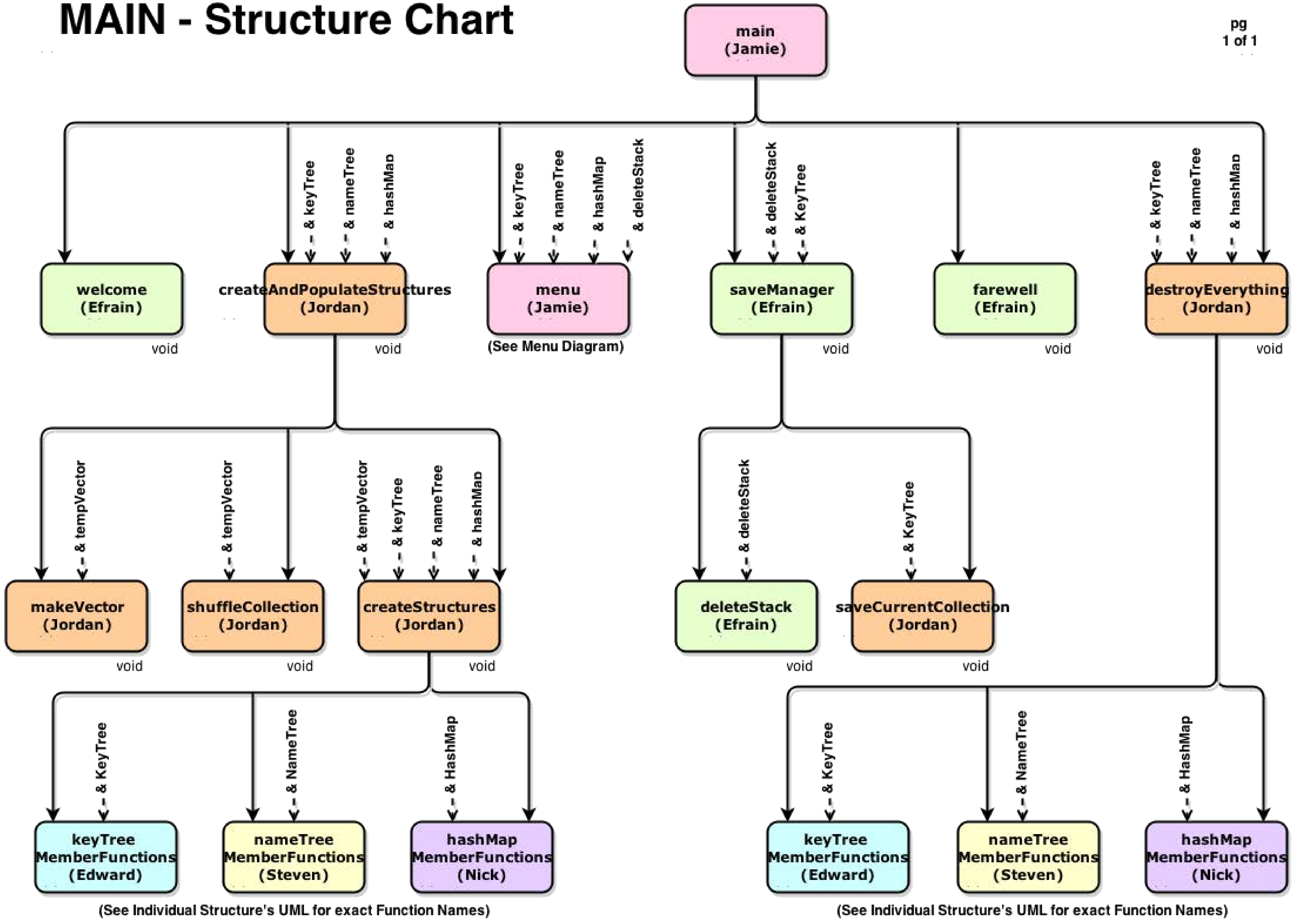
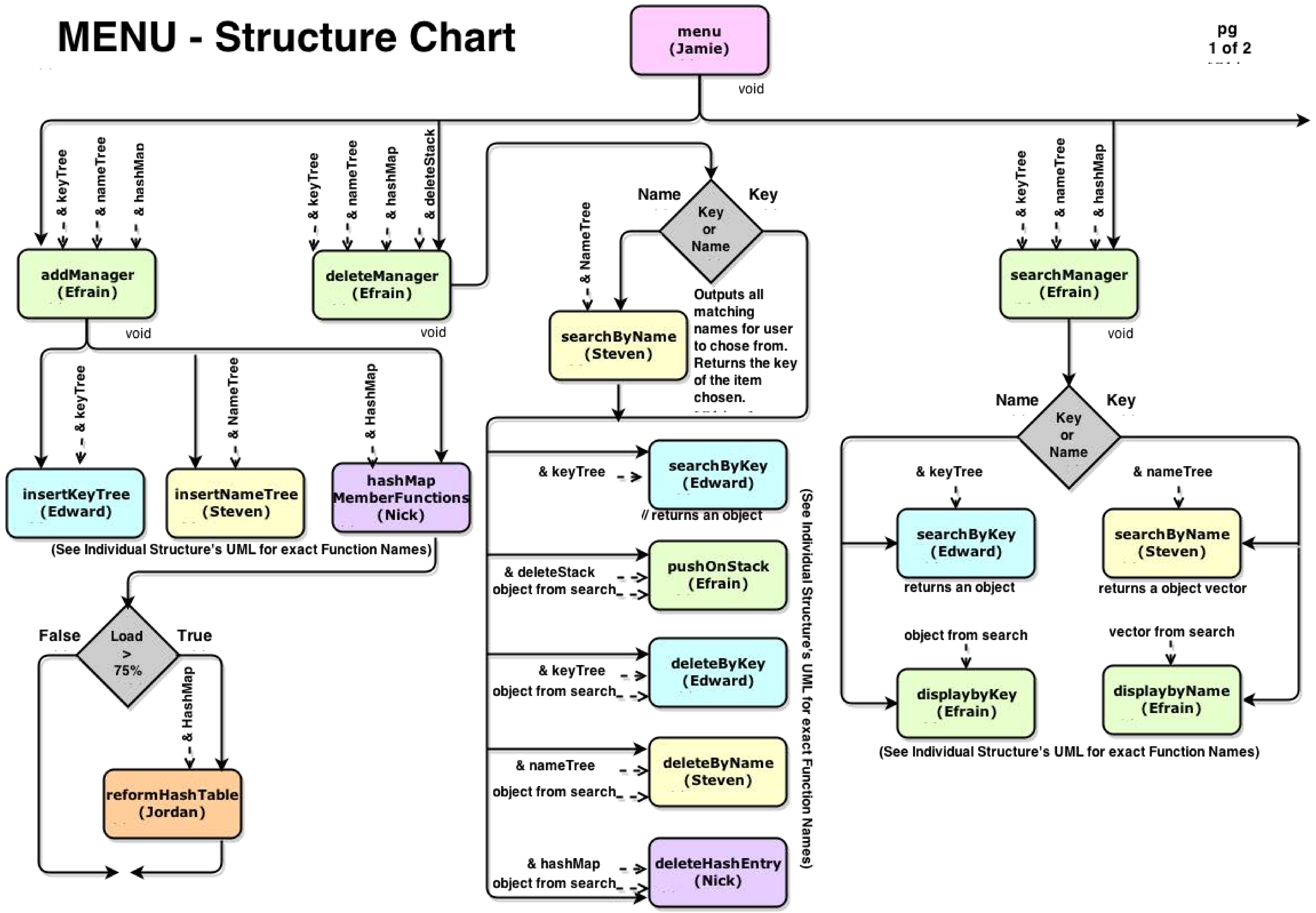
