

## Описание:

В СигмаБанк произошла неудачная миграция с одной системы учёта сотрудников в другую. В результате сейчас на сервере хранится множество файлов с расширением .sb. Данные в этих файлах структурированы, но неупорядочены. Ваша задача создать консольное java приложение, которое упорядочит эти данные. На выходе должен получиться список файлов с именем `<department_name>.sb`, `department_name` – имя департамента. Сам файл должен содержать данные сотрудников. Информация в каждом файле должна быть отсортирована в соответствии с требованиями и входными параметрами. Так же, в зависимости от параметров, должна быть выведена статистика либо в консоль, либо в отдельный файл. Все данные, которые не получится прочесть корректно, распознать либо соотнести с какой-либо группой, должны быть записаны в отдельный файл `error.log`. Входные файлы располагаются в той же директории, где запускается программа. Нужно искать по расширению .sb.

## Требования:

### 1. Входной файл:

- Входной файл содержит строки с информацией о менеджерах (Manager) и сотрудниках (Employee).
- Пример содержимого файла:

```
Employee,102,Emily Johnson,abc,1
Manager,2,Michael Brown,6000,Sales
Employee,103,Chris White,-2900,2
Employee,104,Anna Taylor,3100,2
Manager,1,Jane Smith,5000,HR
Employee,101,John Doe,3000,1
Employee,105,Robert Black,,2
```

### Формат данных:

**Должность , идентификатор , имя , зарплата , департамент / идентификатор менеджера**

Идентификаторы сотрудников уникальны для всех департаментов, т.е. не может быть менеджеров либо сотрудников с одинаковым идентификатором.

Данные могут быть в разном порядке: менеджеры после сотрудников, сотрудники не после своих менеджеров, сотрудники и менеджеры в разных файлах. Определение департамента производится по **имени**, указанному в строке **менеджера**. Признаком того, что сотрудник относится к менеджеру, является последнее значение в строке - идентификатор менеджера.

Например:

```
Manager,1,Jane Smith,5000,HR  
Employee,101,John Doe,3000,1
```

Важно: не гарантировано, что менеджеры и сотрудники в одном файле принадлежат одному департаменту.

Один и тот же человек (по имени) может быть менеджером в разных департаментах с разными ID, но его идентификатор будет один.

Данные могут содержать пробелы либо в начале, либо в конце. В итоговом варианте пробелов быть не должно.

Корректное значение зарплаты – это вещественное число больше нуля. Разделителем между целой и дробной частью является точка.

Выходной файл для департамента должен содержать как минимум менеджера.

У департамента не может быть более одного менеджера, но сотрудников может и не быть.

## 2. Сортировка в файлах департамента:

- Сортировка сотрудников может быть либо по имени, либо по зарплате в прямом либо в обратном порядке.
- Параметр сортировки передается через аргументы командной строки --sort или -s (--sort=name или -s=salary). Параметр сортировки является необязательным. Если он не указан, то данные выводятся так, как указаны в исходных файлах в порядке чтения.
- Порядок сортировки задаётся через флаг --order и имеет значения: asc – прямой, desc – обратный (--order=asc или --order=desc). Если сортировка не задана, то порядок не может быть указан, в таком случае программа должна показывать сообщение об ошибке в параметрах. Если задан

неправильный тип сортировки либо порядка, так же должно быть сообщение об ошибке.

- Сортировка не применяется к менеджерам, т.е. менеджер в выходном файле всегда идёт первым.

### 3. Вывод статистики данных:

- Статистика выводится по параметру `--stat`. Если он не задан, остальные флаги, которые касаются статистики должны быть распознаны как неизвестные параметры.
- Результат должен выводиться либо в консоль, либо в файл. Приложение должно поддерживать два варианта. Если задан флаг `--output=console` или `-o=console` либо не задан, то вывод происходит в консоль. Если задан флаг `--output=file` или `-o=file`, то должен идти следом флаг `--path=<путь к выходному файлу>`. Задание пути без указания `--output=file/-o=file` считается неправильным и должно быть сообщение об ошибке.

