МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» (МАИ)

Кафедра 319 «Системы интеллектуального мониторинга»

Создание и обслуживание динамической структуры данных «Бинарное дерево»

Пояснительная записка

05235 – 01 81 01

Листов 13

Нормоконтроль

_____ / Молчанова С.И.

«__» _____2019 г.

Задание выполнил

студент группы М3О-235Б-17

_____ /Корнилов А. Н.

«__» _____2019 г.

Преподаватель

_____ / Молчанова С.И.

«__» _____2019 г.

Отметка о защите «___»

2019

Оглавление

Введение

Двоичное дерево поиска — это двоичное дерево, для которого выполняются следующие дополнительные условия (*свойства дерева поиска*)

- Оба поддерева — левое и правое — являются двоичными деревьями поиска;
- У всех узлов *левого* поддерева произвольного узла X значения ключей данных *меньше*, нежели значение ключа данных самого узла X;
- У всех узлов *правого* поддерева произвольного узла X значения ключей данных *больше либо равны*, нежели значение ключа данных самого узла X;

Очевидно, данные в каждом узле должны обладать ключами, на которых определена операция сравнения *меньше*.

Как правило, информация, представляющая каждый узел, является записью, а не единственным полем данных. Однако это касается реализации, а не природы двоичного дерева поиска.

Для целей реализации двоичное дерево поиска можно определить так

- Двоичное дерево состоит из узлов (вершин) — записей вида (data, left, right), где data — некоторые данные, привязанные к узлу, left и right — ссылки на узлы, являющиеся детьми данного узла — левый и правый сыновья соответственно. Для оптимизации алгоритмов конкретные реализации предполагают также определения поля parent в каждом узле (кроме корневого) — ссылки на родительский элемент.
- Данные (data) обладают ключом (key), на котором определена операция сравнения «меньше». В конкретных реализациях это может быть пара (key, value) — (ключ и значение), или ссылка на такую

пару, или простое определен ие операции сравнения на необходимой структуре данных или ссылке на неё.

- Для любого узла X выполняются свойства дерева поиска: key[left[X]] < key[X] ≤ key[right[X]], то есть ключи данных родительского узла больше ключей данных левого сына и нестрого меньше ключей данных правого.

1. Постановка задачи

Разработать алгоритмы для работы с динамической структурой данных «Бинарное дерев». Необходимо обеспечить ввод, обработку, хранение, изменение и удаление информации, чтение из файлов и запись в файлы.

Для реализации данной задачи необходимо:

- реализовать вывод текстового меню программы в консоль;
- разработать и реализовать алгоритм выбора пользователем желательной ему функции программы;
- предусмотреть возможные нарушения порядка эксплуатации и реализовать вывод аварийных сообщений;
- реализовать взаимодействие программы с пользователем, разработать и воплотить алгоритмы вывода полученного результата в удобной для пользователя форме.

2. Описание алгоритма

Была проведена объектно-ориентированная декомпозиция задачи. На рисунке 1 представлена диаграмма классов.

Используемые классы

Tree – класс, не имеющий наследников;

Leave – класс, не имеющий наследников;

Menu – статический класс, не имеющий наследников;

Programm – класс, не имеющий наследников.