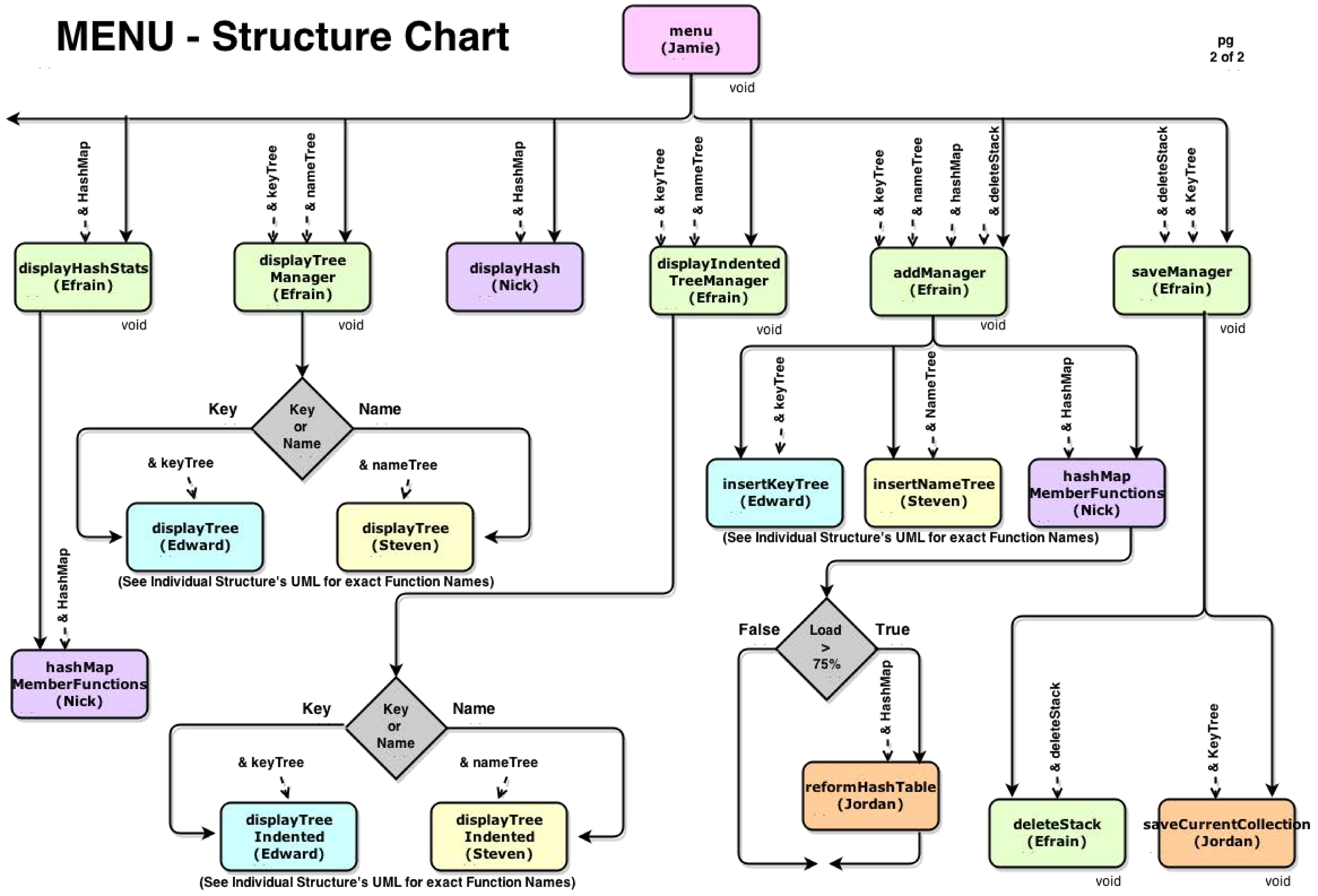
MAIN - Structure Chart



