|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный  исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №7  по дисциплине «Языки программирования»  Вариант 5 | | |
|  | | Работу выполнил  студент группы БАС-1,2023 2 курса  Иванов С.A  «13» Апреля 2025 г. |
| Работу проверил  Ракина В.Д  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2025 г. |
| Пермь 2025 | | |

СОДЕРЖАНИЕ

[Постановка задачи 4](#_Toc195441757)

[Алгоритм решения 4](#_Toc195441758)

[Тестирование 4](#_Toc195441759)

[Код программы 5](#_Toc195441760)

[Постановка задачи 6](#_Toc195441761)

[Алгоритм решения 6](#_Toc195441762)

[Тестирование 6](#_Toc195441763)

[Код программы 7](#_Toc195441764)

[Постановка задачи 8](#_Toc195441765)

[Алгоритм решения 8](#_Toc195441766)

[Тестирование 8](#_Toc195441767)

[Код программы 9](#_Toc195441768)

[Постановка задачи 10](#_Toc195441769)

[Алгоритм решения 10](#_Toc195441770)

[Тестирование 10](#_Toc195441771)

[Код программы 11](#_Toc195441772)

[Постановка задачи 12](#_Toc195441773)

[Алгоритм решения 12](#_Toc195441774)

[Тестирование 12](#_Toc195441775)

[Код программы 13](#_Toc195441776)

[Постановка задачи 14](#_Toc195441777)

[Алгоритм решения 14](#_Toc195441778)

[Тестирование 14](#_Toc195441779)

[Код программы 15](#_Toc195441780)

[Постановка задачи 16](#_Toc195441781)

[Алгоритм решения 16](#_Toc195441782)

[Тестирование 16](#_Toc195441783)

[Код программы 17](#_Toc195441784)

[Постановка задачи 18](#_Toc195441785)

[Алгоритм решения 18](#_Toc195441786)

[Тестирование 19](#_Toc195441787)

[Код программы 19](#_Toc195441788)

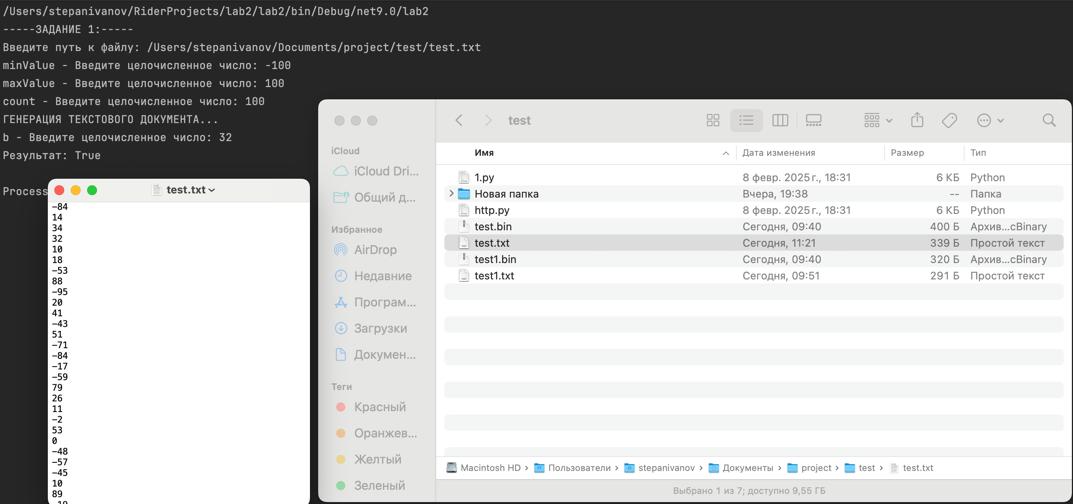
# Постановка задачи

Для заданного файла возвратить true, если он содержит заданное число b, и false в противном случае.

