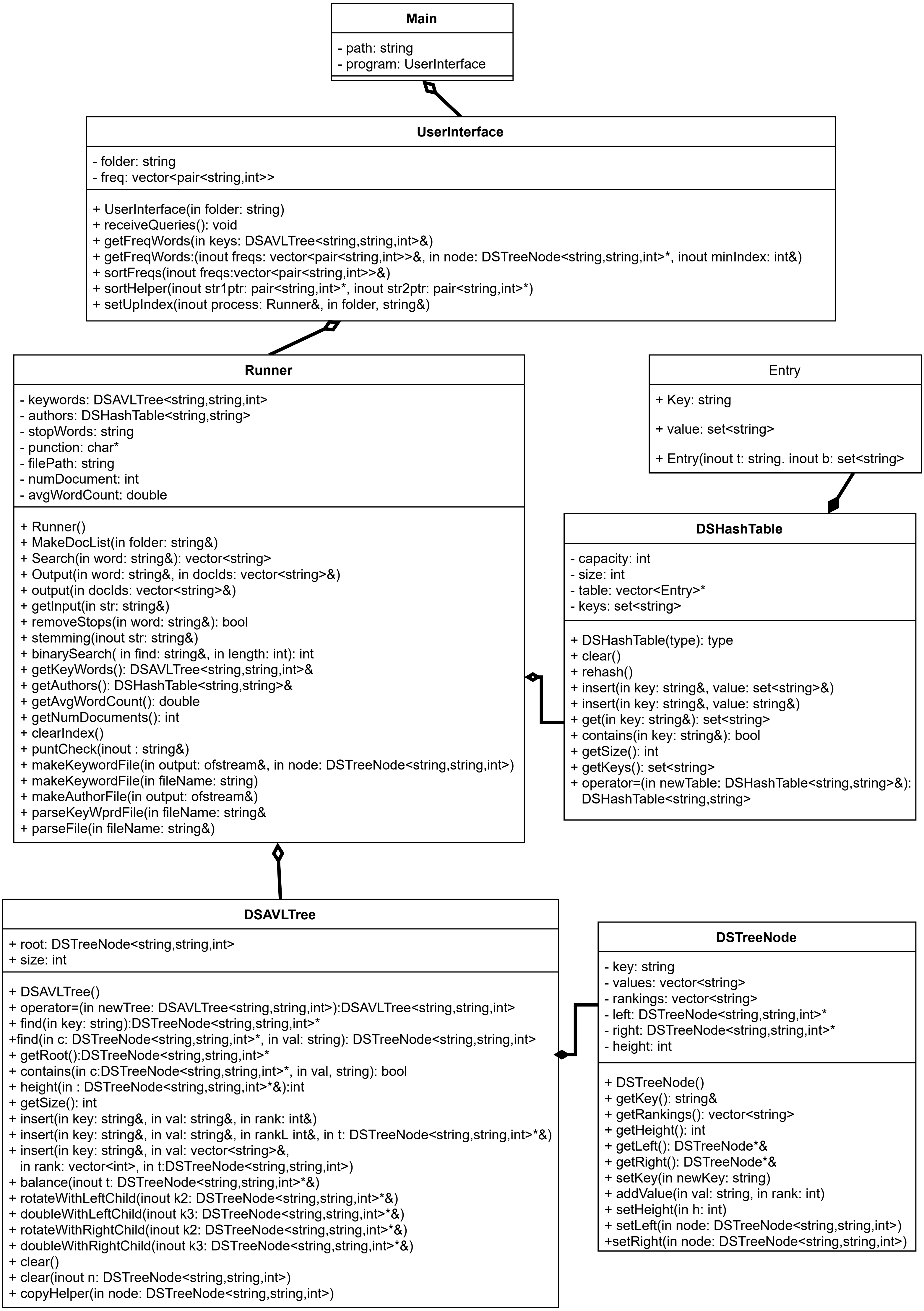

```
Main
- path: string
- program: UserInterface
```

```
UserInterface
- folder: string
- freq: vector<pair<string,int>>

+ UserInterface(in folder: string)
+ receiveQueries(): void
+ getFreqWords(in keys: DSAVLTree<string,string,int>&)
+ getFreqWords:(inout freqs: vector<pair<string,int>>&, in node: DSTreeNode<string,string,int>*, inout minIndex: int&)
+ sortFreqs(inout freqs:vector<pair<string,int>>&)
+ sortHelper(inout str1ptr: pair<string,int>*, inout str2ptr: pair<string,int>*)
+ setUpIndex(inout process: Runner&, in folder, string&)
```

```
Runner
- keywords: DSAVLTree<string,string,int>
- authors: DSHashTable<string,string>
- stopWords: string
- punction: char*
- filePath: string
- numDocument: int
- avgWordCount: double

+ Runner()
+ MakeDocList(in folder: string&)
+ Search(in word: string&): vector<string>
+ Output(in word: string&, in docIds: vector<string>&)
+ output(in docIds: vector<string>&)
+ getInput(in str: string&)
+ removeStops(in word: string&): bool
+ stemming(inout str: string&)
+ binarySearch( in find: string&, in length: int): int
+ getKeyWords(): DSAVLTree<string,string,int>&
+ getAuthors(): DSHashTable<string,string>&
+ getAvgWordCount(): double
+ getNumDocuments(): int
+ clearIndex()
+ puntCheck(inout : string&)
+ makeKeywordFile(in output: ofstream&, in node: DSTreeNode<string,string,int>)
+ makeKeywordFile(in fileName: string)
+ makeAuthorFile(in output: ofstream&)
+ parseKeyWprdFile(in fileName: string&
+ parseFile(in fileName: string&)
```

```
Entry
+ Key: string

+ value: set<string>

+ Entry(inout t: string. inout b: set<string>
```

```
DSHashTable
- capacity: int
- size: int
- table: vector<Entry>*
- keys: set<string>

+ DSHashTable(type): type
+ clear()
+ rehash()
+ insert(in key: string&, value: set<string>&)
+ insert(in key: string&, value: string&)
