

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Алгоритмы и структуры данных»
Тема: Деревья

Студент гр. 9304

Атаманов С.Д.

Преподаватель

Филатов Ар. Ю.

Санкт-Петербург

2020

Цель работы.

Ознакомиться со структурой данных «Бинарное дерево» и «Лес», изучить методы работы с ней, и реализовать эти структуры данных, используя язык программирования C++.

Задание.

Вариант 7.

Для заданного леса с произвольным типом элементов:

- получить естественное представление леса бинарным деревом;
- вывести изображение леса и бинарного дерева;
- перечислить элементы леса в горизонтальном порядке (в ширину).

Выполнение работы.

Описание работы программы.

Программа получает на вход скобочное представление леса. Затем, с помощью функции `getConvenientLine()`, происходит удаление лишних пробелов из строкового представления. Затем с помощью функции `isCorrect()` строка с лесом проверяется на валидность: если строка верная, то работа продолжается, если нет, то сообщение об этом выводится на экран и выполнение работы прекращается. С помощью функции `isEmpty()` проверяется, является ли строка пустой. Если это так, то сообщение об этом выводится на экран и работа программы прекращается. Затем, с помощью функции `stringToVectorForest()` лес разбивается на поддеревья. С помощью метода `getForest()` класса `Forest` происходит создание леса из вектора поддеревьев. С помощью метода `convertForestToBinTree()` класса `BinTreeNode`, происходит преобразование леса в бинарное дерево. Затем лес и бинарное дерево выводятся на экран и с помощью метода `widthSearch()` класса `Forest`, выполняется перечисление леса в горизонтальном порядке.

Описание структур данных.

Лес представляет собой линейный однонаправленный список бинарных деревьев. Следующий элемент указывает на структуру «Лес», значениями списков являются Бинарные деревья.

Бинарное дерево представляет собой дерево, имеющее только два потомка — левое и правое бинарное дерево. Листьями этого дерева могут быть как сами деревья, так и какое-либо значение.

Описание алгоритма.

Для считывания леса была реализована функция `getForest()`. Функция принимает вектор строк с поддеревьями. Функция заполняет линейный список поддеревьями с помощью функции `getBinTree()`. Функция возвращает указатель на голову линейного списка.

Бинарное дерево считывается с помощью рекурсивной функции `getBinTree()`. Функция получает на вход итератор на начало строки дерева. В поле `data` записывается значение списка. С помощью условных операторов выполняется заполнение дерева элементами. Функция возвращает указатель на дерево.

Разработанный программный код смотри в приложении А.

Тестирование.

Для тестирования был написан `bash`-скрипт, который считывает тестовые данные из текстовых документов папки `tests` и передает их программе. Сама программа выводит изображение леса, бинарного дерева и перечисления в горизонтальном порядке в файл `output`, выводы сравниваются и заведомо верными результатами и результат сравнения выводится на экран.



```
neyther@neyther:~/Desktop/Учеба/3 Семестр/Алгоритмы и структуры данных/lab3$ ./lab3
(a(b)(c))
Forest
a
  b
  c
Binary tree
a
  b
  c
Elements in horizontal search:
a b c
neyther@neyther:~/Desktop/Учеба/3 Семестр/Алгоритмы и структуры данных/lab3$
```

Рисунок 1 — Пример работы программы

Результаты тестирования смотри в приложении Б.

Выводы.

В ходе выполнения работы были получены навыки работы с бинарными деревьями и лесом.

Была разработана программа на языке C++, которая выполняет считывание строкового представления леса, преобразует строку в лес, преобразует лес в бинарное дерево и выводит на экран лес, бинарное дерево и перечисление элементов леса в горизонтальном порядке.

ПРИЛОЖЕНИЕ А.
ИСХОДНЫЙ КОД ПРОГРАММЫ.

Название файла: main.cpp

ПРИЛОЖЕНИЕ Б.
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ.

Входные данные	Выходные данные	Комментарий
(a(b)(c))	Forest a b c Binary tree a b c Elements in horizontal search: a b c	Простой лес
(a(b)(c(e))(d)) (f(g(i))(h(j)(k(l))))(m(n))	Forest a b c e d f g i h j k l m n Binary tree a b c e d f g i h j k l	Лес, но большего размера

	<p style="text-align: center;">m</p> <p style="text-align: center;">n</p> <p>Elements in horizontal search: a b c d e f g h i j k l m n</p>	
(a (b) (c))	<p>Forest</p> <pre> a b c Binary tree a b c Elements in horizontal search: a b c </pre>	Лес с пробелами
((()))	Empty forest	Лес с пустыми элементами
(a)	<p>Forest</p> <pre> a Binary tree a Elements in horizontal search: a </pre>	Лес из одного элемента
(a(b)(c)	Is incorrect forest	Лес с недостаточным количеством скобок
<Пустой файл>	Is incorrect forest	Задан пустой файл
((((((((())())))))	Is incorrect forest	Пустой лес с неправильным количеством скобок