

Индивидуальное задание 6

ООП: Операции со строками

Содержание отчета по лабораторной работе:

1. Титульный лист (дисциплина, тема работы, вариант, ФИО, группа)
2. Общая постановка задачи (общие требования и задание выполняемого варианта)
3. Детальные требования и тест план
4. Программа

Общие требования

Программа должна обеспечивать:

- Ввод имени входного файла, содержащего информацию для обработки;
- Ввод имени выходного файла, содержащего результаты работы программы;
- Проверку корректности информации, содержащейся во входном файле;
- Обработку информации в соответствии с вариантом задания;
- Обработку и инициализацию исключительных ситуаций (связанных, например, с проверкой значения полей).

Программа должна быть написана в соответствии со стандартом программирования: C++ Programming Style Guidelines (<http://geosoft.no/development/cppstyle.html>).

Программа должна быть сделана в двух вариантах:

1. С использованием класса `std::string` из STL
2. С использованием собственного класса `String`

Для выполнения задания можно использовать информацию - [string - C++ Reference \(cplusplus.com\)](http://cplusplus.com/string)

Собственный класс `String` должен содержать:

- Конструктор по умолчанию, создающий пустую строку
- Конструктор, создающий строку из строки в стиле C
- Конструкторы копирования и перемещения
- Деструктор
- Операторы копирующего и перемещающего присваивания
- Метод `compare` для сравнения двух строк в лексикографическом порядке

Перед использованием класса в основном задании его необходимо отладить.

Функцию сортировки из п. 1 вашего варианта сделать шаблонной. Можно использовать как компараторы, так и перегруженные операторы сравнения.

Вариант 1. Person

Структура входного файла:

Первая строка – количество записей файла

Каждая следующая строка – фамилия, имя, год рождения.

Между словами может быть произвольное число пробелов. Другие разделители не допускаются.

Структура выходного файла:

Первая строка – количество записей файла

Каждая следующая строка – фамилия, имя, год рождения. Между словами один пробел.

Правила записи входной информации:

Фамилия, имя – записываются латинскими буквами, начинаются с большой буквы.

Внутри фамилии может быть дефис. Фамилия не может состоять менее чем из 3 и более чем 20 символов.

Имя не может быть короче 2 и более 10 символов.

Год рождения – четыре цифры.

Правила ввода-вывода:

Имена файлов вводятся из стандартного потока `std::cin`.

Строка файла читается целиком, затем проверяется корректность информации.

Вывод в файл отсортированных данных также выполняется построчно.

Вариант 1.1

1. Упорядочить корректно введенную информацию по фамилии, имени, году рождения (сортировка по методу выбора).
2. Определить сколько различных фамилий, содержится в файле.
3. Определить число персон с совпадающими фамилиями и именами

Вариант 1.2

1. Упорядочить корректно введенную информацию по году рождения, фамилии, имени. (сортировка по методу ставок).
2. Определить годы, в который родилось наибольшее число персон.
3. Определить, есть ли персоны с одинаковыми фамилиями, родившиеся в один год.

Вариант 1.3

1. Упорядочить корректно введенную информацию по имени, году рождения, фамилии (сортировка по методу «поплавка»).
2. Определить сколько различных имен встречается в файле.
3. Определить, есть ли одинаковы имена у персон, родившихся в один год.

Вариант 1.4

1. Упорядочить корректно введенную информацию по имени, году рождения, фамилии (сортировка по методу вставок).
2. Определить сколько различных имен встречается в файле.
3. Определить, год, в который родилось больше всего персон с одинаковыми именами и фамилиями.

Вариант 2. Subscriber

Структура входного файла:

Первая строка – количество записей файла

Каждая следующая строка – фамилия, имя, номер телефона, код тарифа.

Между словами может быть произвольное число пробелов. Другие разделители не допускаются.

Структура выходного файла:

Первая строка – количество записей файла

Каждая следующая строка – фамилия, имя, номер телефона, тариф. Между словами один пробел.

