**Тестовое задание #3.**

Вариант: Проверка знаний языка JAVA.

Цель: Слияние данных источников и вывод итога обработки.

Написать программу сортировки слиянием нескольких файлов.

Входные файлы содержат данные одного из двух видов: целые числа или строки. Данные записаны в столбик (каждая строка файла – новый элемент). Строки могут содержать любые не пробельные символы. Считается, что файлы предварительно отсортированы.

Результатом работы программы должен являться новый файл с объединенным содержимым входных файлов, отсортированным по возрастанию или убыванию путем сортировки слиянием.

Если содержимое исходных файлов не позволяет произвести сортировку слиянием (например, нарушен порядок сортировки), производится частичная сортировка (насколько возможно для этого алгоритма, как именно обрабатывать поврежденный файл – на усмотрение разработчика).

Выходной файл должен содержать отсортированные данные даже в случае ошибок, однако возможна потеря ошибочных данных.

Необходимо самостоятельно реализовать алгоритм сортировки методом слияния и использовать его для сортировки содержимого файлов. Не использовать библиотечные функции сортировки.

Все возможные виды ошибок должны быть обработаны. Программа не должна «падать». Если после ошибки продолжить выполнение невозможно, программа должна сообщить об этом пользователю с указанием причины неудачи. Частичная обработка при наличии ошибок более предпочтительна чем останов программы. Код программы должен быть «чистым».

Параметры программы задаются при запуске через аргументы командной строки, по порядку:

1. режим сортировки (-a или -d), необязательный, по умолчанию сортируем по возрастанию;

2. тип данных (-s или -i), обязательный;

3. имя выходного файла, обязательное;

4. остальные параметры – имена входных файлов, не менее одного.

Примеры запуска из командной строки для Windows:

sort-it.exe -i -a out.txt in.txt (для целых чисел по возрастанию)

sort-it.exe -s out.txt in1.txt in2.txt in3.txt (для строк по возрастанию)

sort-it.exe -d -s out.txt in1.txt in2.txt (для строк по убыванию)

К решению должна прилагаться инструкция по запуску. В ней также можно отображать

особенности реализации, не уточненные в задании.

**Пример:**

|  |  |  |  |
| --- | --- | --- | --- |
| in1.txt | in2.txt | In3.txt | out.txt |
| 1 | 1 | 1 | 1 |
| 4 | 8 | 2 | 1 |
| 9 | 27 | 3 | 1 |
|  |  |  | 2 |
|  |  |  | 3 |
|  |  |  | 4 |
|  |  |  | 8 |
|  |  |  | 9 |
|  |  |  | 27 |