Описание:

Создайте программу на Java, которая читает данные из текстового файла. Программа должна обрабатывать возможные ошибки в данных, сортировать сотрудников и выводить информацию либо в консоль, либо в файл, а также выводить статистику по департаментам. Некорректные данные должны исключаться из итоговой таблицы и выводиться в конце с заголовком «Некорректные данные».

Требования:

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

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

Формат данных:
`Должность,идентификатор,имя,зарплата,департамент/идентификатор менеджера`

Идентификаторы 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. Признаком того, что сотрудник относится к менеджеру, является последнее значение идентификатор менеджера. Например:

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

Данные могу 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.

2. **Сортировка**:

- 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 данных.
- Порядок сортировки задаётся через флаг --order либо -o и имеет значения: asc – прямой, desc – обратный (--order=ask или -o=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.

3. **Вывод данных**:

- Результат должен выводиться либо в консоль, либо в файл. Приложение должно поддерживать два варианта. Если задан флаг --output=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 считается неправильным и должно быть сообщение об ошибке. Так же как --output=file без задания пути к выходному файлу.
- В начале всегда идёт имя департамента, потом менеджер, потом сотрудники.
- В конце каждого департамента должна выводиться статистика: количество сотрудников и средняя зарплата в формате: `amount,salary`

  Например:
  `2,3200.5`

- У сотрудника может не быть менеджера, что также является некорректным случаем и должно выводиться в конце с некорректными данными.
- 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:

```
HR
Manager,1, Jane Smith, 5500.0
Employee,101, John Doe, 3300.0
```

```
2, 4400.0
Sales
Manager,2, Michael Brown, 6600.0
Employee,104, Anna Taylor, 3410.0
Employee,105, Robert Black, 0.0
3, 3333.33
Некорректные данные:
Employee,102,Emily Johnson,abc,1
Employee,103,Chris White,-2900,2
```

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 и собираться.**

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).

Пример запуска программы из консоли:

```
java -jar <путь к jar файлу> <набор параметров>
```

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