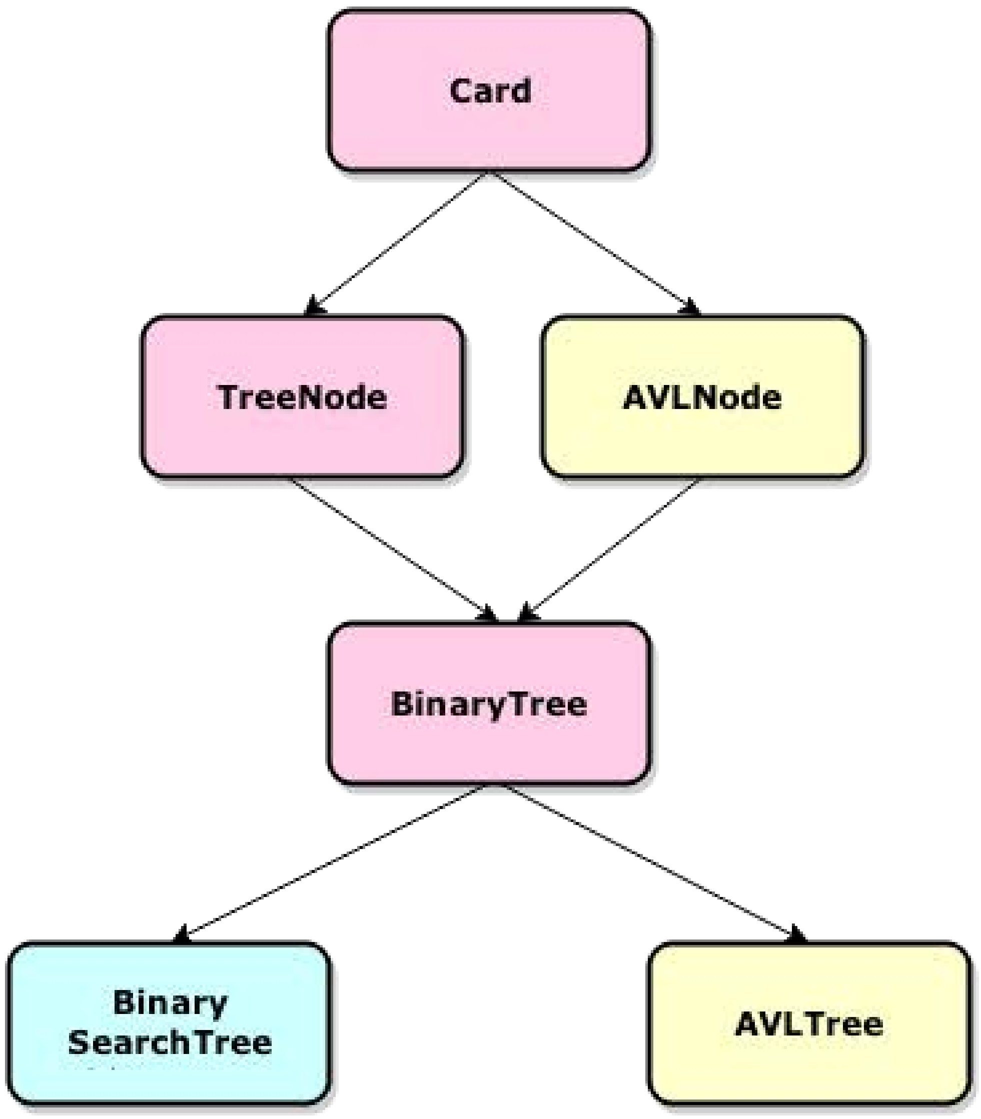
MENU - Structure Chart

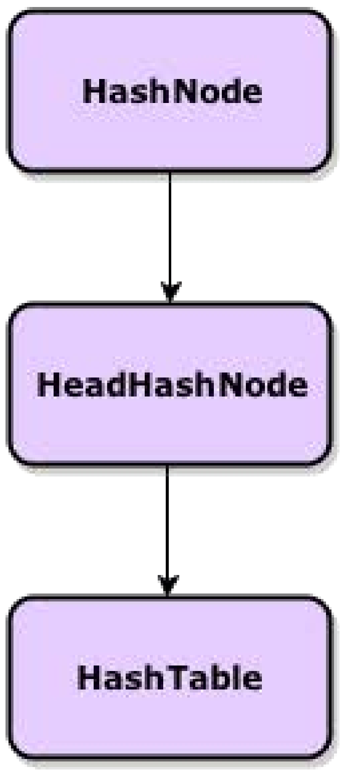


MENU - Structure Chart



UML Structure Chart





BinaryTree - UML

BinaryTree

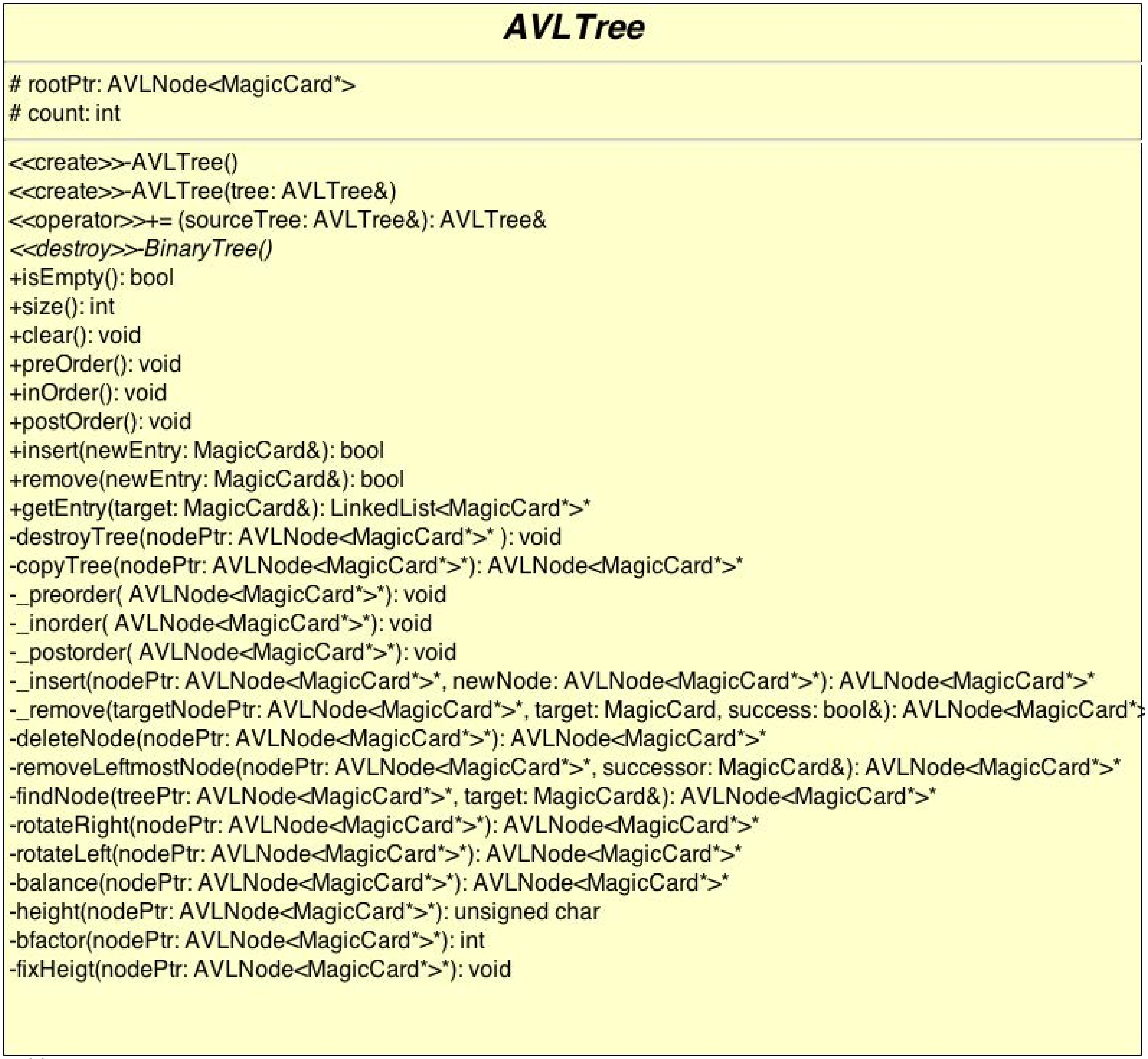
```
# rootPtr: TreeNode*
# count: int
```

```
<<create>>-BinaryTree()
<<create>>-BinaryTree(sourceTree: BinaryTree)
<<operator>>+= (sourceTree: BinaryTree&): BinaryTree
<<destroy>>-BinaryTree()
+isEmpty(): bool
+size(): int
+getHeight(): int
+clear(): void
+preOrder(): void
+inOrder(): void
+postOrder(): void
+insert(code: string): bool
+remove(code: string): bool
+getEntry(anEntry: string, returnedItem: string): bool
-destroyTree(nodePtr: TreeNode*): void
-copyTree(nodePtr: TreeNode*): TreeNode*
-_preorder(nodePtr: TreeNode*): void
-_inorder(nodePtr: TreeNode*): void
-_postorder(nodePtr: TreeNode*): void
```

