Table of Contents

[Project Overview 2](#__RefHeading___Toc4258_1370385848)

[Top Level Functions 3](#__RefHeading___Toc4260_1370385848)

[PalindromeT (Enumeration) 5](#__RefHeading___Toc2147_279954108)

# Project Overview

This project provides a “library” for classifying palindromes.

**Structure Chart**

# Top Level Functions

These functions provide

* A routine to classify a phrase as one of the given palindrome types.
* An enumerated type providing palindrome types
	+ Support routines for this enumerated type.

**Main**

Narrative: No main is provided with this library

**Function:** ClassifyPhrase

Narrative: This routine takes a phrase and returns the palindrome type that best describes the phrase.

Input: a string, the phrase to classify

Output: the enumerated palindrome type value which best classifies the

*If CheckPhrase of the string*

 *return perfect type*

*else*

 *remove all punctuation and make all alphabetic characters uppercase*

 *if CheckPhrase of the modified string*

 *return natural palindrome type*

 *else*

 *build a string of only the alphabetic characters changed to upper case.*

 *If CheckPhrase of the modified string*

 *return palindrome type*

 *else*

 *return non palindrome type.*

**Function:** Check Phrase

Narrative: Check to see if a phrase matches the reverse of the phrase

Input: A string, the phrase to check

Output: A boolean, true of the phrases matches the reverse of the phrase, false otherwise.

*Reverse the string*

*return Compare(string, reverse)*

**Function:** Compare

Narrative: Compare two strings to see if they are the same.

Input: Two strings

Output: A bool, ture if the strings are the same, false otherwise.

*If the sizes of the strings are different*

 *return false*

*else*

 *loop through the strings to check to see if all the letters are the same*

 *If a difference is found*

 *return false*

 *else*

 *return true*

**Function:** ReverseString

Narrative: Given a string, compute a string that is the reverse

Input: A string

Output: A string the reverse of the string

*for each letter in the string*

 *reverse = letter + reverse*

**Function:** Strip1

Narrative: Remove all punctuation from a string and convert it to all upper case letters

Input: a string

Output: the modified string

*for each character in the string*

 *if the caracter is not a punctuation (as determined by ispunct)*

 *add the upper case version of the character to the return value*

**Function:** Strip2

Narrative: Take a string and remove all but the alphabetic characters from the string.

Input: A string

Output: A modified string

*for each character in the string*

 *if the character is a letter*

 *add the uppercase version of the letter to the return value*

# PalindromeT (Enumeration)

This type provides a set of classification types for a palindrome.

Properties

Domain of the EnumeratedT

PERFECT, NATURAL, REGULAR, NON, UNKNOWN

Note: Unknown is not really a class, it provides a value for initialization and errors.

Behaviors

Convert a string to a PalindromeT

Convert a PalindromeT to a string

Given a PalindromeT, calculate the next value.

Discussion:

 Note the “next” value is designated by the order listed above.

 The next value following UNKNOWN is UNKNOWN

**Function:** PalinderomeTToString

Narrative: Given a PalindromeT, return the corresponding string.

Input: A PalindromeT

Output: A string

*switch/case*

**Function:** StringToPalindromeT

Narrative: Given a string, provide the corresponding PalindromeT

Input: A string

Output: A PalindromeT

*if / else if / else statement*

**Function:** NextPalindromeT

Narrative: Compute the next PalindromeT

Input: A PalindromeT

Output: A PalindromeT

*switch/case*