# Алгоритм решения

Создан класс FilesAndCollection, в котором создан метод FillWithRandomNumbers, который генерирует текстовый документ случайными числовыми значениям: по одному числу в строке. Этот метод принимает: 1. count – кол-во строк в текстовом документе; 2. minValue – минимальное значения для Random; 3. maxValue – максимальное значение для Random; 4. filePath – ссылка на текстовый документ. Создается объект random в классе random, с помощью которого программа будет генерировать числа, создается объект f в классе StreamWriter с параметром ссылка на файл, в цикле генерируется числа и через объект f это числа добавляются в текстовый файл. Для решения данной задачи в классе прописан метод ContainsNumber, который в параметрах принимает заданное число b и ссылка на файл он проходит по всем строкам в файле и если есть совпадение происходит возврат True иначе False. В main() создается объект checker в классе InputValidator (Мой собственный класс, который проверяет ввод на валидность, в данном случае, чекер будет проверять, то что ссылка на файл существует). После через этот же чекер пользователь вводит minValue, maxValue, count и вызывается метод FillWithRandomNumbers, который генерирует текстовый документ, после пользователь вводит число b через checker и вызывается метод ContainsNumber.

# Тестирование



# Код программы

https://github.com/6ermutka/yplab/tree/master/C%232

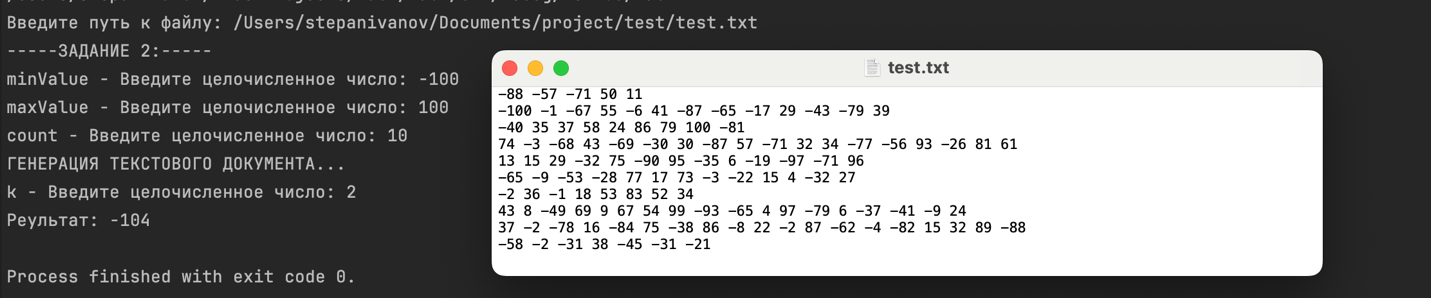
# Постановка задачи

Вычислить сумму элементов, которые кратны заданному числу k.

# Алгоритм решения

Создан класс FilesAndCollection, в котором создан метод FillWithRandomNumbersFull, который генерирует текстовый документ случайными числовыми значениям: по случайному кол-во числ в строке. Этот метод принимает: 1. count – кол-во строк в текстовом документе; 2. minValue – минимальное значения для Random; 3. maxValue – максимальное значение для Random; 4. filePath – ссылка на текстовый документ. Создается объект random в классе random, с помощью которого программа будет генерировать числа, создается объект f в классе StreamWriter с параметром ссылка на файл, в цикле генерируется числа и через объект f это числа добавляются в текстовый файл. Для решения данной задачи в классе прописан метод SumOfMultiples, который в параметрах принимает заданное число k и ссылка на файл, он проходит по всем строкам в файле и если число кратно k, происходит суммирование. В main() создается объект checker в классе InputValidator (Мой собственный класс, который проверяет ввод на валидность, в данном случае, чекер будет проверять, то что ссылка на файл существует). После через этот же чекер пользователь вводит minValue, maxValue, count и вызывается метод FillWithRandomNumbersFull, который генерирует текстовый документ, после пользователь вводит число k через checker и вызывается метод SumOfMultiples.

