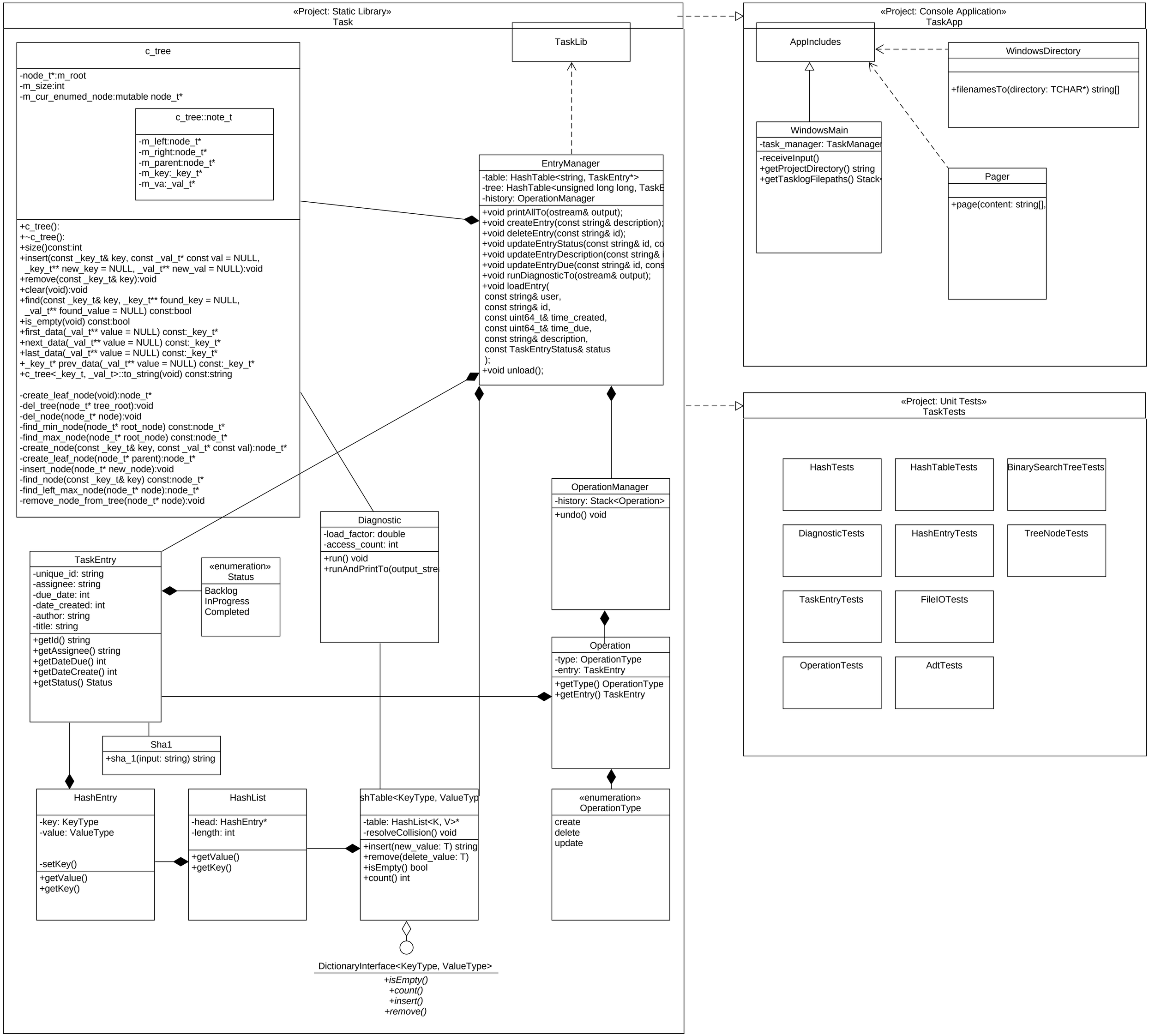
«Project: Static Library»
Task
TaskLib
c_tree
-node_t*:m_root
-m_size:int
-m_cur_enumed_node:mutable node_t*
c_tree::note_t
-m_left:node_t*
-m_right:node_t*
-m_parent:node_t*
-m_key:_key_t*
-m_va:_val_t*
+c_tree():
+~c_tree():
+size()const:int
+insert(const _key_t& key, const _val_t* const val = NULL,
_key_t** new_key = NULL, _val_t** new_val = NULL):void
+remove(const _key_t& key):void
+clear(void):void
+find(const _key_t& key, _key_t** found_key = NULL,
_val_t** found_value = NULL) const:bool
+is_empty(void) const:bool
+first_data(_val_t** value = NULL) const:_key_t*
+next_data(_val_t** value = NULL) const:_key_t*
+last_data(_val_t** value = NULL) const:_key_t*
+_key_t* prev_data(_val_t** value = NULL) const:_key_t*
+c_tree<_key_t, _val_t>::to_string(void) const:string
-create_leaf_node(void):node_t*
-del_tree(node_t* tree_root):void
-del_node(node_t* node):void
-find_min_node(node_t* root_node) const:node_t*
-find_max_node(node_t* root_node) const:node_t*
-create_node(const _key_t& key, const _val_t* const val):node_t*
-create_leaf_node(node_t* parent):node_t*
-insert_node(node_t* new_node):void
-find_node(const _key_t& key) const:node_t*
-find_left_max_node(node_t* node):node_t*
-remove_node_from_tree(node_t* node):void
EntryManager
-table: HashTable<string, TaskEntry*>
-tree: HashTable<unsigned long long, TaskE
-history: OperationManager
+void printAllTo(ostream& output);
+void createEntry(const string& description);
+void deleteEntry(const string& id);
+void updateEntryStatus(const string& id, co
+void updateEntryDescription(const string&
+void updateEntryDue(const string& id, cons
