Michael Wood

09/22/2024

SNHU Analysis and Design 2024

Professor: Michael Rissover

3-3 Project One Milestone One: Vector Data Structure

**Pseudocode for Retrieving Data from a File:**

* Procedure ReadFile(filePath):
* Open file at filePath
* If file is not found:
* Print "Error: File not found."
* Exit program
* If LineIsValid(line):
* ParseLine(line)
* Else:
* Print "Error: Invalid line format."
* Close file
* End Procedure
* Function LineIsValid(line):
* Check if the line has at least two parameters
* tokens = Split(line, ',')
* If Length(tokens) >= 2:
* Return True
* Else:
* Return False
* End Function
* Procedure ParseLine(line):
* Extract course information from the line
* tokens = Split(line, ',')
* courseNumber = tokens[0]
* courseTitle = tokens[1]
* prerequisites = tokens[2]

* Validate prerequisites
* For each prerequisite in prerequisites:
* If not CourseExists(prerequisite):
* Print "Error: Prerequisite not found for course ", courseNumber
* Exit program
* Create course object and store in vector data structure
* courseObject = CreateCourseObject(courseNumber, courseTitle, prerequisites)
* VectorAppend(courseVector, courseObject)
* End Procedure
* Function CourseExists(courseNumber):
* Check if a course with the given courseNumber exists in the vector
* For each course in courseVector:
* If course.courseNumber = courseNumber:
* Return True
* Return False
* End Function

**Pseudocode for Creating Course Objects and Storing in Vector:**

* Procedure CreateCourseObject(courseNumber, courseTitle, prerequisites):
* Create a new course object
* newCourse = new CourseObject()
* Set instance variables with provided data
* newCourse.courseNumber = courseNumber
* newCourse.courseTitle = courseTitle
* newCourse.prerequisites = prerequisites
* Return newCourse
* End Procedure
* Procedure VectorAppend(vector, element):
* Add an element to the end of the vector
* vector[length(vector)] = element
* End Procedure

**Pseudocode for Searching and Printing Course Information:**

* Procedure PrintCourseInformation(courseNumber):
* For each course in courseVector:
* If course.courseNumber = courseNumber:
* Print "Course Number: ", course.courseNumber
* Print "Course Title: ", course.courseTitle
* Print "Prerequisites: ", course.prerequisites
* Return
* If Course Number or Title not found
* Print "Error: Course not found."
* End Procedure

Loading Data from File: Each line of a file is read by the ReadFile process, which also verifies the format before parsing the data into course objects.

If a line has two parameters or more, it is validated using the LineIsValid function. The ParseLine process generates course objects, verifies prerequisites, and pulls course metadata. Making Course Objects and Vector Storage:

A new course object is created, and its properties are configured by the CreateCourseObject process. The course object is appended to the vector data structure using the VectorAppend function. Looking up and printing course materials: Using the course number, the PrintCourseInformation method looks up a specific course in the vector. If the course is located, the details (number, title, prerequisites) are printed; if not, an error notice is printed.