Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

**ОТЧЕТ**

**по практической работе № 2**

по дисциплине «**Алгоритмы и структуры данных**»

Выполнил

студент гр. 3530904/20005  Лебедев В.В.

Руководитель Самочадина Т.Н.

08.03.2022 г.

Санкт-Петербург

2022 г

Содержание

Тема работы ………………………………..……………………………………..……

Общая постановка задачи…………………………………………………………….

Детальные требования и тест план………………………………………………….

Скриншоты……………………………………………………………………………….

Код программы…………………………………………………………………............

Тема работы:Stack and Queue

Общая постановка задачи:

1. Реализуйте структуру данных «ограниченный» стек

* Используйте шаблон для абстрактного типа данных класса стек:
* На основе шаблона для стека создайте шаблон для реализации структуры данных «ограниченный» стек (через массив).
* Создайте классы StackOverflow и StackUnderflow для работы с двумя исключительными ситуациями, которые могут возникнуть при работе со стеком.
* Создайте класс WrongStackSize для работы с исключительной ситуацией, которая может возникнуть, если в конструкторе стека, реализуемого через массив, неправильно задан размер.
* Выполните тестирование созданного класса.

1. Реализуйте функцию анализа правильности расстановки скобок

* Функция должна возвращать True, если количество открывающих и закрывающих скобок одного типа совпадает, и они имеют правильную вложенность. Допускаются три вида скобок: круглые, квадратные и фигурные.
* Прототип функции: bool checkBalanceBrackets (const char\* text, const int maxDeep); или bool checkBalanceBrackets (const string& text, const int maxDeep); text - анализируемый текст, содержащий скобки maxDeep - максимально возможный уровень вложенности скобок
* В реализации используйте шаблон «ограниченный стек».

1. Реализуйте структуру данных «ограниченная» очередь

* Используйте шаблон для абстрактного типа данных класса очередь
* На основе шаблона для очереди создайте шаблон для реализации структуры данных «ограниченная» очередь (через массив).
* Создайте классы QueueOverflow и QueueUnderflow для работы с двумя исключительными ситуациями, которые могут возникнуть при работе с очередью.
* Создайте класс WrongQueueSize для работы с исключительной ситуацией, которая может возникнуть, если в конструкторе очереди, реализуемой через массив, неправильно задан размер.
* Выполните тестирование созданного класса

Детальные требования и тест план

|  |  |  |  |
| --- | --- | --- | --- |
| Требование | Детальные требования | Данные | Ожидаемый результат |
|  |  |  |  |
|  |  |  |  |
| isEmpty() вставка значения | Если введённого ключа не найдено в словаре - вставка и  Сообщение:  “Data successfully inserted.”  Иначе вставка не производится сообщение:  “This data already in dictionary” | Steck<char> L  1 2 3 4 5  isEmpty() | Сообщение:  “no” |
| isEmpty()  вставка значения | Если введённого ключа не найдено в словаре - вставка и  Сообщение:  “Data successfully inserted.”  Иначе вставка не производится сообщение:  “This data already in dictionary” | Steck<char> L  isEmpty() | Сообщение:  “yes” |
| push  поиск по значению | Если введёный ключ найден в словаре  Сообщение:  “Data have found”  Иначе сообщение:  “Data hasn't been found.” | Steck<char> L  1 2 3 4 5  L.push(5) | Сообщение:  “Data haven't found” |
| pop  поиск по значению | Если введёный ключ найден в словаре  Сообщение:  “Data have found”  Иначе сообщение:  “Data hasn't been found.” | Steck<char> L  1 2 3 4 5  L.spop(5) | Сообщение:  “Data have found” |
|  |  |  |  |

Скриншоты:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Код программы

https://github.com/VensFM/Stack-and-Queue