+void runDiagnosticTo(ostream& output);
+void loadEntry(
const string& user,
const string& id,
const uint64_t& time_created,
const uint64_t& time_due,
const string& description,
const TaskEntryStatus& status
);
+void unload();
OperationManager
-history: Stack<Operation>
+undo() void
Diagnostic
-load_factor: double
-access_count: int
+run() void
+runAndPrintTo(output_stre
TaskEntry
-unique_id: string
-assignee: string
-due_date: int
-date_created: int
-author: string
-title: string
+getId() string
+getAssignee() string
+getDateDue() int
+getDateCreate() int
+getStatus() Status
«enumeration»
Status
Backlog
InProgress
Completed
Operation
-type: OperationType
-entry: TaskEntry
+getType() OperationType
+getEntry() TaskEntry
Sha1
+sha_1(input: string) string
HashEntry
-key: KeyType
-value: ValueType
-setKey()
+getValue()
+getKey()
HashList
-head: HashEntry*
-length: int
+getValue()
+getKey()
shTable<KeyType, ValueTyp
-table: HashList<K, V>*
-resolveCollision() void
+insert(new_value: T) string
+remove(delete_value: T)
+isEmpty() bool
+count() int
«enumeration»
OperationType
create
delete
update
DictionaryInterface<KeyType, ValueType>
+isEmpty()
+count()
+insert()
+remove()
«Project: Console Application»
TaskApp
AppIncludes
WindowsDirectory
+filenamesTo(directory: TCHAR*) string[]
WindowsMain
-task_manager: TaskManager
-receiveInput()
+getProjectDirectory() string
+getTasklogFilepaths() Stack<
Pager
+page(content: string[],
«Project: Unit Tests»
TaskTests
HashTests
HashTableTests
BinarySearchTreeTests
DiagnosticTests
HashEntryTests
TreeNodeTests
TaskEntryTests
FileIOTests
OperationTests
AdtTests