# Тестирование



# Код программы

https://github.com/6ermutka/yplab/tree/master/C%232

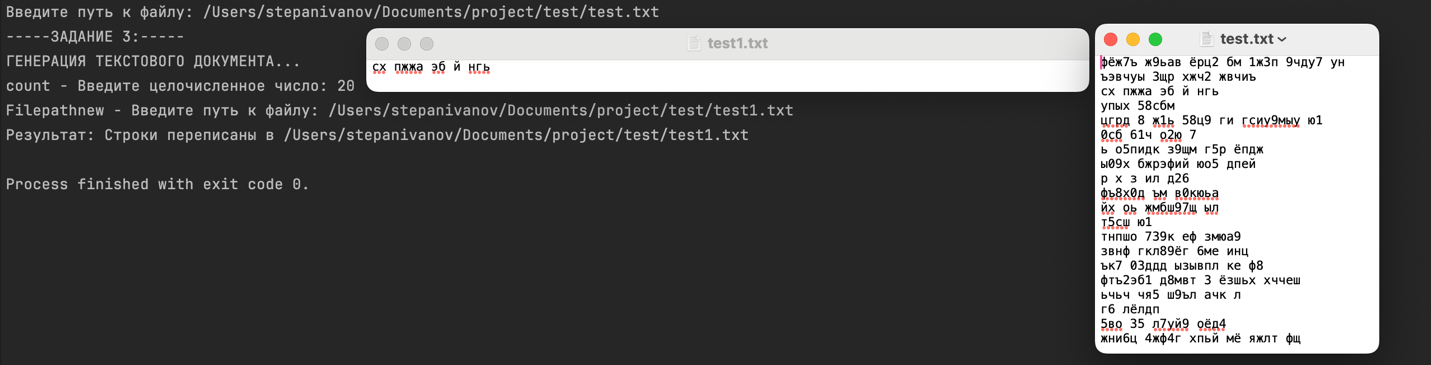
# Постановка задачи

Переписать в другой файл строки, в которых нет цифр.

# Алгоритм решения

Создан класс FilesAndCollection, в котором создан метод GenerateTextFile, который генерирует текстовый документ случайными символами: по случайному кол-во чисел в строке. Этот метод принимает: 1. count – кол-во строк в текстовом документе; 2. filePath – ссылка на текстовый документ. Создается объект random в классе random, с помощью которого программа будет генерировать кол-во слов в строке, также инициализирован и описан массив типа char с русским алфавитом и цифр, создается объект f в классе StreamWriter с параметром ссылка на файл, в цикле берутся случайным образом символы из массива и через объект f это символы добавляются в текстовый файл. Для решения данной задачи в классе прописан метод CopyLinesWithoutDigits, который в параметрах принимает ссылка на исходный файл и ссылка на новый файл, он проходит по всем строкам в файле и если чисел в строке нет(это проверяется через IsDigit), происходит переписывание через объект f2 созданный в классе StreamWriter . В main() создается объект checker в классе InputValidator (Мой собственный класс, который проверяет ввод на валидность, в данном случае, чекер будет проверять, то что ссылка на файл существует). После через этот же чекер пользователь вводит ссылку на новый файл и вызывается метод GenerateTextFile, который генерирует текстовый документ, после вызывается метод CopyLinesWithoutDigits.

