Costumer (2,"Pedro",”game1”);	The queue now is empty.

Test Objective: Clear all the elements in the queue.

Class	Method	Scenary	Input	Output
Queue	clear()	setupScenary 2	null	The queue now is empty.

Test Objective: Convert the queue in one array

Class	Method	Scenary	Input	Output
Queue	convertQueueToArray()	setupScenary 1	Costumer costumer1 = new Costumer(1,"Juanita",”game2, game3”); Costumer costumer2= new Costumer (2,"Pedro",”game1”);	The costumer1 was added to array successfully. The costumer1 was added to array successfully.

Test Objective: Add one value to the stack.

Class	Method	Scenary	Input	Output
Stack	push()	setupScenary 1	Game game1= new Game(1,"Clash of clans ","Fun",40000,"A",10); Game game2 =new Game(2,"Clash Royale","Fun",20000,"A",20);	The game1 was added successfully to the stack. The game2 was added successfully to the stack.

Test Objective: Add one value to the stack.

Class	Method	Scenary	Input	Output
Stack	push()	setupScenary 2	null Game game2 =new Game(2,"Clash Royale","Fun",20000,"A",20);	The game1 couldn't be added because is null. The game2 was added successfully to the stack.

Test Objective: Delete one value to the stack

Class	Method	Scenary	Input	Output
Stack	pop()	setupScenary 1	game1 game2	The game1 was deleted successfully. Return game1. The game2 was deleted successfully. Return game2.

Test Objective: Delete one value to the stack

Class	Method	Scenary	Input	Output
Stack	pop()	setupScenary 2	null game2	The object is null, the costumer couldn't be deleted. Return null. The game2 was deleted successfully. Return game2.

Test Objective: Validate if the stack is empty.

Class	Method	Scenary	Input	Output
Stack	isEmpty()	setupScenary 1	Null	The stack is empty.

Test Objective: Validate if the stack is empty.

Class	Method	Scenary	Input	Output
Stack	isEmpty()	setupScenary 1	Game game1= new Game(1,"Clash of clans ","Fun",40000,"A",10);	The stack is not empty.

Test Objective: Obtain the data of the element in one stack.

Class	Method	Scenary	Input	Output
Queque	top()	setupScenary 1	Game game1= new Game(1,"Clash of clans ",”Fun”,40000,"A",10); Game game2 =new Game(2,"Clash Royale",”Fun”,20000,"A",20);	Return the value of the game2.

Test Objective: Obtain the data of the element in one stack

Class	Method	Scenary	Input	Output
Stack	top()	setupScenary 2	null	Not it´s possible return one object null.

Test Objective: Obtain the size of the stack

Class	Method	Scenary	Input	Output
Stack	size()	setupScenary 1	Game game1= new Game(1,"Clash of clans ","Fun",40000,"A",10); Game game2 =new Game(2,"Clash Royale","Fun",20000,"A",20);	The size of the stack is 2.

Test Objective: Return the size of the stack

Class	Method	Scenary	Input	Output
Stack	size()	setupScenary 2	null	Cause the first object is null, the size is 0.

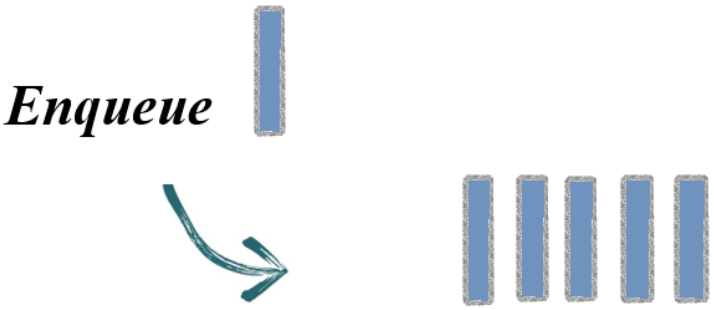
Test Objective: Return a String with the current information of the stack

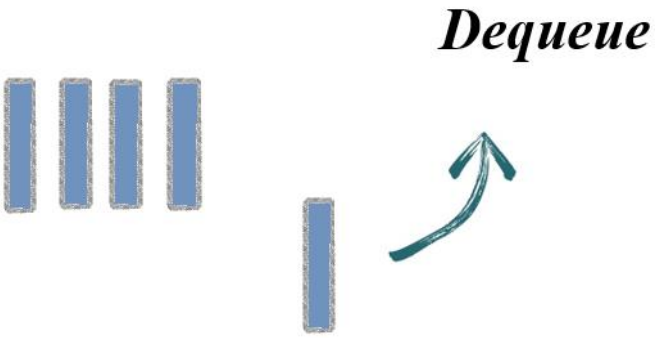
Class	Method	Scenary	Input	Output
Stack	getInfo()	setupScenary 1	Game game1= new Game(1,"Clash of clans ","Fun",40000,"A",10); Game game2 =new Game(2,"Clash Royale","Fun",20000,"A",20);	2-1

