Hassan Harajly

Cis 200

9/25/17

**My String Program**

1. **Problem Statement**

Must create a class like the string class. It must hold a maximum of 25 characters. Will have various methods.

1. **Requirements**
   1. **Assumptions**

The user will enter integers, characters and symbols, only 25 of any combination.

Will enter input through command line input/output.

Programmers can also use class for their programs, we want low level of codependence(coupling).

* 1. **Specifications**

The class will contain the capability to store 25 characters, symbols and numbers.

Will have to create a character array to carry the input. Inside of the mystring class.

End array with “/0” in order to know when the end of array is.

Methods of the mystring class:

Size() will determine how many caharacters are in the string.

addstart(mystring)- will add the string parameter to the current string in the front of it.25 char limit.

Addend(mystring)-will add the string in the input parameter to the end of the current string.not above 25.

partString(startPos, length) – returns string from startPos for length given (handle startPos < 0, startPos = size returns null string, handle startPos > size).

replPartString(myString, startPos) – replaces charaters starting at startPos with input string – (what if exceeds 25?).

compareString(myString) – compare current value of string with input parameter string.

initString() – resets/initializes string to null string

setString(string) – assign string to myString data value

getString() – returns string of data from myString data value

printString() – prints myString data value to the monitor

1.The user will enter up to 25 characters.

If the user enters more than 25 characters they will be prompted with an error message to reenter.

When the mystring input is correctly stored, the user will be able to choose from the methods listed above depending upon what the user wants.

Methods will have a status variable to them, meaning if method cant be executed then method will return -1, if executed correctly it will return 0.

1. **Decomposition Diagram** (Used to break program down into components visually. Diagram can have as many components as needed. Defines functionality that will solve the problem – does NOT define a flow of actions)

String program

Input

process

Output

User will enter characters by the command line

Check if input is within limits.

Print out message when string Is input correctly

Display error message when user inputs to much data

Ask user which method they would like to use

User inputs which method they would like to use.

String will become modifiable with different strings.

String will be stored inside of a class and can use any method on string.

1. **Test Strategy**

* Valid Data
* Invalid Data
* files

1. **Test Plan Version 1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Strategy | Test Number | Description | Input | Expected Output | Actual Output | Pass/Fail |
| Invalid Data | 1 | User enters 26 characters |  |  |  |  |
| Invalid Data | 2 | User enters  No data |  |  |  |  |
| Valid Data | 3 | User enters 24 characters |  |  |  |  |

1. **Initial Algorithm**
2. Create a mystring class, it will accept no more than 25 characters.

Will validate input by limiting the amount of characters inputted.

Once string is stored inside class. Methods will be called to modify string.

Will have appropriate get and set functions in order to set characters to the character array.

**Size** function will step through character array and count how many characters are inside of it. Then pass the size of the string to all other methods that need it.

**While chararray[i]!=”/0”**

**Increment counter variable.**

**Return increment variable**

**Addstart(mystring)** function will have access to the character array by parameter. Will make sure string isn’t already at 25 characters.

**If ((mystring+chararray<=24))**

**Do**

**replace chararray with mystring starting at 0**

**end if**

**Addend(mystring)-** will append string parameter to the current string. Will find out where the string ends and add the characters to it. If string exceeds 25 characters combined will display an error message.

If ((mystring+chararray<=24))

Do

add mystring to chararray

end if

**Partstring(startPos,length)-**returns string. Accepts a startpos integer and length integer in its parameters. Function must take into account the size of both strings being combined.

Replpartstring(mystring,startPos)-will use the starting position of the current string to be replaced with the input string.

for i<startpos

do

replace currentstring with mystring

end for.

Replwholestring(mystring)-rewrites current string. Will start at position 0 and delete current string and enter and a new string.

While(mystring[i]!=/0)

Loop until all characters cleared

End while

Rewrite string

Comparestring(mystring)-will compare current value of string with input parameter string.

Loop and compare each string until they reach \0

Will use the differences in asci values to compare each character in both strings.

Initstring()-resets string to null string.

Will continually loop and delete the string and initialize it to null.

loop and delete until ‘\0’ is found then set the string to nullptr.

Setstring(string)-assign string to myString data value.

Create character array and set userinput equal to it.

Getstring()

Return the character array when this function is called

Printstring()

Prints mystring data value to the monitor.

While(mystring!=/0)

Print each each character in the char array.

1. **Test Plan Version 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Strategy | Test Number | Description | Input | Expected Output | Actual Output | Pass/Fail |
| Invalid Data | 1 | User enters 26 characters | Hawurnfjrhejskslnrjdisnwkw | Incorrent input please input 25 characters or less |  |  |
| Invalid Data | 2 | User enters  No data |  | Nothing happens |  |  |
| Valid Data | 3 | User enters 24 characters | Aaaabcdefghiklmnopaaaaa | “Which method would you like to use on inputted string?”  Print menu |  |  |
| Invalid function call | 4 | User enters string to add to current string that exceeds the limit, and vice versa | (using add end function)  Add 2 strings equaling more than 25 characters | Please enter a total of 25 characters to be combined |  |  |
| Invalid part string function call | 5 | User enters invalid starting position for the function | -1 | Error Please enter a number between 0-25 |  |  |
| Invalid reppartstring function call | 6 | User enters invalid starting position for the function | -5 | Error  Please enter a number between 0-25 |  |  |

1. **Code**

1. **Updated Algorithm**
2. **Test Plan Version 3**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Strategy | Test Number | Description | Input | Expected Output | Actual Output | Pass/Fail |
| Invalid Data | 1 |  |  |  |  |  |
| Invalid Data | 2 |  |  |  |  |  |
| Valid Data | 3 |  |  |  |  |  |
| Valid Data | 4 |  |  |  |  |  |
| Valid Data | 5 |  |  |  |  |  |
| Valid Data | 6 |  |  |  |  |  |
| Valid Data | 7 |  |  |  |  |  |

1. **Screenshots**

Test Cases 1-9

1. **Error Log**

|  |  |  |
| --- | --- | --- |
| Error Type | Cause of Error | Solution to Error |
|  |  |  |

1. **Status**

The program works 100% with assumptions in place