# Тестирование



# Код программы

https://github.com/6ermutka/yplab/tree/master/C%232

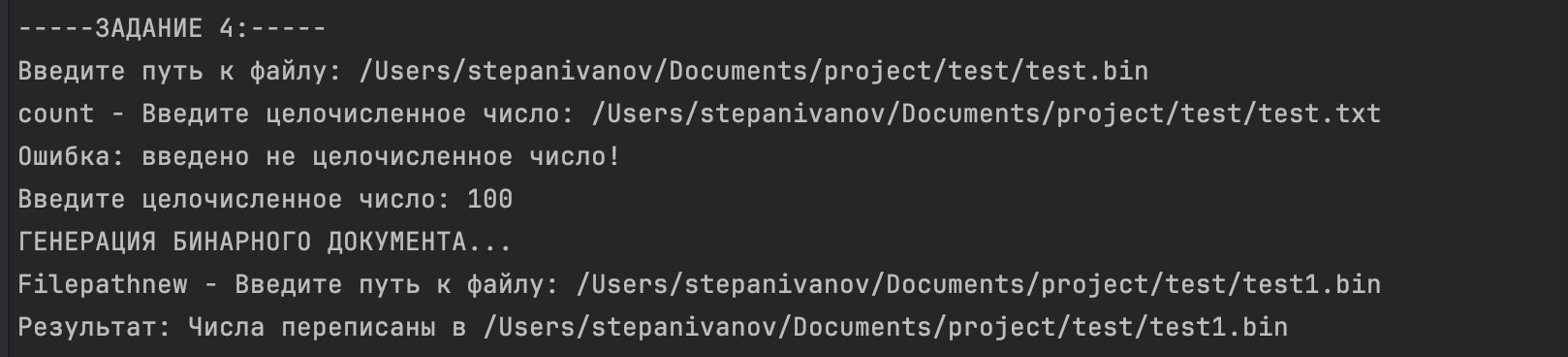
# Постановка задачи

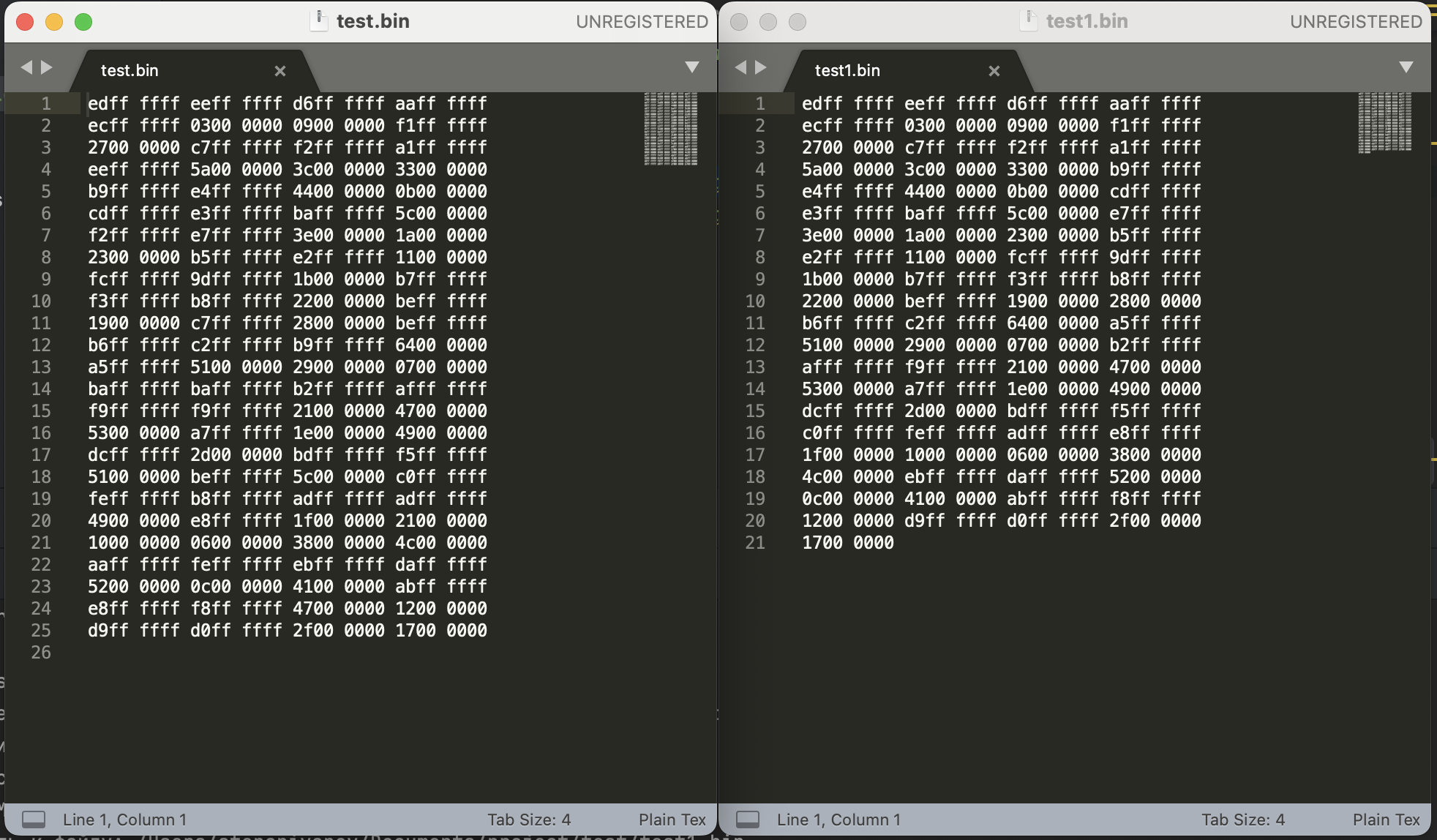
Из исходного файла получить новый файл, исключив повторные вхождения чисел.