Test Objective: Return a String with the current information of the stack

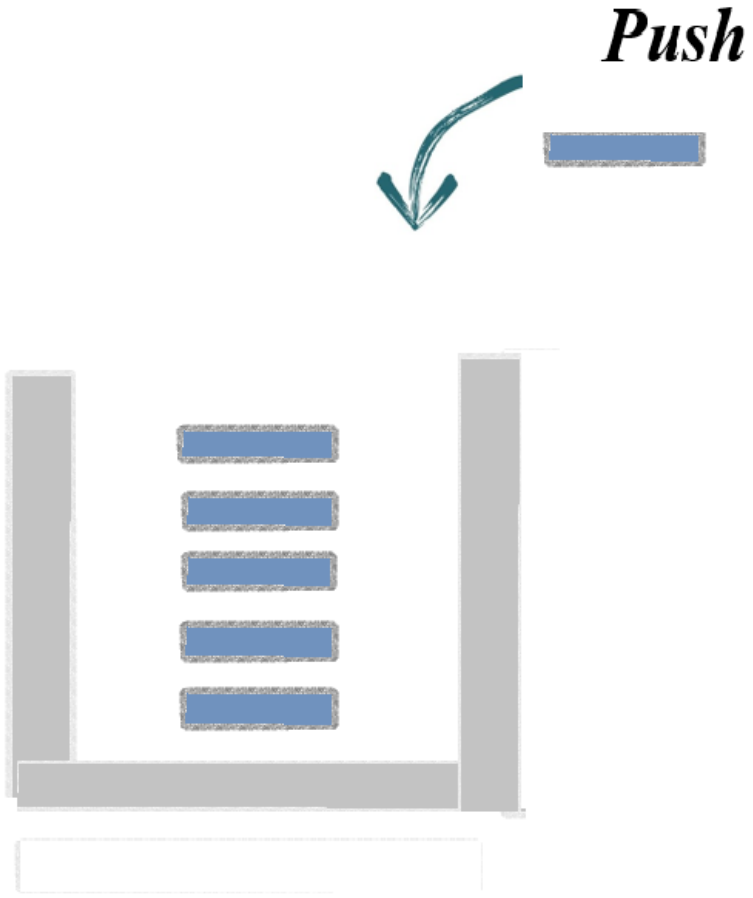
Class	Method	Scenary	Input	Output
Stack	getInfo()	setupScenary 1	NULL	""

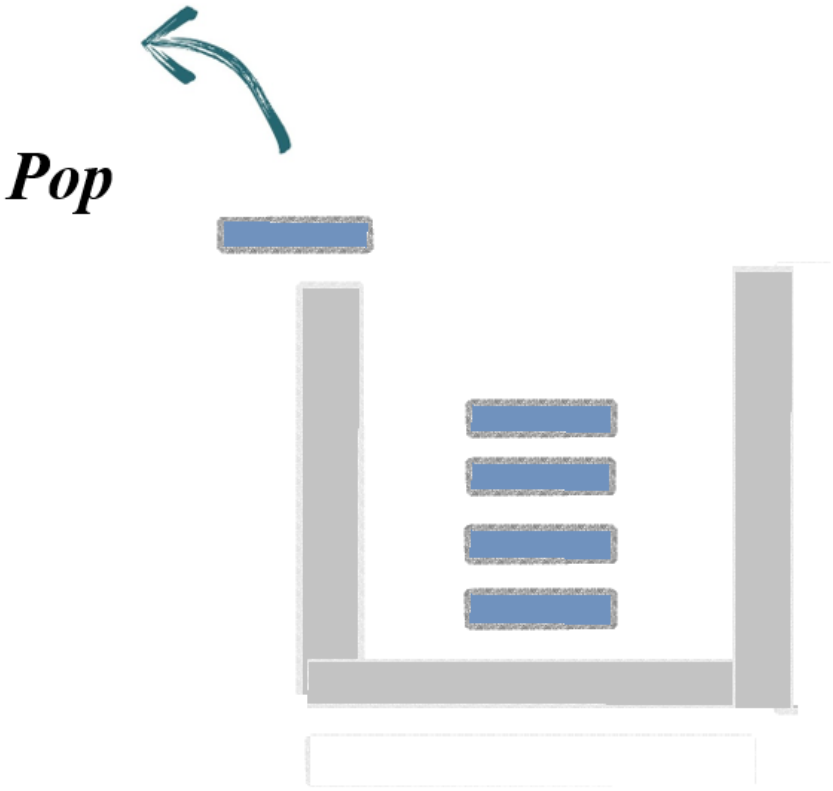
Costumers in Queue





Games in Stack





Hash as a shelve