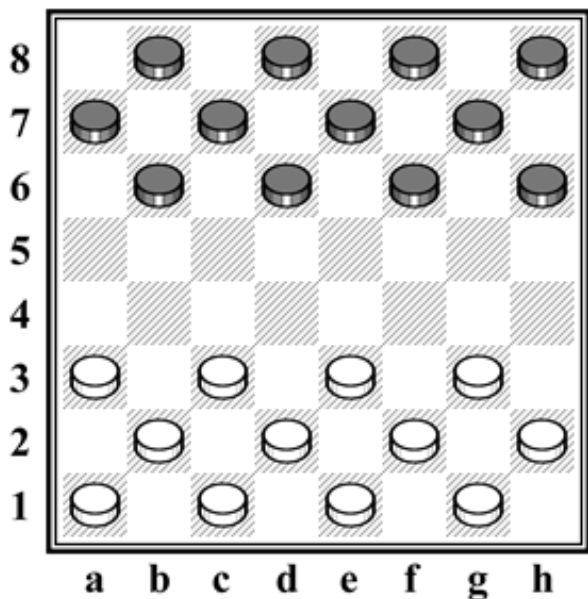


Задача. Русские шашки.

!! Должно работать на Java 11 версии.



боя несколько, можно взять любой.

➤ **error** - другие ошибки.

Все исходные позиции - валидные.

Требования к коду

- При написании кода для ошибок нужно создавать свои классы исключений. Исключения отлавливаются на самом высшем уровне (метод `main()`) и выводится соответствующий error message. Ловля checked-исключений “по месту” (`e.printStackTrace()`) запрещена.
- Нужно реализовать JavaDoc для нетривиальных публичных методов и классов.
- Также нужно сделать хотя бы 5 unit-тестов.
- (Можно создать Checkstyle Rule) запрещено использовать устаревшие API в Java, заменённые в новых версиях. Прежде всего это касается `File` и `FileInputStream` (эти классы чаще всего продолжают использовать новички, хотя есть `Java NIO API`), также: `Date`, `Calendar`, `DateFormat`, `StringBuffer`, `Vector`, `HashTable`.
- Парсинг входных данных осуществлять с помощью регулярных выражений, конечных автоматов или чего-то подобного. “Наивный” парсинг с помощью `split`, `indexOf`, `replace` не приветствуется.

Формат входных данных

Координаты шашек на доске и ходы записываются в [стандартной шашечной нотации](#). Характеристики ходов (“!” - хороший ход, “?” - плохой ход и т.д.) в записи не применяются. Если шашка является дамкой, то её координата обозначается с большой буквы (например, не d4, а D4). Если за 1 ход происходит несколько взятий, то они записываются через “:” так: e5:c3:a1.

1. Строка с координатами белых шашек
2. Строка с координатами чёрных шашек
3. Список ходов. Пара ходов (белые + черные) в строке.

Формат выходных данных

1. Строка с координатами белых шашек
2. Строка с координатами чёрных шашек

Либо сообщение об ошибке

Дана композиция начала или середины партии [русских шашек](#). Далее даётся список из 1 или нескольких ходов. На выходе нужно вывести итоговую композицию или сообщение об ошибке если какой-то из ходов списка невозможен.

В данной реализации шашки могут не только бить, но и *ходить назад*.

Типы сообщений об ошибке:

- **busy cell** - целевая клетка занята
- **white cell** - целевая клетка белая (шашки расставляются только на чёрные и, в силу правил, оказаться на белых не могут),
- **invalid move** - в шашках бить обязательно. Причём, бить надо всю цепочку до конца. Ошибка выводится в том случае если у игрока есть вариант побить шашку, но он его не использует, а идёт на другую клетку. Если вариантов

Примеры входных данных

in	out
a1 a3 b2 c1 c3 d2 e1 e3 f2 g1 g3 h2 a7 b6 b8 c7 d6 d8 e7 f6 f8 g7 h6 h8 g3-f4 f6-e5 c3-d4 e5:c3 b2:d4 d6-c5 d2-c3 g7-f6 h2-g3 h8-g7 c1-b2 f6-g5 g3-h4 g7-f6 f4-e5 f8-g7	a1 a3 b2 c3 d4 e1 e3 e5 f2 g1 h4 a7 b6 b8 c5 c7 d8 e7 f6 g5 g7 h6