# Алгоритм решения

Создан класс FilesAndCollection, в котором создан метод GenerateBinaryFile, который генерирует бинарный документ случайными числовыми значениям в одной строке. Этот метод принимает: 1. count – кол-во чисел в бинарном документе; 2. filePath – ссылка на текстовый документ. Создается объект random в классе random, с помощью которого программа будет генерировать числа, создается объект f в классе BinaryWriter с параметром ссылка на файл, в цикле генерируется числа (от -100 до 101) и через объект f это числа добавляются в бинарный файл. Для решения данной задачи в классе прописан метод GenerateUniqueBinaryFile, который в параметрах принимает ссылку на исходный бинарный файл и ссылку на новый бинарный файл. Создается f в классе BinaryReader, с помощью которого программа будет считывать числа в бинарном файле, в первом цикле все числа раскодируются с помощью ReadInt32 и записываются в массив, во втором массиве происходит проверка чисел, если он не уникальный, то он не добавляется в список уникальных чисел, после этот список уникальных чисел происходит запись нового бинарного файла через объект f1 в классе BinaryWriter. В main() создается объект checker в классе InputValidator (Мой собственный класс, который проверяет ввод на валидность, в данном случае, чекер будет проверять, то что ссылка на файл существует). После через этот же чекер пользователь вводит ссылку на новый файл и вызывается метод GenerateBinaryFile, который генерирует бинарный документ, после пользователь вводит число b через checker и вызывается метод GenerateUniqueBinaryFile.

# Тестирование





# Код программы

https://github.com/6ermutka/yplab/tree/master/C%232

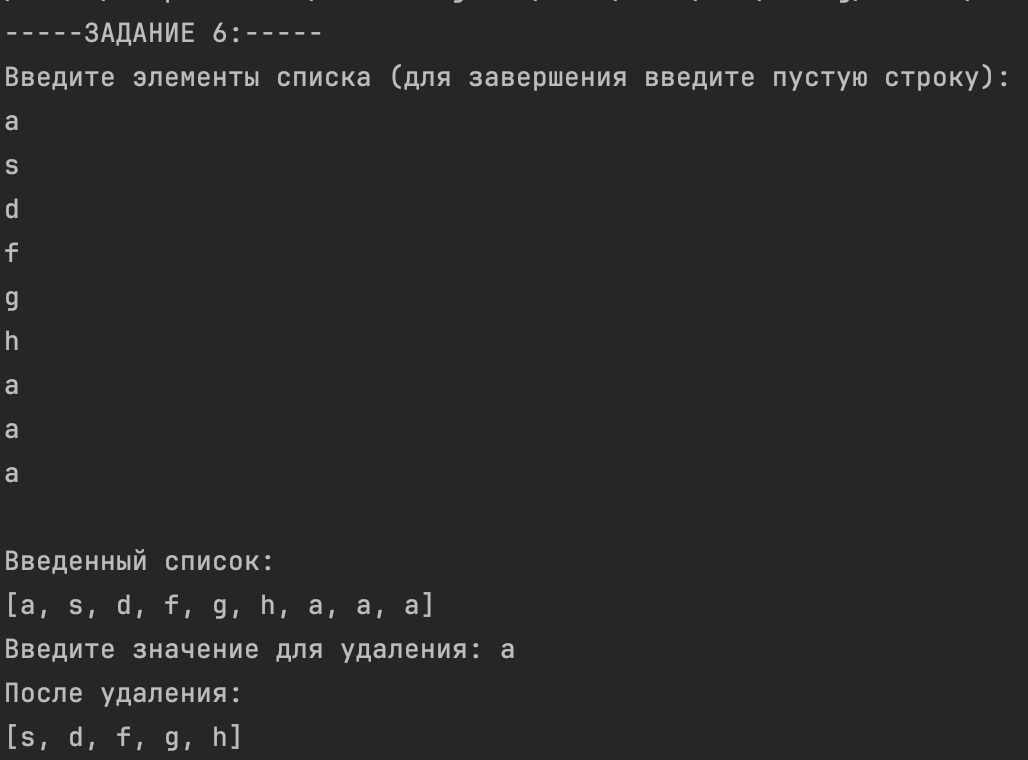
# Постановка задачи

Составить программу, которая удаляет из списка L за каждым вхождением элемента E один элемент если такой есть, и он отличен от E.

