

Лабораторная 5. Ввод-вывод в java.

Цель:

- Разработка приложений, использующие потоки ввода вывода.

Упражнение 1. Простая работа с файлом.

1. Откройте исходный файл ListOfNumbers2.java.
2. Добавьте метод readList(). Этот метод должен считывать целочисленные (int) значения из файла.
3. Напечатайте каждое значение и добавьте их в конец массива. Следует обработать все возможные соответствующие данной операции ошибки.
4. Для тестирования программы потребуется создать текстовый файл, содержащий целые числа.

Упражнение 2. Знакомство с классом *BufferedReader*

В этом упражнении необходимо написать простое консольное приложение, которое считывает с системной консоли строку для поиска совпадений в текстовом файле, размещенном на диске. Имя файла, используемого для поиска, задается также с консоли.

Для решения поставленной задачи:

1. Откройте файл FileScanInteractive.

Класс FileScanInteractive уже имеет метод countTokens, созданный за вас. В качестве параметров этот метод принимает наименование файла и строку для поиска. Метод открывает файл с наименованием, которое вы ему передали, и использует экземпляр класса Scanner для поиска передаваемой строки. Для каждой передаваемой строки метод позволяет производить учет количества ее вхождений, каждый раз увеличивая счетчик (instanceCount). Когда файл будет прочитан до конца, метод вернет значение instanceCount. Метод пробрасывает любые возникающие IOException, поэтому убедитесь, что его вызов выполняется в блоке try-catch.

2. Код в методе main проверяет количество переданных аргументов. Программа ожидает как минимум один аргумент (строку с именем файла, чтобы открыть данный файл). Если количество аргументов меньше одного, то программа выходит с кодом ошибки (-1).

- а. Метод main принимает массив строк. Используйте свойство length для определения ситуации, когда массив содержит менее одного аргумента.

b. Выведите сообщение, если передано меньше одного аргумента и используйте `System.exit()` для выхода с кодом ошибки (-1 — типичный код для индикации ошибки).

3. Сохраните первый аргумент, переданный в приложение, как строку (`String`).

4. Создайте экземпляр класса `FileScanInteractive`. Вам понадобится этот объект для использования метода `countTokens`, так как метод `countTokens` не является статическим.

5. Откройте системную консоль для ввода с использованием `BufferedReader`.

a. Используйте `try-with resources` для открытия `BufferedReader`, работающего с системной консолью.

b. Убедитесь, что вы добавили `catch` к блоку `try`. Любое исключение будет с типом `IOException`.

c. В бесконечном цикле `while` считывайте с консоли строку, пока пользователь не введет строку “q”, если это произошло – прервите цикл.

Для сравнения строк можно использовать `String.equalsIgnoreCase()`, чтобы позволить пользователю ввести “q” как в верхнем, так и в нижнем регистрах. Кроме того, использование метода `String.trim()`, хорошее решение для удаления пробелов и других служебных символов из строки.

d. Если строка, считанная из консоли, не является строкой для выхода из программы (“q”), тогда вызовите метод `countTokens`, передав ему в качестве параметра имя файла и строку для поиска.

e. Выведите сообщение о том, сколько раз поисковая строка встречается в файле.

f. Добавьте необходимые инструкции `import`.

6. Сохраните класс `FileScanInteractive`.

7. Если у вас нет ошибок компиляции, вы можете протестировать программу, используя файл `DeclarationOfIndependence.txt`, который находится в корне проекта.

a. Нажмите правой кнопкой на проекте и выберите «Свойства»

b. Нажмите “Выполнение” (Run)

c. Введите имя файла по которому будет производиться поиск (например, `C:\...\ObamaInaguralAddress.txt`)

d. Нажмите ОК

e. Запустите приложение и попытайтесь найти любые слова, которые встречаются в файле, и которых там нет. Вывод должен выглядеть примерно так:

Searching through the file:

D:\labs\resources\ObamaInaguralAddress.txt

Enter the search string or q to exit: when

The word "when" appears 3 times in the file.

Enter the search string or q to exit: rights

The word "rights" appears 3 times in the file.

Enter the search string or q to exit: free

The word "free" appears 4 times in the file.

Enter the search string or q to exit: q

BUILD SUCCESSFUL (total time: 16 seconds)