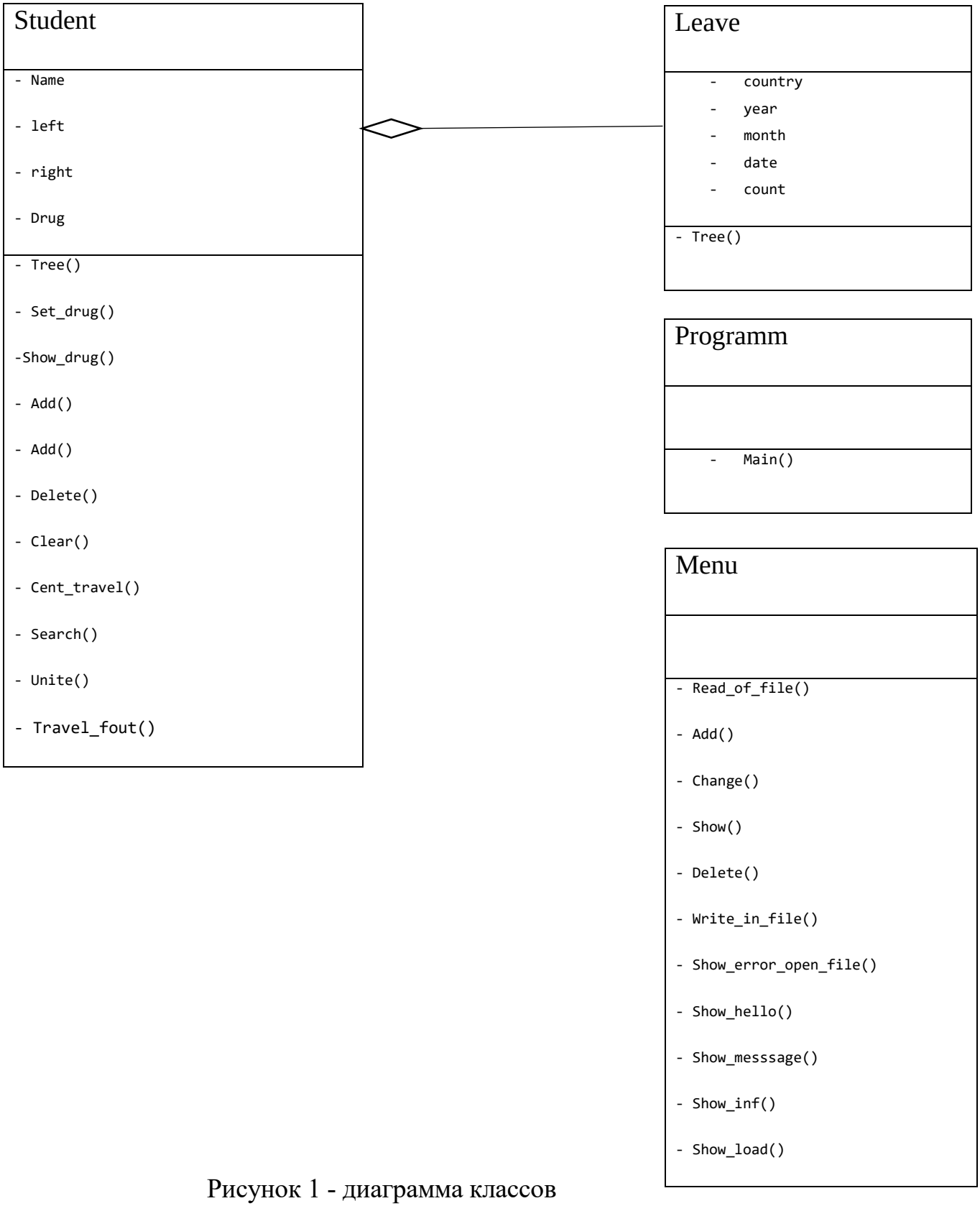

Рисунок 1 - диаграмма классов

В классе Tree описаны следующие компонентные поля

- Name – название препарата;
- Drug – сведения о препарате;
- left – «левый» потомок;
- right – «правый потомок».

И следующие компонентные функции

- Компонентная функция класса Tree: Tree(): конструктор по умолчанию;
- Компонентная функция класса Tree: Set_drug(): не имеет принимаемых параметров и не возвращает никаких значений;
- Компонентная функция класса Tree: Show_drug(): не имеет принимаемых параметров и не возвращает никаких значений;
- Компонентная функция класса Tree: Add(): в качестве параметра принимает ссылку на объект класса Tree и не возвращает никаких значений;
- Компонентная функция класса Tree: Add(): в качестве параметра принимает объект класса string и не возвращает никаких значений;
- Компонентная статическая функция класса Tree: Delete(): в качестве параметров принимает ссылку на объект класса Tree и объект класса string и возвращает значение типа Tree;
- Компонентная функция класса Tree: Clear(): не имеет принимаемых параметров и не возвращает никаких значений;
- Компонентная функция класса Tree: Cent_travel(): не имеет принимаемых параметров и не возвращает никаких значений;
- Компонентная функция класса Tree: Search(): в качестве параметра принимает объект класса name и возвращает значение типа Tree;
- Компонентная статическая функция класса Tree: Unite(): в качестве параметров принимает ссылку на объект класса Tree, ссылку на

объект класса Tree и объект предопределенного класса bool и не возвращает никаких значений;

- Компонентная функция класса Tree: Travel_fout: в качестве параметров принимает объект предопределенного класса string и ссылку на объект класса StreamWriter и не возвращает никаких значений.

В классе Leave описаны следующие компонентные поля:

- country – страна-изготовитель препарата;
- year – год изготовления препарата;
- month – месяц изготовления препарата;
- date – срок годности препарата в месяцах;
- count – количество препаратов.

И следующие компонентные функции

- Компонентная функция класса Leave: Leave(): конструктор по умолчанию.

В классе Programm описана компонентная статическая функция Main(): в качестве параметра принимает ссылку на объект предопределенного класса string и не возвращает никаких значений.