# Алгоритм решения

Создан класс FilesAndCollection, в котором инициализировано поле list типа List<string>, который будет хранить коллекцию строк, после в конструкторе FilesAndCollection(List<string> inputList) описывается поле. После создан метод RemoveAllOccurrences, который удаляет все вхождения заданного элемента пользователем. Этот метод принимает: 1. value – элемент, который надо удалить. В цикле программа проходит все элементы списка и если элемент совпадает с заданным элементом, удаляет его. Также в классе перегружен ToString() для вывода списка. В main() пользователь вводит строки, которые добавляются в userList до ввода пустой строки. Создается обьект list, передавая userList в конструктор и через ToString список выводится. После пользователь вводит значение valueToRemove и вызывается RemoveAllOccurrences с параметром valueToRemove. Обновленный список снова выводится через ToString().

# Тестирование



# Код программы

https://github.com/6ermutka/yplab/tree/master/C%232

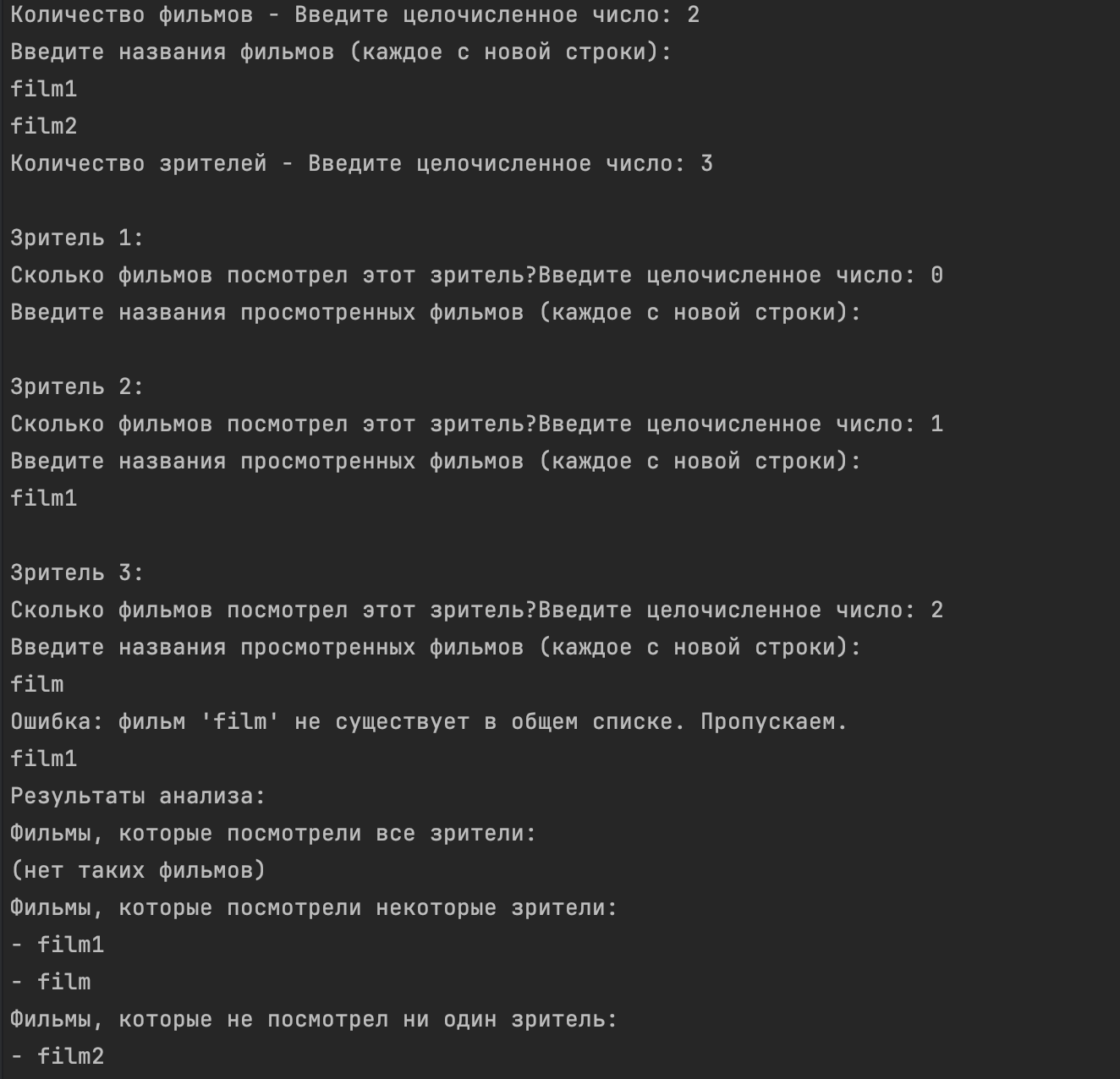
# Постановка задачи

Есть перечень фильмов. Определить для каждого фильма, какие из них посмотрели все n зрителей, какие – некоторые из туристов, и какие – никто из туристов.