BinarySearchTree

```
-_insert(nodePtr: TreeNode*, newNode: TreeNode*): TreeNode*
+findNode(treePtr: TreeNode*, target: string&): TreeNode*
+FindMin(root: TreeNode*): TreeNode*
-_remove(nodePtr: TreeNode*, target: string, success: bool&): TreeNode*
+displayTree(nodePtr: TreeNode*): void
+displayIndentedTree(nodePtr: TreeNode*, lineNum: int&): void
```

AVLTree - UML



AVLTree

rootPtr: AVLNode<MagicCard*>
count: int

<<create>>-AVLTree()
<<create>>-AVLTree(tree: AVLTree&)
<<operator>>+= (sourceTree: AVLTree&): AVLTree&
<<destroy>>-BinaryTree()
+isEmpty(): bool
+size(): int
+clear(): void
+preOrder(): void
+inOrder(): void
+postOrder(): void
+insert(newEntry: MagicCard&): bool
+remove(newEntry: MagicCard&): bool
+getEntry(target: MagicCard&): LinkedList<MagicCard*>*
-destroyTree(nodePtr: AVLNode<MagicCard*>*): void
-copyTree(nodePtr: AVLNode<MagicCard*>*): AVLNode<MagicCard*>*
-_preorder(AVLNode<MagicCard*>*): void
-_inorder(AVLNode<MagicCard*>*): void
-_postorder(AVLNode<MagicCard*>*): void
-_insert(nodePtr: AVLNode<MagicCard*>*, newNode: AVLNode<MagicCard*>*): AVLNode<MagicCard*>*
-_remove(targetNodePtr: AVLNode<MagicCard*>*, target: MagicCard, success: bool&): AVLNode<MagicCard*
-deleteNode(nodePtr: AVLNode<MagicCard*>*): AVLNode<MagicCard*>*
-removeLeftmostNode(nodePtr: AVLNode<MagicCard*>*, successor: MagicCard&): AVLNode<MagicCard*>*
-findNode(treePtr: AVLNode<MagicCard*>*, target: MagicCard&): AVLNode<MagicCard*>*
-rotateRight(nodePtr: AVLNode<MagicCard*>*): AVLNode<MagicCard*>*
-rotateLeft(nodePtr: AVLNode<MagicCard*>*): AVLNode<MagicCard*>*
-balance(nodePtr: AVLNode<MagicCard*>*): AVLNode<MagicCard*>*
-height(nodePtr: AVLNode<MagicCard*>*): unsigned char
-bfactor(nodePtr: AVLNode<MagicCard*>*): int
-fixHeigt(nodePtr: AVLNode<MagicCard*>*): void

