

borodark / wsu

Branch: master wsu / csc5050 / ass4 /

Create new file Upload files Find file History

borodark docs

Latest commit 43a7f25 3 minutes ago

..		
include	iterative	36 minutes ago
src	docs	3 minutes ago
CMakeLists.txt	q1	yesterday
README.md	docs	3 minutes ago
q1.out	iterative	36 minutes ago
q2.out	q1	yesterday
q3.out	grammar	yesterday

README.md

CS5050 Assignment 4

To run

Developed and tested in Mac OSX

Install `cmake` and `make` , pull this repo change to this directory, run:

```
cmake . to create Makefile
```

```
make to comple
```

Q1

Using stacks, write a **non-recursive function** `solveTowers()` to solve the towers of Hanoi problem. Furthermore, implement a driver file to test your code. (2 points)

- `src/q1.cpp` - driver and implementation

Results

- 2 and 3 disks
 - **Start**
 - **Destination stack print:**
- 10 disks
 - **Start iteration over 10**
 - **Destination stack print:**

I have also **implemented a recursive way** with **printing of state**:. The printing can be seen here:

- **2 and 3 disk**
- **10 disks**

Q2

- Q2 - [src/q2.cpp](#) driver
- Results: [q2.out](#)

Write an array-based implementation of the ADT list that expands the size of the array of list entries as needed so that the list can always accommodate a new entry. (2 points)

- The [include/ArrayList.h](#) defines behaviour
- The [include/ArrayList.cpp](#) provides implementation

The expand is implemented here: [include/ArrayList.cpp](#) and used during

- insert [include/ArrayList.cpp](#)
- setEntry [include/ArrayList.cpp](#)

Q3

The popular social network Facebook was founded by Mark Zuckerberg and his classmates at Harvard University in 2004. At the time, he was a sophomore studying computer science.

Design and implement an application that maintains the data for a simple social network.

- *The implementation is provided as library, no UI provided.*
- All testing is done from the driver: [src/q3.cpp](#).
 - See output [q3.out](#)

Each person in the network should have a profile that contains the person's name, optional image, current status, and a list of friends.

The profile can be created without joining:

- The [include/Profile.h](#) defines behaviour
- The [include/Profile.cpp](#) provides implementation

The *Member* encapsulates the [Profile](#) and the [list of friends](#)

- The [include/TFBMember.h](#) defines behaviour
- The [include/TFBMember.cpp](#) provides implementation

Your application should allow a user to join the network, leave the network, create a profile, modify the profile, search for other profiles, and add friends. (6 points)

- Interface [include/TheFaceBooks.h](#) provide encapsulation of data and the public interface for:
 - createProfile [include/TheFaceBooks.cpp](#)
 - deleteProfile [include/TheFaceBooks.cpp](#)
 - joinTFB [include/TheFaceBooks.cpp](#)
 - search [include/TheFaceBooks.cpp](#)
 - leaveTFB [include/TheFaceBooks.cpp](#)
 - beFriend [include/TheFaceBooks.cpp](#)
 - unFriend [include/TheFaceBooks.cpp](#)