Студент: Шитов Степан Вадимович

Группа: КИ17-07б

Задание: Формирование тестовых наборов данных

Вариант: 11-3

|  |  |  |
| --- | --- | --- |
| Обобщенный артефакт, используемый в задании | Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив) | Параметры, общие для всех альтернатив |
| 11. Растения | Деревья (возраст – длинное целое) Кустарники (месяц цветения – перечислимый тип) | Название |

Для описания базовых альтернатив будут использоваться типы данных:

long int – описание возраста деревьев;

enum – описание месяца цветения кустарников (enum Months {MONTH\_JANUARY, MONTH\_FEBRUARY, …};).

|  |  |  |  |
| --- | --- | --- | --- |
| Действие | Входные данные | Выходные данные | Примечание |
| Файл пуст. Чтение данных из файла, вывод данных в файл. | In\_1.txt --- Пустой файл | Out\_1.txt --- Пустой файл. | При чтении файла будет выведено предупреждение о наличии пустого файла, под элемент не будет выделено памяти. |
| В файле находится один объект. Чтение данных из файла, вывод данных в файл. | In\_2.txt --- Один объект:  Type: “tree”; Age: 5; Name: “treeName”; | Out\_2.txt --- Будет записан один объект: Type: “tree”; Age: 5; Name: “treeName”; | Так как все поля заполнены в соответствии с обозначенными типами данных, данные будут выведены в файл. |
| В файле находится 10 объектов. Чтение данных из файла, вывод данных в файл. | In\_3.txt --- 10 объектов:  Type: “tree”; Age: 5; Name: “Ash”;  Type: “tree”; Month: MONTH\_JUNE; Name: “Oak”;  Type: “tree”; Age: 23; Name: 123;  Type: “bush”; Month: MONTH\_APRIL; Name: “Cherry”;  Type: “bush”; Month: MONTH\_JULE; Name: “Basil”;  Type: “bush”; Month: MONTH\_SEPTEMBER; Name: “ivy”;  Type: “bush”; Month: 10; Name: “Basil2”;  Type: “tree”; Age: ; Name: “oak2”;  Type: “tree”; Age: “done”; Name: “ivy”;  Type: 2; Age: 5; Name: “treeName”; | Out\_3.txt --- 3 объекта:  Type: “tree”; Age: 5; Name: “Ash”;  Type: “bush”; Month: MONTH\_APRIL; Name: “Cherry”;  Type: “bush”; Month: MONTH\_SEPTEMBER; Name: “ivy”; | Остальные случаи будут отброшены по причине не соответствия данных заданному типу. |
| В файле находятся данные о 5 деревьях и 3 кустах. Чтение данных из файла, удаление из контейнера всех деревьев, вывод данных в файл. | In\_4.txt --- 8 объектов:  Type: “tree”; Age: 5; Name: “Ash”;  Type: “tree”; Age: ; Name: “Ash1”;  Type: “tree”; Age: “7”; Name: ;  Type: “tree”; Age: 8; Name: “Ash2”;  Type: “tree”; Age: 9; Name: “Ash3”;  Type: “bush”; Month: MONTH\_APRIL; Name: “Cherry”;  Type: “bush”; Month: MONTH\_SEPTEMBER; Name: “ivy”;  Type: “bush”; Month: MONTH\_JULE; Name: “Cherry2”; | Out\_4.txt --- 2 объекта:  Type: “bush”; Month: MONTH\_APRIL; Name: “Cherry”;  Type: “bush”; Month: MONTH\_SEPTEMBER; Name: “ivy”; | В ходе первоначальной обработки будет отброшено 3 объекта по причине не соответствия типов, затем после удаления останется всего 2. |
| В файле находятся данные о 5 деревьях и 3 кустах. Чтение данных из файла, удаление из контейнера всех кустов, вывод данных в файл. | In\_5.txt --- 8 объектов:  Type: “tree”; Age: 5; Name: “Ash”;  Type: “tree”; Age: ; Name: “Ash1”;  Type: “tree”; Age: “7”; Name: ;  Type: “tree”; Age: 8; Name: “Ash2”;  Type: “tree”; Age: 9; Name: “Ash3”;  Type: “bush”; Month: MONTH\_APRIL; Name: “Cherry”;  Type: “bush”; Month: MONTH\_SEPTEMBER; Name: “ivy”;  Type: “bush”; Month: MONTH\_JULE; Name: “Cherry2”; | Out\_5.txt --- 3 объекта:  Type: “tree”; Age: 5; Name: “Ash”;  Type: “tree”; Age: 8; Name: “Ash2”;  Type: “tree”; Age: 9; Name: “Ash3”; | В ходе первоначальной обработки будет отброшено 3 объекта по причине не соответствия типов, затем после удаления останется всего 3. |
| В файле находятся данные об 1 дереве. Чтение данных из файла, удаление из контейнера дерева, вывод данных в файл. | In\_6.txt --- 1 объект:  Type: “tree”; Age: 5; Name: “Ash”; | Out\_6.txt --- Пустой файл. | При записи из файла будет выделена память под объект. В ходе удаления необходимо очистить узел списка от данных с помощью команды free(\*указатель\*)(Я.П. С) |