Правила записи входной информации:

Фамилия записывается латинскими буквами, начинается с большой буквы.

Внутри фамилии может быть дефис. Фамилия не может состоять менее чем из 3 и более чем 20 символов.

Имя записывается одной большой буквой (первая буква имени), за которой следует точка.

Номер телефона записывается в форме: +x(xxx)xxx-xx-xx

x- десятичная цифра.

Тариф – код тарифа – три цифры.

Правила ввода-вывода:

Имена файлов вводятся из стандартного потока `std::cin`.

Строка файла читается целиком, затем проверяется корректность информации.

Вывод в файл также выполняется построчно.

Вариант 2.1

1. Упорядочить корректно введенную информацию по фамилии, имени, тарифу, номеру телефона (сортировка по методу выбора).
2. Определить сколько различных абонентов, содержится в файле.
3. Определить число абонентов, имеющих несколько номеров.

Вариант 2.2

1. Упорядочить корректно введенную информацию по тарифу, фамилии, имени. номеру телефона (сортировка по методу ставок).
2. Определить тариф, которыми пользуются наибольшее число абонентов.
3. Определить, есть ли абоненты с одинаковыми фамилиями, пользующиеся одним тарифом.

Вариант 2.3

1. Упорядочить корректно введенную информацию по тарифу, фамилии, имени, номеру телефона (сортировка по методу «поплавка»).
2. Определить сколько различных тарифов встречается в файле.
3. Определить тариф, которыми пользуются наибольшее число абонентов.

Вариант 2.4

1. Упорядочить корректно введенную информацию по фамилии, имени, тарифу, номеру телефона (сортировка по методу вставок).
2. Определить сколько абонентов пользуются разными тарифами.
3. Определить, сколько абонентов пользуются разными номерами по одному тарифу.

Вариант 3. BankCard

Структура входного файла:

Первая строка – количество записей файла

Каждая следующая строка – имя держателя карты, номер карты, срок действия карты.

Между словами может быть произвольное число пробелов. Другие разделители не допускаются.

Структура выходного файла:

Первая строка – количество записей файла

Каждая следующая строка – имя держателя карты, номер карты, срок действия карты.

Между словами один пробел.

Правила записи входной информации:

Имя держателя карты – имя фамилия записываются большими латинскими буквами, между именем и фамилией должен быть один пробел.

Фамилия не может состоять менее чем из 3 и более чем 15 символов.

Имя не может быть короче 2 и более 10 символов.

Номер карты – 16 цифр. Первая цифра номера – идентификатор типа финансовой организации.

Срок действия карты – mm/yy

Правила ввода-вывода:

Имена файлов вводятся из стандартного потока std::cin.

Строка файла читается целиком, затем проверяется корректность информации.

Вывод в файл также выполняется построчно.

Вариант 3.1

1. Упорядочить корректно введенную информацию по фамилии, имени, номеру карты, сроку действия карты (сортировка по методу выбора).
2. Определить сколько различных фамилий, содержится в файле.
3. Определить число владельцев с совпадающими фамилиями и именами

Вариант 3.2

1. Упорядочить корректно введенную информацию по фамилии, имени, номеру карты, сроку действия карты (сортировка по методу ставок).
2. Определить число карт, срок действия заканчивается в текущем году.
3. Определить, есть ли владельцы с одинаковыми фамилиями, но разными именами.

Вариант 3.3

1. Упорядочить корректно введенную информацию по фамилии, имени, сроку действия карты, номеру карты (сортировка по методу «поплавка»).
2. Определить сколько владельцы имеют несколько карт.
3. Определить идентификатор типа финансовой организации, у которой представлено максимальное число карт.

Вариант 3.4

1. Упорядочить корректно введенную информацию по фамилии, имени, номеру карты, сроку действия карты (сортировка по методу вставок).
2. Определить сколько держателей карт имеют карты различных типов финансовых организаций.
3. Определить сколько владельцев имеют все карты с закончившимся сроком действия.