В статическом классе Menu описаны следующие статические функции

Компонентная функция класса Menu: Read_of_file(): в качестве параметра принимает ссылку на объект класса Tree и не возвращает никаких значений;

Компонентная функция класса Menu: Add(): в качестве параметра принимает ссылку на объект класса Tree и не возвращает никаких значений;

Компонентная функция класса Menu: Change(): в качестве параметра принимает ссылку на объект класса Tree и не возвращает никаких значений;

Компонентная функция класса Menu: Show(): в качестве параметра принимает ссылку на объект класса Tree и не возвращает никаких значений;

Компонентная функция класса Menu: Delete(): в качестве параметра принимает ссылку на объект класса Tree и не возвращает никаких значений;

Компонентная функция класса Menu: Write_in_file(): В качестве параметра принимает ссылку на объект класса Tree и не возвращает никаких значений;

Компонентная функция класса Menu: Show_error_open_file(): не имеет принимаемых параметров и не возвращает никаких значений;

Компонентная функция класса Menu: Show_hello(): не имеет принимаемых параметров и не возвращает никаких значений;

Компонентная функция класса Menu: Show_message(): в качестве параметров принимает объект предопределенного класса string, объект предопределенного класса int, объект предопределенного класса int и объект предопределенного класса int и не возвращает никаких значений;

Компонентная функция класса Menu: Show_inf(): не имеет принимаемых параметров и не возвращает никаких значений;

Компонентная функция класса Menu: Show_load(): в качестве параметра принимает объект предопределенного класса int и не возвращает никаких значений.

3. Организация данных

3.1. Организация входных данных

В качестве входных данных в программе используются целочисленные**,** буквенные и символьные значения, вводимые пользователем в консоль по запросу.

3.2. Организация выходных данных

Выходными данными в программе являются

- Пользовательский интерфейс, а также пояснения на русском языке для упрощения навигации пользователя;
- Информация о препаратах.

4. Технические и программные средства

4.1. Технические средства

Техническими средствами, необходимыми для запуска и штатного функционирования программы, являются:

- процессор с тактовой частотой не ниже 1,8 ГГц;
- не менее 512 Мб оперативной памяти;
- монитор SVGA с минимальным разрешением 800x600 пикселей;
- манипулятор типа «мышь» и клавиатура.

4.2. Программные средствами

Для разработки программы STUD необходим следующий инструментарий

- MS VisualStudio 2017 – платформа выбрана для использования языка C++;
- операционная система MS Windows 7 или выше.

5. Результаты работы программы и их оценка

5.1. Объект испытаний

Объектом испытаний является программа Tree.

Для проверки работоспособности была разработана система тестов, проверяющая основной функционал данной программы.

5.2. Цель испытаний

Целью испытаний данной программы является проверка работоспособности при ее эксплуатации пользователем с набором базовых навыков обращения с компьютером.

5.3. Требования к программе

Программа должна обладать достаточной степенью надежности, удобства визуальной части, тестируемостью и доступностью последующих исправлений.

5.4. Методы испытаний

Результаты тестирования программы приведены в Таблице 1

Таблица 1 - Система испытаний работы программы

Назначение теста	Входные данные	Фактическая реакция программы
Проверка реакции программы на ввод пользователем некорректного ключа меню	8	Сообщение об ошибке
Проверка реакции программы на ввод пользователем корректного ключа меню	1	Продолжение программы
Проверка реакции программы на ввод пользователем некорректного года изготовления препарата	7200	Сообщение об ошибке
Проверка реакции программы на ввод пользователем корректного года изготовления препарата	2012	Продолжение программы

Продолжение Таблицы 1

Проверка реакции программы на ввод пользователем некорректного месяца изготовления препарата	00001ь.	Сообщение об ошибке
Проверка реакции программы на ввод пользователем корректного месяца изготовления препарата	12	Продолжение программы
Проверка реакции программы на ввод пользователем некорректного срока годности препарата	двадцать	Сообщение об ошибке
Проверка реакции программы на ввод пользователем корректного срока годности препарата	30	Продолжение программы
Проверка реакции программы на ввод пользователем некорректного количества препарата	0	Сообщение об ошибке
Проверка реакции программы на ввод пользователем корректного количества препарата	2500	Продолжение программы
Проверка реакции программы на ввод пользователем недоступного пути файла	C:\	Сообщение об ошибке

Продолжение Таблицы 1

Проверка реакции программы на ввод пользователем пути файла, где он отсутствует	C:\Users\Admin	Сообщение об ошибке
Проверка реакции программы на ввод пользователем некорректных символов	*	Сообщение об ошибке
Проверка реакции программы на ввод пользователем корректного пути файла	C:\Users\Admin\Desktop	Продолжение работы программы

5.5.Оценка результатов тестирования

Все испытания проведены успешно, результаты работы программы соответствуют ожидаемым.

Заключение

Программа выполнена в полном соответствии с поставленной задачей, была проверена и отлажена. Эксперименты показали состоятельность алгоритма. Приложение может иметь практическое применение. Программа полностью работоспособна.

Выполнение курсовой работы помогло закрепить материал и знания, полученные на лекциях и лабораторных работах, а также были получены практические навыки в написании полноценных функциональных приложений.

Список литературы

- Курс лекций по дисциплине «Объектно-ориентированное программирование» Молчанова С.И, Сдобнов А. Г.;
- Научно-популярное издание Герберт Шилдт C# 4.0: полное руководство