CS 202 Fall 2022 - Assignment 7

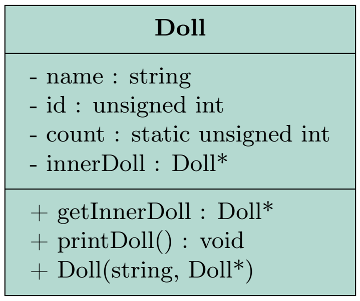
Recursion

Overview

Unlike other assignments, this one will be more of a worksheet/ collection of functions to implement rather than a cohesive program. You can think of it as similar to a library - it is a collection of functions we provide for others to be able to use in their code.

There will be a total of 4 separate recursive functions, each of which should be implemented in the provided header file. Each function will be explained below as well as in the provided video, which can be found here: https://youtu.be/HOOacoUmHAA

UML Diagrams



Doll

- name : string
- id : unsigned int
- count : static unsigned int
- innerDoll : Doll*

+ getInnerDoll : Doll*
+ printDoll() : void
+ Doll(string, Doll*)

Classes and Functions

Below is a list of functions and variables that are important for this assignment. ***Variables are in green***, **functions that you will need to write are in red**, **functions implemented for you already are in blue**, and **functions that are abstract that will be implemented later are in magenta**.

Doll

The Doll class will be used for the final recursive function, **findCenter**. This class represents a Matryoshka Doll (The kind you stack inside each other). This keeps track of a Doll nested within this one along with some extra metadata like the name of the Doll and an id. Everything is already implemented for this class, it will only be used inside the **findCenter** function and main.

- ***string name*** - The name of this Doll for printing purposes
- ***unsigned int id*** - An id for this Doll for printing purposes
- ***static unsigned int count*** - Count of total number of Dolls for generating the ***id***
- ***Doll* innerDoll*** - A pointer to the Doll nested within this Doll. If there is no Doll within this one, then this will point to nullptr
- **Doll* getInnerDoll()** - Getter for the ***innerDoll*** variable
- **void printDoll()** - Prints the Doll's member variables to the terminal
- **Doll(string name, Doll* inside)** - Constructor to set the ***name*** of the Doll and the Doll that is nested inside it

Recursive Functions

For this assignment, we will be implementing four recursive functions. When running the main program, it will ask which one of the four functions to test, which can be selected with a number 1-4. It is encouraged to look at at main to get an idea of the input for each function. All functions are implemented globally. The descriptions for these functions is relatively brief; the goal here is to focus on problem solving

- **int power(const int& base, int power)** - This function should return the base to the given power recursively. For example power(2, 3) returns 8. Remember to consider what the trivial base case is for taking a power and to have the function call itself as part of the return value in the recursive case
- **int findBiggestNumber(int arr[], int index)** - This function returns the biggest value in the passed array up to the given index. This should recursively find the biggest number in the array to the left of the current index and then return the max between that value and the number at the current index. If the array we're looking at consists of only one number, then that case is trivial and the biggest number is that single value.
- **void findTheX(char grid[HEIGHT][WIDTH], const int& x, const int& y)** - Given a 2D grid of characters, this should attempt to find the 'X' char in the array starting at the given x, y position. Recall that when accessing an array, y generally comes before x. This should check the current spot for the 'X' and if not found, tries the 4 adjacent tiles above, below, left, and right of the current spot. If the 'X' is not found, mark the current spot as checked by setting the position to 'O'. If a space is already marked as an 'O', you can return without checking adjacent tiles, since that spot has already been checked. Before trying any one direction, ensure that that spot is in bounds of the 2D array with the given HEIGHT and WIDTH. Once the X is found, print where it was found similar to the sample output. This function does not return a value.
- **void findCenter(Doll* doll)** - This function will attempt to find the center of a stack of Matryoshka nesting dolls. If the given pointer points to a real Doll, this should print its information using the Doll's **printDoll** function and then open this Doll and try to find the center within the Doll contained inside it. Otherwise, if there is no Doll to open (i.e. the Doll is nullptr), then we have reached the center and can stop recursing. Main will already correctly construct a nest of Dolls for you, you can assume no error checking needs to be done.

Compiling / TO-DO

This program consists of only one cpp file with a header file that you will write (recursion.h and recursion.cpp). You will only need to compile the single cpp file: **g++ -o rec -std=c++11 -Wall recursion.cpp** .You need only submit the header file.

Sample Output

power

```
2^10 = 1024
4^0 = 1
```

findBiggestNumber

```
Array = [ 18, 17, 10, 25, 7, 9, 10, 12, 15, 18, 19, 20, 4, 23, 8, 0 ]
Biggest number in the array is 25
```

findTheX

```
---- Maze 1 ----
Found the X at (0, 3)
---- Maze 2 ----
Found the X at (0, 1)
```

findCenter

```
Doll: 3 Red Doll
Doll: 2 Green Doll
Doll: 1 Blue Doll
Doll: 0 Last One
```