



Hacettepe University

Computer Science and Engineering Department

NAME & SURNAME

Candaş Nasıf

ID

21328232

COURSE

BBM203

EXPERIMENT

Assignment II

SUBJECT

Data Structures

DUE DATE

13.11.2015

ADVISORS

R.A. Yasin Şahin

E-MAIL

b21328232@cs.hacettepe.edu.tr

Content

1. Software Using Documentation

- 1.1. Software Usage
- 1.2. Errors

2. Software Design Notes

- 2.1.1. Problem
- 2.1.2. Solution
- 2.2. Algorithm
- 2.3. Data Structures

3. Software Testing Notes

- 3.1. Bugs and Software Reliability

1. Software Using Documentation

1.1. Software Usage

This software is designed a collection. This collection can store double, integer and string types. You can putting, deleting, adding, print, undo and redo operations on this collection. Software take input and output files names from command line.

1.2. Errors

1. If the taken input from input file is not a string, double or integer software print in output file an error message like that

"UNKNOWN COMMAND OR DATA TYPE"

2.If the taken input from input file is "ADD" when the collection has one or zero element software print in output file an error message like that

"ADDITION IGNORED"

3.If the taken input from input file is "REDO" and previous input is "PUT x","DELETE" or"ADD" software print in output file an error message like that

"REDO IGNORED"

4.If the taken input from input file is "DELETE" but list has no element print in output file an error message like that

"DELETION IGNORED"

2. Software Design Notes

2.1. Problem

How can I create a store different types collection?

2.1.Solution

I use LinkedList and this list store string data but I can define real type of this data with functions and I can use it.

2.2 .Problem

How can I stored "UNDO" or "REDO" commands ?

2.2.Solution

I want to use stack but experiment does not allow this so I write a struct and methods for it. Finally I use my own stack.

2.2.ALGORITHM

1.Take arguments from command line for input and output files names.

2.Create a LinkedList and send this list operations method

3.Open the input and output files.

4.Read the input file line by line.

3.While Loop

3.1.If there is "PUT" in the line

3.1.1.Split the line from " " and keep second element of this line

3.1.2.Define data type.

3.1.3.If data type is string,integer or double

3.1.3.1Add data to linkedlist

3.1.3.2 Write "A <TYPE OF DATA> ADDED TO COLLECTION".
to output file

3.1.3.3.Push the line in undo stack.

3.1.4.Else Write "UNKNOWN COMMAND OR DATA TYPE" to output
file

3.2. If there is "DELETE"

3.2.1.If the list is not empty

3.2.1.1Delete the first insert in the collection

3.2.1.2.Write "A <TYPE OF DATA> DELETED" to output file.

3.2.1.3. Push the line in undo stack

3.2.2.Else write "DELETION IGNORED" to output file.

3.3.If there is "ADD"

3.3.1. If the list has two element minimum

3.3.1.1. Add first element to second element of linkedlist ,delete this element and put the this new element to end of the linkedlist.

3.3.1.2.WRITE "A/AN <TYPE OF DATA> ADDED TO A\AN <TYPE OF DATA> AND NEW <TYPE OF DATA> PUT END OF COLLECTION" to output file.

3.3.1.3. Push the line in undo stack

3.3.2.Else write "ADDITION IGNORED" to output file.

3.4.If there is "UNDO"

3.4.1 Check " stack is empty?"and if stack is not empty

3.4.1.1.Call the undo.top and reverse this step.

3.4.1.2.Write "<LAST STEP> REVERSED"to output file

3.4.1.3.Push the undo.top in the redo stack

3.4.1.4.Call the undo.pop and delete last line into the stack.

3.5.If there is "REDO"

3.5.1.If redo stack is not empty and last step is not "DELETE", "ADD" or "PUTT x"

3.5.1.1.Call the redo.top and reverse this step.

3.5.1.2.Write "UNDO REVERSED" to output file

3.5.1.3.Push the redo.top in the undo stack

3.5.1.4.Call redo.pop and delete last line into the stack

3.5.2.Else Write "REDO IGNORED" to output file.

3.6.If there is "PRINT"

3.6.1.Write all elements of linkedlist to output file.

4.Close input and output files.

Data Structers

1.LINKEDLIST

This struct has three data.Link, data and type.Link shows next element of the list.Data is the element and type shows type of element.

2.Stack

Stack has a string array and integer n.

Methods of Stack:

1.Void push(string)"Insert a string in the stack"

2.Void pop()"Delete first added element of stack"

3.String top()“Return last addd element”

4.Bool empty()“If the stack is empty return TRUE else FALSE”

5.Bool full()“If the stack is full return TRUE else FALSE”

6.Int size()“Return size of stack”

3. Software Testing Notes

3.1. Bugs and Software Reliability