|  |  |  |
| --- | --- | --- |
| **TAD Stack** | | |
| A stack is a list of elements in which the last object in is the first out | | |
| *inv*:   * Each new element is added to the top of the stack * The last element added is the firts to be removed | | |
| create |  | * Stack |
| push | Stack x element | * Stack |
| pop | Stack | * Stack |
| peek | Stack | * Stack |
| isEmpty | Stack | * boolean |

|  |
| --- |
| create |
| Constructor |
| Build a new stack without elements |
| precondition: |
| postcondition: A new empty stack is created |

|  |
| --- |
| push |
| Modifier |
| Add an element to a stack |
| precondition: There must be a stack previously created |
| postcondition: A new element is add/save in the stack |

|  |
| --- |
| pop |
| Modifier |
| Deleted the element at the top of the stack |
| precondition: There must be a stack previously created |
| postcondition: The top element is deleted |

|  |
| --- |
| peek |
| Analyzer |
| Show the top element of the stack without modify or deleted it |
| precondition: There must be a stack previously created |
| postcondition: The top element is returned |

|  |
| --- |
| isEmpty |
| Analyzer |
| Show if the stack contains elements or not |
| precondition: There must be a stack previously created |
| postcondition: The stated of the stack is show |

|  |  |  |
| --- | --- | --- |
| **TAD Queue** | | |
| A queue is a list of elements in which the first object in is the first out | | |
| *inv*:   * Each new element is added to the end of the queue * The first element added is the first to be removed | | |
| create |  | * Queue |
| enqueue | Queue x element | * Queue |
| dequeue | Queue | * Queue |
| peek | Queue | * Queue |
| isEmpty | Queue | * boolean |

|  |
| --- |
| create |
| Constructor |
| Build a new queue without elements |
| precondition: |
| postcondition: A new empty queue is created |

|  |
| --- |
| enqueue |
| Modifier |
| Add an element to the end of a queue |
| precondition: There must be a queue previously created |
| postcondition: A new element is added in the queue |

|  |
| --- |
| dequeue |
| Modifier |
| Deleted the firts element of the queue |
| precondition: There must be a queue previously created |
| postcondition: The firts element is deleted |

|  |
| --- |
| peek |
| Analyzer |
| Show the firts element of the queue without modify or deleted it |
| precondition: There must be a queue previously created |
| postcondition: The firts element is returned |

|  |
| --- |
| isEmpty |
| Analyzer |
| Show if the queue contains elements or not |
| precondition: There must be a queue previously created |
| postcondition: The stated of the queue is show |

|  |  |  |
| --- | --- | --- |
| **TAD Hash Table** | | |
| It is a structure where an ***n*** number of elements is stored that needs to be added and searched efficiently. Each element has a key and a value. | | |
| *inv*:   * An element must be added and searched and deleted using just the key. | | |
| create |  | * HashTable |
| hash | HashTable x key | * HashTable |
| insert | HashTable x key x element | * HashTable |
| search | HashTable x key | * element |
| deleted | HashTable x key | * element |
| isEmpty | HashTable | * boolean |

|  |
| --- |
| create |
| Constructor |
| Build a new hash table without elements |
| precondition: |
| postcondition: A new empty hash table is created |

|  |
| --- |
| hash |
| Modifier |
| Transforms the key in a position of the hash table |
| precondition: There must be a hash table previously created |
| postcondition: A position is determined |

|  |
| --- |
| insert |
| Modifier |
| Add a value in the hash table in the position determined for the hash function |
| precondition: There must be a hash table previously created |
| postcondition: A new value is added on the hash table |

|  |
| --- |
| search |
| Analyzer |
| Found an element in the hash table using the key |
| precondition: There must be a hash table previously created |
| postcondition: Return the element in the position determined for the hash function on the hash table |

|  |
| --- |
| deleted |
| Modifier |
| Deleted an element in the position determined for the hash function of the hash table |
| precondition: There must be a hash table previously created |
| postcondition: The element is deleted |

|  |  |  |
| --- | --- | --- |
| **TAD Priority Queue** | | |
| It priority queue is a list of elements in which the first object in is the first out, but, the elements in the priority queue are organized according to the level of priority they have. | | |
| *inv*:   * The highest priority element is the smallest if it is a minimum priority queue or the biggest if it is a maximum priority queue. | | |
| create |  | * PriorityQueue |
| builHeap | PriorityQueue | * PriorityQueue |
| heapify | PriorityQueue x element | * PriorityQueue |
| insert | PriorityQueue x element | * PriorityQueue |
| maximun | PriorityQueue | * element |
| deleted | PriorityQueue | * element |
| isEmpty | PriorityQueue | * boolean |

|  |
| --- |
| create |
| Constructor |
| Build a new hash table without elements |
| precondition: |
| postcondition: A new empty hash table is created |

|  |
| --- |
| hash |
| Modifier |
| Transforms the key in a position of the hash table |
| precondition: There must be a hash table previously created |
| postcondition: A position is determined |

|  |
| --- |
| insert |
| Modifier |
| Add a value in the hash table in the position determined for the hash function |
| precondition: There must be a hash table previously created |
| postcondition: A new value is added on the hash table |

|  |
| --- |
| search |
| Analyzer |
| Found an element in the hash table using the key |
| precondition: There must be a hash table previously created |
| postcondition: Return the element in the position determined for the hash function on the hash table |

|  |
| --- |
| deleted |
| Modifier |
| Deleted an element in the position determined for the hash function of the hash table |
| precondition: There must be a hash table previously created |
| postcondition: The element is deleted |