Так же как `--output=file` без задания пути к выходному файлу.

- Зарплата менеджеров **не учитывается** в статистике. Если в департаменте нет сотрудников с корректной зарплатой, статистика будет: 0.00.
- В статистике должны быть учтены только корректные данные.
- Статистика содержит имя департамента, минимальная, средняя, максимальная зарплаты.
- Первой строкой вывода статистики должны быть столбцы:

department, min, max, mid

- Числовой тип должен выводиться с округлением до 2 знаков после точки в большую сторону.
- Департаменты должны быть отсортированы по имени в лексикографическом порядке.

#### Формат вывода:

Допустим у нас есть файлы

*first.sb, second.sb*

Содержимое файла first.sb

```
first.sb
Employee,102,Emily Johnson,abc,1
Manager,2,Michael Brown,6000,Sales
Manager,1,Jane Smith,5000,HR
```

## Содержимое файла second.sb

```
second.sb
Manager,3,Jane Smith,5000,Dev
Employee,103,Chris White,-2900,2
Employee,104,Anna Taylor,3100,2
Employee,101,John Doe,3000,1
Employee,105,Robert Black,,2
```

## Выходные файлы

### **HR.sb**

```
Manager,1,Jane Smith,5000
Employee,101,John Doe,3000,1
```

### **Sales.sb**

```
Manager,2,Michael Brown,6000
Employee,104,Anna Taylor,3100,2
```

### **Dev.sb**

```
Manager,3,Jane Smith,5000
```

### **error.log**

```
Employee,102,Emily Johnson,abc,1
Employee,103,Chris White,-2900,2
Employee,105,Robert Black,,2
```

## Статистика

department, min, max, mid

```
HR, 3000.00, 3000.00, 3000.00
Sales,3100.00, 3100.00, 3100.00
Dev,0.00,0.00,0.00
```

## Параметры командной строки

Параметр	Альт. вариант	Обязательный	Допустимые значения	Описание
--sort	-s	Нет	name, salary	Тип сортировки сотрудников: по имени или зарплате. Без сортировки данные выводятся как есть.
--order	—	Нет	asc, desc	Порядок сортировки. <b>Только с --sort.</b>
--stat	—	Нет	—	Включает генерацию статистики по департаментам.
--output	-o	Нет (обязателен с --path)	console, file	Способ вывода статистики. По умолчанию — console.
--path=<путь>	—	Да, если --output=file	путь к выходному файлу	Путь к файлу со статистикой. Без --output=file — ошибка.
другие параметры	—	—	—	Любые неизвестные параметры считаются ошибочными.

## Важно!

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 остановка программы.

Код программы должен быть «чистым»: не должно быть закомментированного или неиспользуемого кода, также к коду должен быть применён единообразный code style (например тот, что используется в IDEA). **Самое главное: код должен компилироваться, собираться и соответствовать предоставленному формату вывода. Никаких собственных форматов вывода.**

Для реализации необходимо использовать язык программирования Java, допустимо использовать стандартные системы сборки проекта (Maven, Gradle).

Решение принимается в виде исходного кода проекта. К решению должна прилагаться инструкция по запуску. В ней можно отображать особенности реализации, не уточненные в задании. В частности, в инструкции необходимо указывать:

- версию Java (желательно не выше 17ой);
- при использовании системы сборки – указать систему сборки и ее версию;
- при использовании сторонних библиотек указать их название и версию, а также приложить ссылки на такие библиотеки (можно в формате зависимостей системы сборки).

Примеры корректного запуска

```
java -jar app.jar --sort=name --order=asc --stat
```

```
java -jar app.jar -s=salary --order=desc --stat -o=file -  
-path=output/statistics.txt
```

```
java -jar app.jar --stat
```

**Совет:** перед отправкой задания, попробуй запустить программу в консоли командой, приведённой выше.