

Elementary Tasks

Общие требования

1. При передаче некорректных параметров на исполнение приложение не должно завершать работу сбоем.
2. Запуск без параметров выводит инструкции по использованию программы.
3. Параметры передаются в порядке, приведённом в описании задания.
4. Исходный код должен быть написан в ООП стиле. Части программы должны легко поддаваться изменениям и расширению. Код должен быть переносимым.
5. Весь код должен быть написан с использованием только Java 8 SE.

Задания

1. Шахматная доска

Вывести шахматную доску с заданными размерами высоты и ширины, по принципу:

```
* * * * *
* * * * *
* * * * *
* * * * *
```

Программа запускается через вызов главного класса с параметрами.

2. Анализ конвертов

Есть два конверта со сторонами (a,b) и (c,d) определить, можно ли один конверт вложить в другой. Программа должна обрабатывать ввод чисел с плавающей точкой. Программа спрашивает у пользователя размеры конвертов по одному параметру за раз. После каждого подсчёта программа спрашивает у пользователя хочет ли он продолжить. Если пользователь ответит “y” или “yes” (без учёта регистра), программа продолжает работу сначала, в противном случае – завершает выполнение.

3. Сортировка треугольников

Разработать консольную программу, выполняющую вывод треугольников в порядке убывания их площади. После добавления каждого нового треугольника программа спрашивает, хочет ли пользователь добавить ещё один. Если пользователь ответит “y” или “yes” (без учёта регистра), программа попросит ввести данные для ещё одного треугольника, в противном случае – выводит результат в консоль.

- Расчёт площади треугольника должен производиться по формуле Герона.
- Каждый треугольник определяется именем и длинами его сторон.

Формат ввода (разделитель - запятая):

<имя>, <длина стороны>, <длина стороны>, <длина стороны>

- Приложение должно обрабатывать ввод чисел с плавающей точкой.
- Ввод должен быть нечувствителен к регистру, пробелам, табу.
- Вывод данных должен быть следующим примере:

```
===== Triangles list: =====
1. [Triangle first]: 17.23 cm
2. [Triangle 22]: 13 cm
3. [Triangle 1]: 1.5 cm
```

4. Файловый парсер

Нужно написать программу, которая будет иметь два режима:

1. Считать количество вхождений строки в текстовом файле.
2. Делать замену строки на другую в указанном файле

Программа должна принимать аргументы на вход при запуске:

1. <путь к файлу> <строка для подсчёта>
2. <путь к файлу> <строка для поиска> <строка для замены>

5. Число в пропись

Нужно преобразовать целое число в прописной вариант: 12 – двенадцать. Программа запускается через вызов главного класса с параметрами.

6. Счастливые билеты

Счастливые билеты.

Есть 2 способа подсчёта счастливых билетов:

1. Московский — если сумма цифр автобусного билета напечатано шестизначное число, и сумма первых трёх цифр равна сумме последних трёх, то этот билет считается счастливым.
2. Ленинградский, или Питерский — если сумма чётных цифр билета равна сумме нечётных цифр билета, то билет считается счастливым.

Задача:

Номер билета - шестизначное число. Нужно написать консольное приложение, которое может посчитать количество счастливых билетов. Для выбора алгоритма подсчёта читается текстовый файл. Путь к текстовому файлу задаётся в консоли после запуска программы. Индикаторы алгоритмов:

- 1 - слово 'Moscow'
- 2 - слово 'Piter'

После задания всех необходимых параметров, программа в консоль должна вывести количество счастливых билетов для указанного способа подсчёта.

7. Числовая последовательность

Программа выводит ряд натуральных чисел через запятую, квадрат которых меньше заданного n . Программа запускается через вызов главного класса с параметрами.

8. Ряд Фибоначчи для диапазона

Программа позволяет вывести все числа Фибоначчи, которые находятся в указанном диапазоне, либо имеют указанную длину. Параметры задаются аргументами при вызове главного класса. Числа выводятся через запятую по возрастанию.

9. Вознесение в степень

Разработать метод быстрого возведения в степень с использованием алгоритма вида:

`pow(5, 5)`

- 1.
2. $5 * 5 = 25$
3. $25 * 25 = 625$
4. $625 * 5 = 3125$

10. Хранилище

Создать библиотеку, которая позволяет сохранять, изменять и удалять объекты в хранилище.

Объекты имеют уникальный идентификатор. **Объекты могут иметь связи.**

Сделать реализации хранилища в виде:

Допустимо использование:

- * Память. (RAM);
- * Бинарного файла;
- * XML файла;
- * JSON файла;
- * DB (MySQL).

mysql:mysql-connector-java, com.fasterxml.jackson.core.jackson-databind

Используйте паттерн проектирования DAO.