+ get(in key: string&): set<string>
+ contains(in key: string&): bool
+ getSize(): int
+ getKeys(): set<string>
+ operator=(in newTable: DSHashTable<string,string>&):
  DSHashTable<string,string>
```

```
DSAVLTree
+ root: DSTreeNode<string,string,int>
+ size: int

+ DSAVLTree()
+ operator=(in newTree: DSAVLTree<string,string,int>):DSAVLTree<string,string,int>
+ find(in key: string):DSTreeNode<string,string,int>*
+find(in c: DSTreeNode<string,string,int>*, in val: string): DSTreeNode<string,string,int>
+ getRoot():DSTreeNode<string,string,int>*
+ contains(in c:DSTreeNode<string,string,int>*, in val, string): bool
+ height(in : DSTreeNode<string,string,int>*&):int
+ getSize(): int
+ insert(in key: string&, in val: string&, in rank: int&)
+ insert(in key: string&, in val: string&, in rankL int&, in t: DSTreeNode<string,string,int>*&)
+ insert(in key: string&, in val: vector<string>&,
  in rank: vector<int>, in t:DSTreeNode<string,string,int>)
+ balance(inout t: DSTreeNode<string,string,int>*&)
+ rotateWithLeftChild(inout k2: DSTreeNode<string,string,int>*&)
+ doubleWithLeftChild(inout k3: DSTreeNode<string,string,int>*&)
+ rotateWithRightChild(inout k2: DSTreeNode<string,string,int>*&)
+ doubleWithRightChild(inout k3: DSTreeNode<string,string,int>*&)
+ clear()
+ clear(inout n: DSTreeNode<string,string,int>)
+ copyHelper(in node: DSTreeNode<string,string,int>)
```

```
DSTreeNode
- key: string
- values: vector<string>
- rankings: vector<string>
- left: DSTreeNode<string,string,int>*
- right: DSTreeNode<string,string,int>*
- height: int

+ DSTreeNode()
+ getKey(): string&
+ getRankings(): vector<string>
+ getHeight(): int
+ getLeft(): DSTreeNode*&
+ getRight(): DSTreeNode*&
+ setKey(in newKey: string)
+ addValue(in val: string, in rank: int)
+ setHeight(in h: int)
+ setLeft(in node: DSTreeNode<string,string,int>)
+setRight(in node: DSTreeNode<string,string,int>)
```