Hash - UML

HashTable

-table: HeadHashNode<KeyType, ItemType>**
-tableSize: int
-filledSlots: int
-collisions: int
-loadFactor: double
-listCount: int
-longListCount: int
-totalInLists: int
-avgInLists: int

<<create>>-HashTable(size: int)
<<create>>-HashTable(hash(const KeyType &, int): int, HashTable&, size: int)
<<destroy>>-HashTable()
-getHashValue(hash(const KeyType &, int): int, key: KeyType&)
-init(size:int): void
+addEntry(hash(const KeyType &, int): int, key: KeyType&, item: ItemType&): void
+displayTable(display(ItemType&): void): void
+printTable(display(ItemType&): void): void
+search(hash(const KeyType &, int): int, key: KeyType&, item: ItemType&): bool
+remove(hash(const KeyType &, int): int, key: KeyType&, item: ItemType&): bool
+displayStatistics(): void
+getTableSize(): int
+getFilledSlots(): int
+getCollisions(): int
+getLoadFactor(): double
+getListCount(): int
+getTLongListCount(): int
+getTotalInLists(): int
+getAvgInLists(): int
+getKeys(keys: vector<KeyType>&): void
+getItems(items: vector<ItemType>&): int
+destroyTable(): void

HashNode

-key: KeyType
-item: ItemType
-next: HashNode*

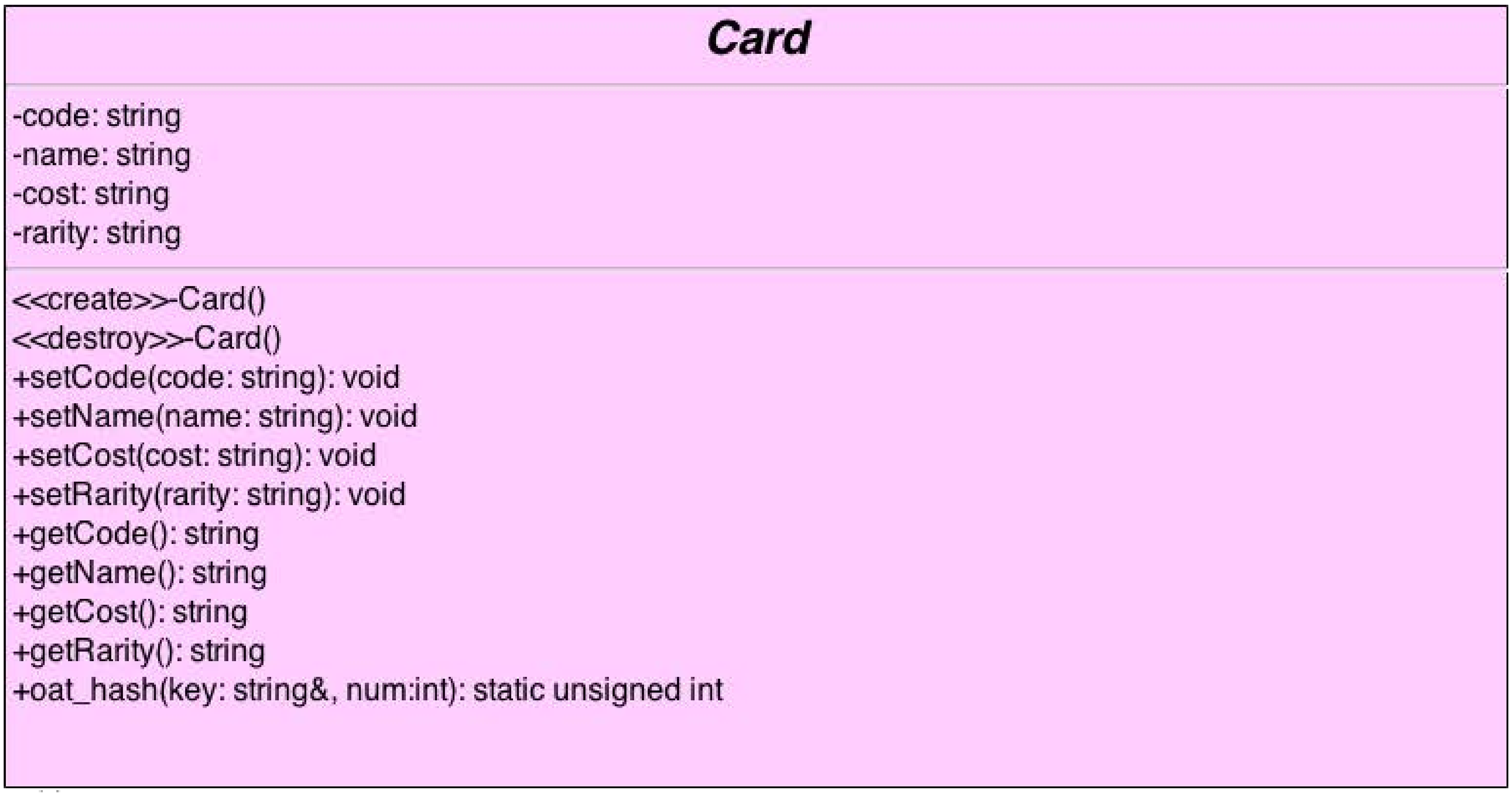
<<create>>-HashNode(key: KeyType&, item: ItemType&)
<<destroy>>-HashNode()
+setKey(k: KeyType&); void
+setItem(i: ItemType&); void
+setNext(n: HashNode*); void
+getKey(kt: KeyType&); void
+getItem(it: ItemType&); void
+getNext(); HashNode*

