

master ▾

...

[CompareMerge](#) / README.md

73 lines (59 sloc) | 5.94 KB

...

↪ MergeSort

Сортировка слиянием файлов со строковыми или целочисленными массивами

↪ Вариант запуска №1 (с установкой Java)

1. Должна быть установлена [Java SE Development Kit 19.0.2](#) на компьютере.
2. Прописана переменная окружения. Для Windows это %JAVA_HOME%
3. Перезагрузиться
4. Скачать на компьютер папку [RunMe](#)
5. Перейти в командной строке в скачанную папку.

Версия java

```
F:\MergeSort>java -version
java version "19.0.2" 2023-01-17
Java(TM) SE Runtime Environment (build 19.0.2+7-44)
Java HotSpot(TM) 64-Bit Server VM (build 19.0.2+7-44, mixed mode, sharing)
```

Переменная

окружения

System variables

Variable	Value
JAVA_HOME	%ProgramFiles%\Java\jdk-19\

↪ Вариант запуска №2 (Без установки Java)

GitHub не даёт заливать файлы больше 100мб в проект. Проект со вложенным окружением JRE, где всё включено и ничего не нужно устанавливать, выложил [здесь](#)

🔗 Аргументы

Аргумент	Описание
java -jar MergeSort.jar	Для запуска jar файла
или	
MergeSort.exe	Или для запуска exe
-----	-----
-a	Не обязательный (по умолчанию сортировка Ascending)
или	
-d	Не обязательный (для выбора сортировки Descending)
-----	-----
	Обязательно выбрать тип данных в файлах: либо Строки (-s), либо целые числа (-i)
-s str1.txt,str2.txt,int3.txt	Указание и выбор файлов со строками. Файлы перечисляются СРАЗУ после аргумента (-s) через ЗАПЯТУЮ
или	
-i int1.txt,int2.txt,int3.txt	или Указание и выбор файлов сцелыми числами. Файлы перечисляются СРАЗУ после аргумента (-i) через ЗАПЯТУЮ
-----	-----
-o result.txt	Обязательный. Указание файла для итоговой выгрузки

🔗 Примеры использования

Для входных файлов с целыми числами. Сортировка по возрастанию.

jar в случае ошибок с входными данными - покажет ошибку:

```
F:\MergeSort>java -jar MergeSort.jar -a -i int1.txt,int2.txt,int3.txt -o result.txt

F:\MergeSort\int1.txt
Взятая строка HE integer

F:\MergeSort\int2.txt
Взятая строка HE integer
Взятая строка HE integer
Взятая строка HE integer

F:\MergeSort\int3.txt
Взятая строка HE integer
Взятая строка HE integer

Output result.txt
```

Для входных файлов с целыми числами. Сортировка по убыванию.

jar в случае ошибок с входными данными - покажет ошибку:

```
F:\MergeSort>java -jar MergeSort.jar -d -i int1.txt,int2.txt,int3.txt -o result.txt

F:\MergeSort\int1.txt
Взятая строка НЕ integer

F:\MergeSort\int2.txt
Взятая строка НЕ integer
Взятая строка НЕ integer
Взятая строка НЕ integer

F:\MergeSort\int3.txt
Взятая строка НЕ integer
Взятая строка НЕ integer

Output result.txt
```

Для входных файлов с целыми числами. Сортировка по убыванию.

exe в случае ошибок с входными данными ошибок не выведет:

```
F:\MergeSort>MergeSort.exe -i int1.txt,int2.txt,int3.txt -o result.txt -d

F:\MergeSort>
```

Для входных файлов со строками. Сортировка по возрастанию.

jar в случае ошибок с входными данными - покажет ошибку:

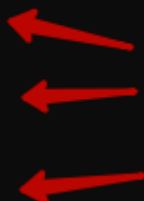
```
F:\MergeSort>java -jar MergeSort.jar -a -s str1.txt,str2.txt,str3.txt -o result.txt

F:\MergeSort\str1.txt
Строка содержит пробел

F:\MergeSort\str2.txt
Строка содержит пробел

F:\MergeSort\str3.txt
Строка содержит пробел

Output result.txt
```



**Если в файле строка с пробелом,
покажет ошибку взятия этой строки**

Для входных файлов со строками. Сортировка по убыванию.

jar в случае ошибок с входными данными - покажет ошибку:

```
F:\MergeSort>java -jar MergeSort.jar -d -s str1.txt,str2.txt,str3.txt -o result.txt

F:\MergeSort\str1.txt
Строка содержит пробел

F:\MergeSort\str2.txt
Строка содержит пробел

F:\MergeSort\str3.txt
Строка содержит пробел

Output result.txt
```

Для входных файлов со строками. Сортировка по убыванию.

exe в случае ошибок с входными данными ошибок не выведет:

```
F:\MergeSort>MergeSort.exe -s str1.txt,str2.txt,str3.txt -o result.txt -d
F:\MergeSort>
```

🔗 Реализация

[Java 19.0.2](#)

[Maven 4.0.0](#)

- [commons-io](#)
- [commons-cli](#)
- [maven-assembly-plugin](#)
- [launch4j-maven-plugin](#)