class VectorTextFile

VectorTextFile

```
/**************************************************************************
 Class: VectorTextFile
 Purpose: Represents a text file as a vector, i.e., as a sorted array of
 word/count pairs that appear in the text file.

 Constructor: VectorTextFile(String fileName)
 Behavior: Reads the specified text file and parses it appropriately.

 Public Class Methods:
    int Norm() : Returns the norm of the vector.

 Static Methods:
    double DotProduct(VectorTextFile A, VectorTextFile B) :
        Returns the dot product of two vectors.
    double Angle(VectorTextFile A, VectorTextFile B) :
        Returns the angle between two vectors.
 **************************************************************************/

// This class is part of the cs2020 package.
package sg.edu.nus.cs2020;

// This class uses the following two packages (associated with reading files):
import java.io.FileInputStream;
import java.io.IOException;
```

Compare Two Documents

Given: documents A and B

1. Read and parse text
2. Create vectors v_A and v_B
3. Calculate norm: $|v_A|$
4. Calculate norm: $|v_B|$
5. Calculate dot product: $(v_A \cdot v_B)$
6. Calculate angle $\Phi(v_A, v_B)$

Document Distance

Basic object/class:

VectorTextFile

Functionality:

- Initialization: Reads in file
- public method: Norm
- static: Dot-product of two vectors
- static: Angle between two vectors

Document Distance

Basic object/class:

VectorTextFile

Public functionality:

```
double norm()
intDotProduct(VectorTextFile A, VectorTextFile B)
double Angle(VectorTextFile A, VectorTextFile B)
```

All other functionality is private / internal.

Document Distance Main

```
package sg.edu.nus.cs2020;

public class DocumentDistanceMain
{

    public static void main(String[] args)
    {
        VectorTextFile A = new VectorTextFile("FileOne.txt");
        VectorTextFile B = new VectorTextFile("FileTwo.txt");

        double theta = VectorTextFile.Angle(A,B);

        System.out.println("The angle between A and B is: " + theta + "\n");
    }
}
```

Document Distance

Basic object/class:

VectorTextFile

Constructor: parameter filename

- Reads in file
- Parses file into words
- Sorts words
- Counts word frequencies

Member Variable Declaration

```
// Class declaration:
public class VectorTextFile {

    /*******************************
     * Class member variables
     *******************************/

    // Array of words in the file
    String[]  m_WordList;

    // Number of words in the file
    int m_FileWordCount;

    // Array of word/count pairs
    WordCountPair[] m_CountedWords;

    // Number of word/count pairs
    int m_WordPairCount;

    // Has the word list been sorted?
    boolean m_Sorted;
```

Constructor

```
//-----------------------------------------------------------------------
// Constructor: Reads and parses the specified file
//
// Input: String containing a filename
// Assumptions: fileName is a text file that exists on disk.
// Properties: On completion, m_WordList contains a sorted array of all the
// words in the text file, m_FileWordCount is the number of words in the
// text file, m_CountedWords contains a sorted array of word/count pairs
// with one entry for every distinct word in the text file, m_WordPairCount
// is the number of word/count pairs, and the flag m_Sorted is true.
// Characters in the file are treated in the following manner:
// (a) Every letter is made lower-case.
// (b) All punctuation is removed.
// (c) Each end-of-line marker ('\n') is replaced with a space.
// (d) All (other) non-letters and non-spaces are removed.
//-----------------------------------------------------------------------
public VectorTextFile(String fileName)
{
    // Begin a block of code that handles exceptions (i.e., errors)
    try{

        // First, initialize class variables
        m_WordList = null;
        m_CountedWords = null;
        m_FileWordCount = 0;
        m_WordPairCount = 0;
        m_Sorted = false;
```

Constructor

```
            // Next, read in the file and parse it into words.
            ParseFile(fileName);

            // Check for errors:
            if ((m_FileWordCount < 1) || (m_WordList == null))
            {
                throw new Exception("Reading the file failed.");
            }

            // Next, sort the words.
            InsertionSortWords();

            // Check for errors:
            if (m_Sorted == false)
            {
                throw new Exception("Sorting failed.");
            }
            VerifySort();

            // Finally, count the number of times each word appears in the file.
            CountWordFrequencies();

            // Check for errors:
            if ((m_WordPairCount < 1) || (m_CountedWords == null))
            {
                throw new Exception("Counting the word frequencies failed.");
            }
        }
        // Catch any exceptions (i.e., errors) and report problems.
        catch(Exception e)
        {
            System.out.println("Error creating VectorTextFile.");
        }
    }
```

Document Distance

Secondary object/class: WordCountPair

Encapsulates:

- String word
- int count

Functionality:

- Constructor: sets word and counts
- `String getWord()`
- `int getCount()`

For next time…

Tutorial:

- Today: another example…

Wednesday:

- More OOP
- Inheritance
- Lists

Problem Set 1:

- Released. Due next Tuesday night.