HeadHashNode

-listCount: Int

<<create>>-HeadHashNode(key: KeyType&, item: ItemType&)
<<destroy>>-HeadHashNode()
+addToList(key: KeyType&, item: ItemType&): void
+removeFromList(target: KeyType&, item: ItemType&)
+getListCount(): int

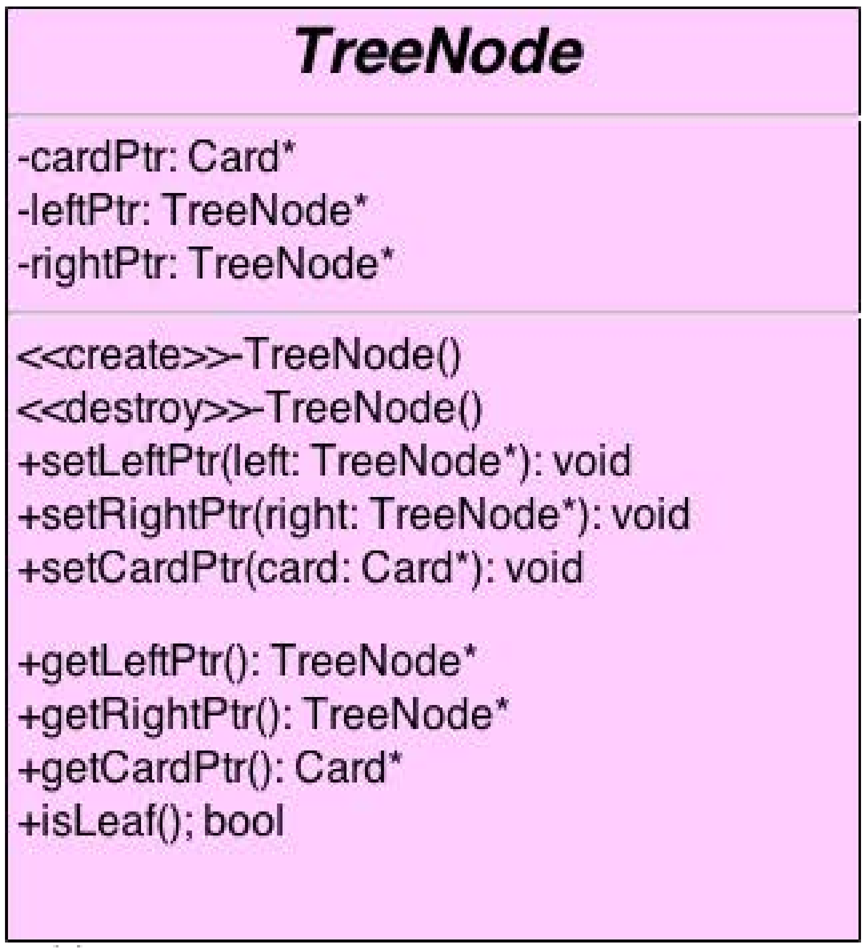
Misc Classes - UML



Card

-code: string
-name: string
-cost: string
-rarity: string

<<create>>-Card()
<<destroy>>-Card()
+setCode(code: string): void
+setName(name: string): void
+setCost(cost: string): void
+setRarity(rarity: string): void
+getCode(): string
+getName(): string
+getCost(): string
+getRarity(): string
+oat_hash(key: string&, num:int): static unsigned int

