

Реализация алгоритма генерации пар.

Описание:

Требуется разработать алгоритм для генерации пар участников парного программирования и запрограммировать его в виде консольного приложения Java. Приложение в качестве входных и выходных файлов должно использовать текстовые файлы. Входной и выходной файлы должны задаваться в качестве аргументов при вызове консольной программы (первый аргумент путь до входного файла, второй аргумент путь до выходного файла).

Входной файл содержит данные о номере участника (номер по порядку), имени участников и номерах групп на каждую неделю. Входной файл может не содержать информации о номерах групп, если генерация происходит впервые. Формат входного файла:

Номер по порядку;ФИО программиста 1;номер группы первой недели;номер группы второй недели

Номер по порядку;ФИО программиста 2;номер группы первой недели;номер группы второй недели

Номер по порядку;ФИО программиста 3;номер группы первой недели;номер группы второй недели

Каждая строка файла должна иметь одинаковое количество полей которые разделяются знаком “;”, после последнего поля разделительный знак не ставится. Остальные поля содержат номера групп для каждой недели: 3 поле для первой недели, 4 поле для второй недели и так далее.

Выходной файл содержит те же данные что и входной файл, только прибавляются по одному полю в каждую строку - генерируемые группы.

Условия:

1. Каждая группа должна содержать не более двух участников
2. Алгоритм должен предусматривать уникальную генерацию пар опираясь на данные предыдущих генераций, на сколько это возможно. Если так получилось и при новой генерации появляется пара которая была раньше, то к названию номера в конце добавляется * (звездочка).
3. Номер для конкретной пары на конкретной недели является уникальным. Иначе говоря номера пар не должны повторяться.
4. Номера пар формируются по порядку в 33-ричной системы счисления и используют в себе символы от 0 до 9 и от A до Z. Например: 1, 9Z, 5FA и так далее.
5. Группы в которых содержится только один участник должны помечаться символом ^ в начале названия группы.

6. Если участник не участвовал в парном программировании на каких либо неделях (пропускал или же добавился недавно и то есть не участвовал во всех предыдущих), то вместо наименования групп стоит ставить символ - (прочерк).
7. Если количество полей в строках не совпадает стоит записывать сообщения об ошибках с указанием номеров строк (за эталон брать первую строку). Сообщения должны записываться в текстовый файл **error.txt** (его стоит создавать если только ошибки появились, в этом случае выходной файл не создается).
8. Если перед именем участника стоит символ **#** (решетка), то он не должен участвовать в расчетах, этот пользователь считается удаленным.
9. Номер пользователя представляет из себя число в 10-ричной системе счисления.

Пример:

входной файл **programmers.txt**

```
1;Иванов Сергей;1
2;Николаев андрей;1
3;Сидорова Елена;2
4;Березко Николай;2
5;Петрова Ольга;3
```

файл **pait_programmers.txt**

```
1;Иванов Сергей;1;1
2;Николаев андрей;1;2
3;Сидорова Елена;2;1
4;Березко Николай;2;3
5;Петрова Ольга;3;2
```

Дополнительное задание:

1. Добавить возможность добавления новых участников в конец списка. Новым участникам должны присваиваться номера, проставляться все предыдущие недели как пропущенные. Добавление участников должно происходить перед генерацией групп. Добавление участника должно осуществляться с помощью аргументов при вызове консольной программы. Среди аргументов ищется ключ **-add** после которого должны идти имена пользователей в кавычках. Если ключ есть а имен участников поле него нет, то это считается ошибкой и сообщение об ошибке должно быть записано в файл **error.txt**. Пример вызова:

```
java -jar программа c:/programmers.txt c:/pair_programmers.txt -add "Пушкин Александр"
"Стивен Кинг"
```

2. Добавить возможность удаления участников парного программирования. К имени участника который подлежит удалению в начале ставится знак **#** (решетка). Удаление участников задается через аргументы передаваемые

консольной программе. Среди аргументов ищется ключ `-del` и после него должны перечисляться номера участников которые подлежат удалению. Если номеров нет а ключ есть, то это считается ошибкой и сообщение об ошибке должно быть записано в файл **error.txt**. Удаление пользователей должно происходить перед генерацией пар. Пример вызова:

```
java -jar programma c:/programmers.txt c:/pair_programmers.txt -del 2 10
```