# Алгоритм решения

Создан класс FilesAndCollection, в котором инициализировано поле allMovies типа HashSet<string> - содержит все существующие фильмы и viewers типа List<HashSet<string> - список зрителей, где каждый зритель представлен множеством просмотренных фильмов, после в конструкторе FilesAndCollection описывается поля. После создан метод AnalyzeMovies, который считывает фильмы. Метод для начала вычисляет фильмы просмотренные всеми зрителями и сохраняет в watchedByAll типа HashSet, с помощью IntersecWith, дальше фильмы, просмотренные хотя бы одним зрителем с помощью UnionWith, а после фильмы, которые никто не посмотрел с помощью ExceptWith. В main() пользователь вводит фильмы в HashSet и создает список зрителей типа HashSet и сохраняет фильмы, которые посмотрели пользователи и вызывается метод AnalyzeMovies.

# Тестирование



# Код программы

https://github.com/6ermutka/yplab/tree/master/C%232

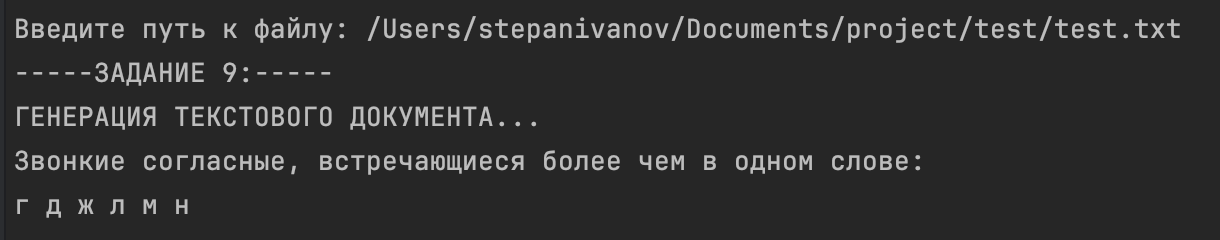
# Постановка задачи

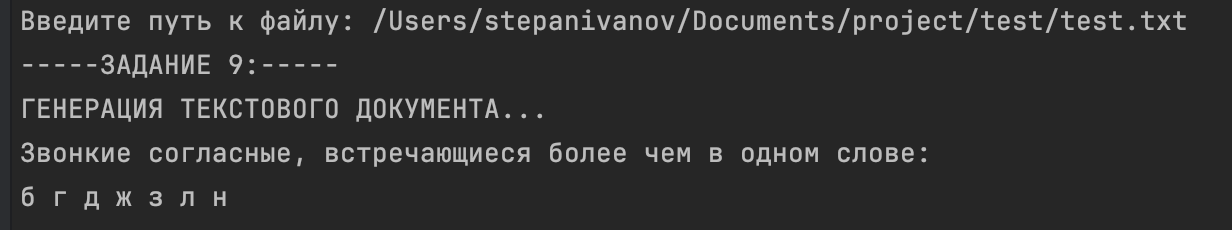
Файл содержит текст на русском языке. Напечатать в алфавитном порядке все звонкие согласные буквы, которые входят ровно в одно слово.

# Алгоритм решения

Создан класс FilesAndCollection, в котором инициализировано поле filePath типа string который будет хранить ссылку на текстовый файл, после в конструкторе описывается поле. После создан метод GenerateTextFileFOR9, генерирует случайный текст в одну строку. В этом методе с помощью random выбираются случайным образом символы из массива chars и после записываются в файл. Также прописан метод ProcessFile, который обрабатывает файл – Сначала разбивает текст на слова, после инициализирует HashSet(allConsonants – хранит все уникальные звонкие согласные, repeatedConsonants – хранит согласные, встречающиеся в нескольких словах), после происходит анализ слов: для каждого слова создается временный набор согласных, которые проверяются на наличие в основном наборе и из всех согласных исключаются те, что встречались более 1 раза, оставшиеся согласные сортируются в алфавитном порядке. В main() пользователь вводит ссылку на файл с помощью чекера и вызываются вышеперечисленные методы.