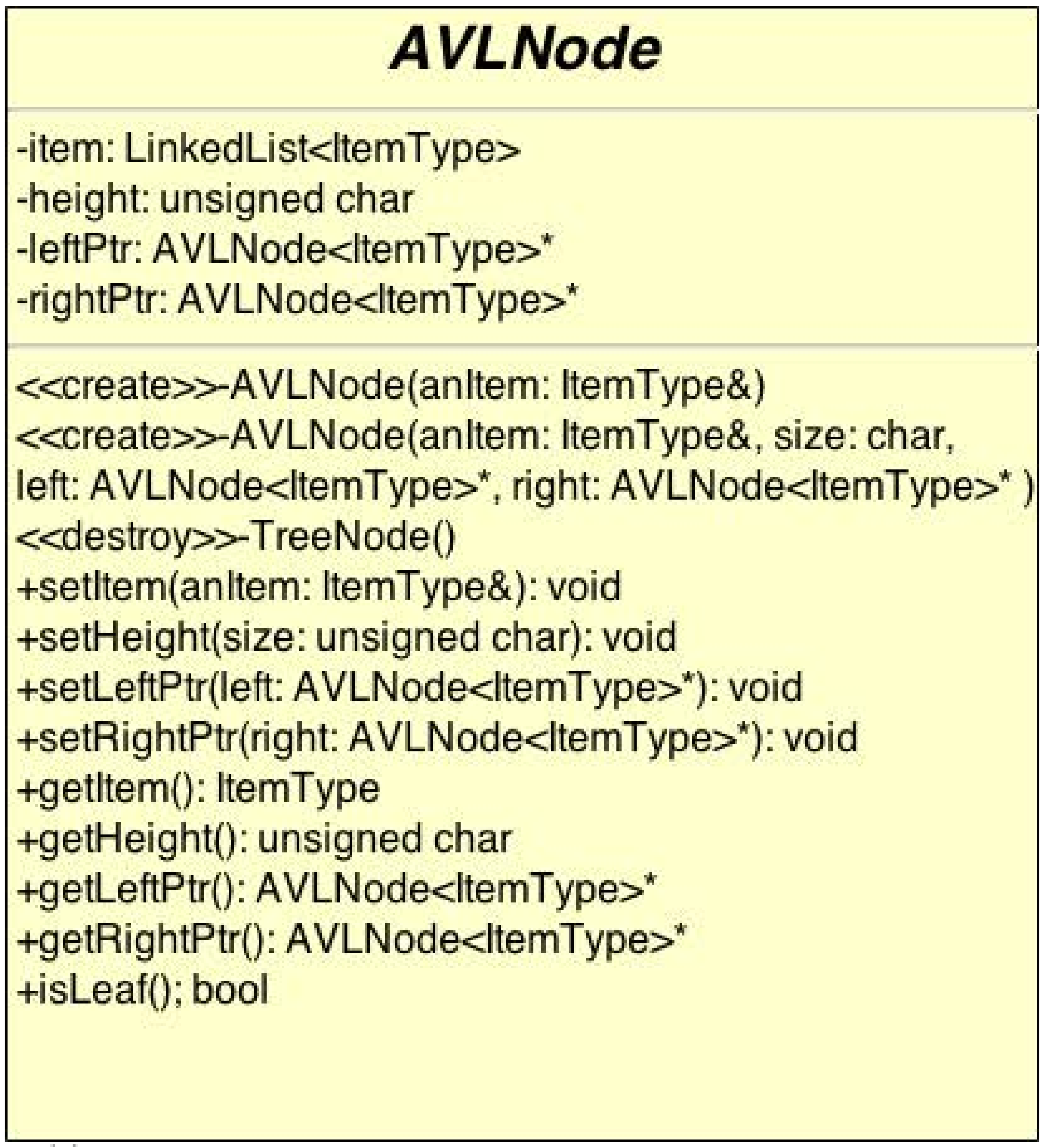


TreeNode

-cardPtr: Card*
-leftPtr: TreeNode*
-rightPtr: TreeNode*

<<create>>-TreeNode()
<<destroy>>-TreeNode()
+setLeftPtr(left: TreeNode*): void
+setRightPtr(right: TreeNode*): void
+setCardPtr(card: Card*): void

+getLeftPtr(): TreeNode*
+getRightPtr(): TreeNode*
+getCardPtr(): Card*
+isLeaf(); bool



AVLNode

-item: LinkedList<ItemType>
-height: unsigned char
-leftPtr: AVLNode<ItemType>*
-rightPtr: AVLNode<ItemType>*

<<create>>-AVLNode(anItem: ItemType&)
<<create>>-AVLNode(anItem: ItemType&, size: char,
left: AVLNode<ItemType>*, right: AVLNode<ItemType>*)
<<destroy>>-TreeNode()
+setItem(anItem: ItemType&): void
+setHeight(size: unsigned char): void
+setLeftPtr(left: AVLNode<ItemType>*): void
+setRightPtr(right: AVLNode<ItemType>*): void
+getItem(): ItemType
+getHeight(): unsigned char
+getLeftPtr(): AVLNode<ItemType>*
+getRightPtr(): AVLNode<ItemType>*
+isLeaf(); bool