Michael Donaldson

10/13/2024

Vector-Hash-Tree Pseudocode

Tree Data Structure Pseudocode

Function to Print Course Information

Function searchCourse(BST<Course> tree, String courseNumber):

Set course = tree.search(courseNumber)

If course is not found:

Print "Course not found."

Else:

Print course.number, course.title

If course has prerequisites:

Print "Prerequisites:"

For each prerequisite in course.prerequisites:

Print prerequisite

In-Order Traversal to Print All Courses

Function inOrderTraversal(BST<Course> tree):

If tree is not empty:

Traverse left subtree

Print current course information (course number, title)

Traverse right subtree

Open file "courses.txt" for reading.

Parse

While not end of file:

Read line

Split line by commas (or spaces)

Set course number = first token

Set course title = second token

Set prerequisites = remaining tokens

Validate

For each prerequisite:

If prerequisite does not exist in the set/map of course numbers:

Log error (invalid prerequisite)

Insert Course into BST

Create new Course object with course number, title, and prerequisites

Insert Course object into BST

Add course number to course set/map

Hash Table Data Structure

Open File and Read Data:

BEGIN

FUNCTION loadCourses(fileName)

DECLARE fileHandle

DECLARE line

DECLARE courseHashTable as HashTable

TRY

OPEN fileName FOR reading AS fileHandle

WHILE fileHandle HAS MORE LINES

READ line FROM fileHandle

PARSE line INTO courseNumber, courseTitle, prerequisites

IF INVALID\_LINE\_FORMAT(line) THEN

PRINT "Error: Line is not correctly formatted."

CONTINUE TO NEXT LINE

CREATE CourseObject WITH courseNumber, courseTitle, prerequisites

IF prerequisites NOT EMPTY THEN

FOR EACH prerequisite IN prerequisites

IF NOT courseHashTable.CONTAINS(prerequisite) THEN

PRINT "Error: Prerequisite does not exist in the file."

CONTINUE TO NEXT LINE

END FOR

STORE CourseObject IN courseHashTable USING courseNumber AS KEY

END WHILE

CATCH error

PRINT "Error: Could not open file."

FINALLY

CLOSE fileHandle

END FUNCTION

END

Parse and Check Errors:

FUNCTION PARSE line INTO courseNumber, courseTitle, prerequisites

SPLIT line BY comma

IF line HAS AT LEAST TWO PARTS

ASSIGN courseNumber, courseTitle FROM first two parts

ASSIGN remaining parts as prerequisites

ELSE

RETURN INVALID

END IF

END FUNCTION

FUNCTION INVALID\_LINE\_FORMAT(line)

IF SPLIT line BY comma HAS LESS THAN TWO ELEMENTS

RETURN TRUE

ELSE

RETURN FALSE

END IF

END FUNCTION

Create and Store Course Objects:

FUNCTION CREATE CourseObject(courseNumber, courseTitle, prerequisites)

DECLARE CourseObject WITH

courseNumber AS number,

courseTitle AS title,

prerequisites AS prereqList

END FUNCTION

Print Out Course Information and Prerequisites:

FUNCTION printCourseInformation(courseHashTable)

FOR EACH courseKey IN courseHashTable

DECLARE course = courseHashTable.GET(courseKey)

PRINT "Course Number: " + course.number

PRINT "Course Title: " + course.title

PRINT "Prerequisites: " + course.prereqList (OR "None" IF EMPTY)

END FUNCTION

**Vector Data Structure**

**Open and Read the File, Validate Format**

Function loadCourseData(fileName)

Open fileName for reading

If file cannot be opened

Print "Error: File cannot be opened."

Return

End If

Initialize an empty vector called courseList

Initialize an empty set called courseNumbers (for validation)

While there are lines in the file

Read line from file

// Split the line by commas to extract course data

Split the line into courseDetails by commas

If courseDetails contains less than 2 elements

Print "Error: Invalid format - not enough fields."

Continue to the next line

End If

// Extract course information

courseNumber = courseDetails[0]

courseTitle = courseDetails[1]

// Add courseNumber to courseNumbers set (used for validation later)

Add courseNumber to courseNumbers

Initialize an empty list called prerequisites

// Check if there are prerequisites on the line

For each element in courseDetails starting from index 2

Add element to prerequisites list

End For

// Create a new course object

course = createCourse(courseNumber, courseTitle, prerequisites)

// Add course to the courseList vector

Add course to courseList

End While

// Now, validate that all prerequisites exist

For each course in courseList

For each prerequisite in course.prerequisites

If prerequisite is not found in courseNumbers

Print "Error: Prerequisite" + prerequisite + " not found for course " + course.courseNumber

End If

End For

End For

Close file

Return courseList

End Function

**Create Course Object and Store in Vector**

// Struct definition for a course object

Struct Course

String courseNumber

String courseTitle

List of Strings prerequisites

End Struct

Function createCourse(courseNumber, courseTitle, prerequisites)

Initialize newCourse of type Course

Set newCourse.courseNumber = courseNumber

Set newCourse.courseTitle = courseTitle

Set newCourse.prerequisites = prerequisites

Return newCourse

End Function

**Search for Course and Print Course Information**

// Function to print course information and prerequisites for a specific course

Function printCourseInformation(courseList, targetCourseNumber)

Initialize found = False

For each course in courseList

If course.courseNumber equals targetCourseNumber

Print "Course Number: " + course.courseNumber

Print "Course Title: " + course.courseTitle

If course.prerequisites is not empty

Print "Prerequisites: "

For each prerequisite in course.prerequisites

Print prerequisite

End For

Else

Print "No prerequisites"

End If

found = True

Break

End If

End For

If found equals False

Print "Error: Course not found."

End If

End Function