# Тестирование





# Код программы

https://github.com/6ermutka/yplab/tree/master/C%232

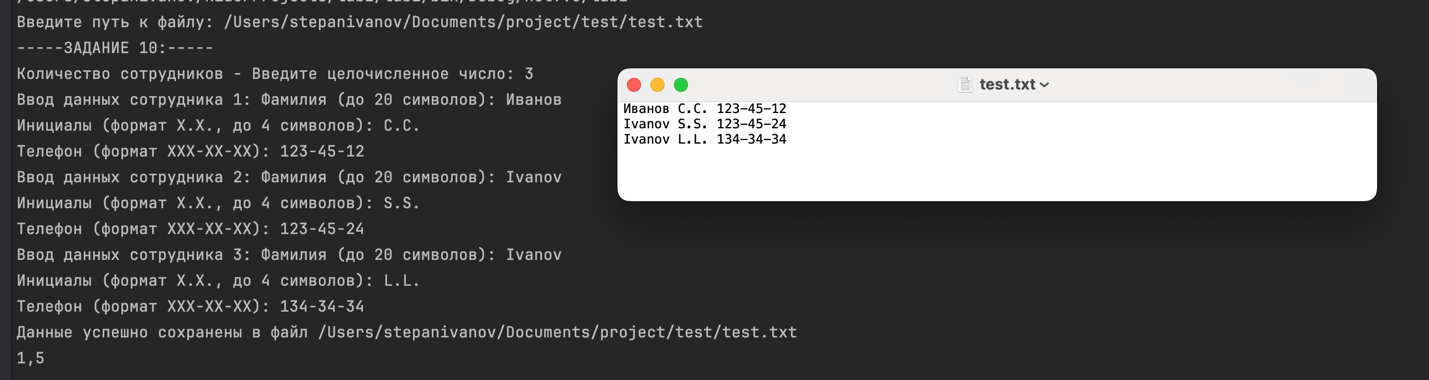
# Постановка задачи

На вход программе подаются сведения о телефонах всех сотрудников некоторого учреждения. В первой строке сообщается количество сотрудников N. Каждая из следующих N строк имеет строго определенный формат: сначала идет фамилия сотрудника (строка длиной не более 20 символов), затем через пробел инициалы (ровно 4 символа в формате "Б.Б." где Б - это буква), затем через пробел семизначный номер телефона в формате "XXX-XX-XX" (где 3-я и 4-я цифры разделены дефисом, и 5-я и 6-я цифры также разделены дефисом). Пример правильно оформленной строки: "Иванов П.С. 555-66-77". Важно учитывать, что сотрудники одного подразделения всегда имеют одинаковый номер телефона, при этом все номера телефонов в учреждении отличаются только двумя последними цифрами. Требуется разработать эффективную программу, которая проанализирует полученные данные и выведет на экран среднее количество сотрудников, приходящееся на одно подразделение этого учреждения. Для решения задачи необходимо сначала определить, какие сотрудники относятся к одному подразделению (по одинаковым номерам телефонов), затем подсчитать количество сотрудников в каждом подразделении, после чего вычислить среднее арифметическое этих значений. Результат должен быть выведен с точностью до двух знаков после запятой в формате "X.XX".

# Алгоритм решения

Создан класс FilesAndCollection, в котором инициализировано поле filePath типа string который будет хранить ссылку на текстовый файл, после в конструкторе описывается поле. Прописан метод CalculateAverageEmployeesPerDepartment, который вычисляет среднее кол-во сотрудников на подразделение – Сначала создается словарь departmentCounts для подсчета сотрудников по подразделениям и переменная totalEmpleyes для общего числа сотрудников. Дальше происходит чтение файла где каждая строка разбивается на части по пробелам, номер телефона разбивается по дефисам и формируется ключ из первых 5 цифр номера, если ключ уже есть в словаре, то увеличиваем счетчик, а если нет то добавляем это ключ в словарь и после вычисляется среднее значение. В Main() пользователь сам формирует текстовый документ с помощь чекера и после вызывается вышеперечисленная функция.

# Тестирование



# Код программы

https://github.com/6ermutka/yplab/tree/master/C%232