Занятие 2.

Задание: Редактирование исходных текстовых наборов данных в соответствии с замечаниями.

В ходе заполнения входных файлов могут быть введены только следующие данные:

* Данные, не соответствующие заданному типу. Например, в поле принимающее числовое значение пользователь может подать строку или перечислимый тип.
* Пустая строка. Например, пользователь может ну ввести никакое значение и сохранить “”.

При поступлении таких данных, программа не должна сохранять данную запись и вывести предупреждение. Также, для того чтобы избежать ввода произвольных данных (напр. при вводе сорта дерева), можно представить данный параметр в виде перечисляемого типа, который при вводе неверного значения будет также отвергать введенные данные.

|  |  |  |  |
| --- | --- | --- | --- |
| В данном случае будут рассмотрены различные варианты входных данных, описанных выше. | In\_7.txt  //1.a - Type: “tree”; Age: 5; Name: “Ash”;  //1.b - Type: 1; Age: 5; Name: “Ash”;  //1.c - Type: “”; Age: 5; Name: “Ash”;  //1.d - Type: “tree”; Age: “five”; Name: “Ash”;  //1.e - Type: “tree”; Age:; Name: “Ash”;  //1.f - Type: “tree”; Age: 5; Name:1;  //1.g - Type: “tree”; Age: 5; Name: “”;  //2.a - Type: “bush”; Month: MONTH\_APRIL; Name: “Cherry”;  //2.b - Type: 2; Month: MONTH\_APRIL; Name: “Cherry”;  //2.c - Type: “”; Month: MONTH\_APRIL; Name: “Cherry”;  //2.d - Type: “bush”; Month: 4; Name: “Cherry”;  //2.e - Type: “bush”; Month: ; Name: “Cherry”;  //2.f - Type: “bush”; Month: MONTH\_APRIL; Name: 3;  //2.g - Type: “bush”; Month: MONTH\_APRIL; Name: “”; | Out\_7.txt --- В выходной файл были записаны только “идеальные” записи – 1.a & 2.a  //1.a - Type: “tree”; Age: 5; Name: “Ash”;  //2.a - Type: “bush”; Month: MONTH\_APRIL; Name: “Cherry”; | При записи в файл было получено два файла, так как остальные не прошли проверку на начальном этапе. Но случай с записью //2.d можно предусмотреть, добавив обработчик, использующий static\_cast<…>(…) при проверке нового значения. |

Далее по этапам работы программы:

1. Считывание и вывод на экран элементов списка будет производиться за счет работы с двунаправленным кольцевым списком. При добавлении элемента, для его хранения будет выделяться участок памяти, содержащий в себе необходимые поля для хранения значений, а также указатели на следующий и предыдущий элементы. Вывод элементов на экран будет осуществляться с помощью последовательного прохождения данного списка и считывания необходимых полей.
2. При добавлении элемента, для его хранения будет выделяться участок памяти, содержащий в себе необходимые поля для хранения значений, а также указатели на следующий и предыдущий элементы.
3. При добавлении нового семейства, пользователь должен будет назначить имя и тип для каждого из полей. Непосредственно в коде.
4. При добавлении нового действия, разработчик столкнется с задачей, при выполнении которой ему придется самому вносить изменения в программу, а также создавать алгоритм функции.
5. Сортировку можно производить по многим характеристикам, но самой очевидной является сортировка по алфавиту, производимая по любому из параметров со строковым типом данных. Алгоритм состоит в том, что каждый элемент посимвольно будет сравниваться с другими, тем самым меняя свое положение.
6. Удаление по признаку можно производить за счет обхода всего списка и удаления элементов всем известным способом (изменение указателей на следующий и предыдущий элемент).