Gui приложение с использованием библиотеки javax.swing

5) Тыколка, две команды

6) Калькулятор, крестики нолики

7) 1. String text = “Вася заработал 5000 рублей, Петя - 7563 рубля, а Маша - 30000 рублей”

Напишите код, который считает сумму заработка Васи и Маши.

Использование регулярных выражений в данном задании не допускается.

2. Напишите программу, которая на входе через консоль принимает фамилию, имя и отчество одной строкой (например, «Иванов Сергей Петрович») и выводит фамилию, имя и отчество отдельно в формате:

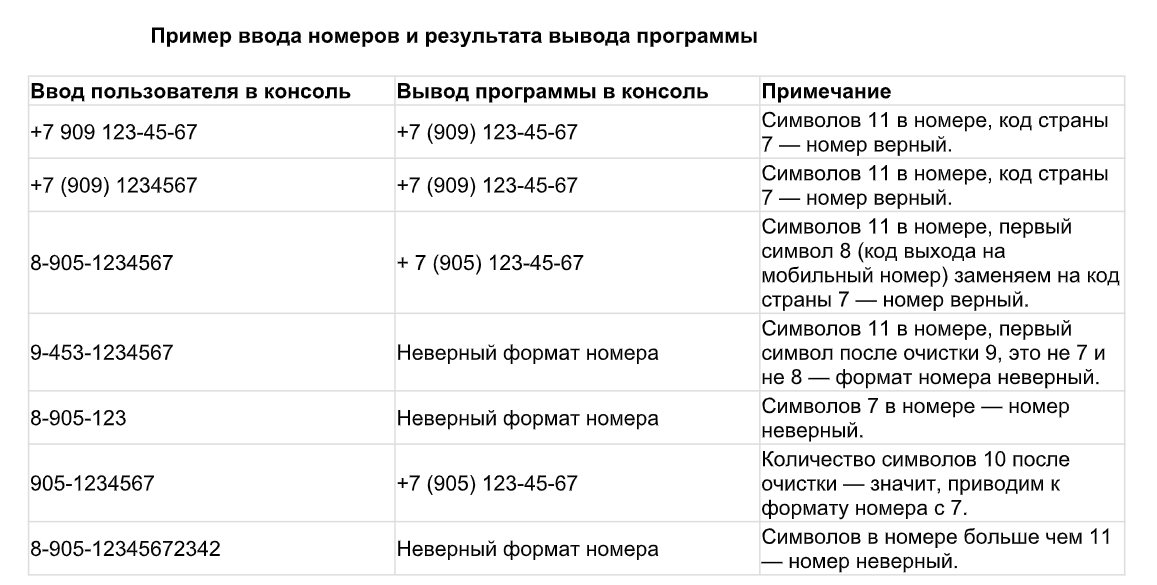
Фамилия: Иванов

Имя: Сергей

Отчество: Петрович

Валидная строка от пользователя, которую мы можем интерпретировать как Ф. И. О., должна содержать три слова, состоящих из символов кириллицы, разделённых пробелом, может содержать дефис. Если у человека отсутствует отчество, то выводить ФИ. Если строка не соответствует формату, то вывести в консоль: Введенная строка не является ФИО

3. Реализуйте удаление лишних символов при вводе номера телефона в консоли и проверку соответствия номера формату мобильных номеров России. Если введённую строку нельзя привести к формату мобильного номера — выводите сообщение о неверном вводе. Телефон может быть введён не только в формате 79091234567, но и с лишними символами.



8)

1. Разработайте список дел с помощью ArrayList, который управляется командами в

консоли. Команды: LIST, ADD, EDIT, DELETE.

Принцип работы команд:

 LIST — выводит дела с их порядковыми номерами;

 ADD — добавляет дело в конец списка или дело на определённое место, сдвигая

остальные дела вперёд, если указать номер;

 EDIT — заменяет дело с указанным номером;

 DELETE — удаляет.

Примеры команд

 LIST

 ADD Какое-то дело

 ADD 4 Какое-то дело на четвёртом месте

 EDIT 3 Новое название дела

 DELETE 7

Команды вводятся одной строкой пользователем в консоль.

2. Напишите программу используя TreeSet, в которой будет храниться перечень адресов

электронной почты. Адреса можно добавлять через консоль командой ADD и печатать

весь список командой LIST.

Программа должна проверять корректность вводимых email’ов и печатать сообщение об

ошибке при необходимости.

Принцип работы команд

 LIST — выводит список электронных адресов.

 ADD — проверяет и, если формат адреса верный, добавляет в множество.

Примеры команд

 LIST

 ADD hello@skillbox.ru

Команды вводятся одной строкой пользователем в консоль.

3. Напишите программу используя коллекцию Map, которая будет работать как

телефонная книга:

 Если пишем новое имя, программа просит ввести номер телефона и запоминает

его. Если новый номер телефона — просит ввести имя и также запоминает.

 Если вводим существующее имя или номер телефона, программа выводит всю

информацию о контакте.

 При вводе команды LIST программа печатает в консоль список всех абонентов в

алфавитном порядке с номерами.

Определяйте имя и телефон с помощью регулярных выражений.

Подумайте, что выбрать в качестве ключа и значения для Map, выберите лучший вариант

по вашему мнению. Опишите, какие минусы и плюсы видите в вашем выборе.

9)

1. Напишите генератор «красивых» автомобильных номеров

Используйте правила генерации номеров для получения более 2 млн номеров:

X, Y, Z — различные буквы автомобильного номера (&quot;А&quot;, &quot;В&quot;, &quot;Е&quot;, &quot;К&quot;, &quot;М&quot;, &quot;Н&quot;, &quot;О&quot;, &quot;Р&quot;,

&quot;С&quot;, &quot;Т&quot;, &quot;У&quot;, &quot;Х&quot;), N — цифры, R — регион (от 01 до 199);

XNNNYZR — пример, A111BC197, Y777HC66

2. Используя генератор «красивых» методов проведите поиск номера, введенного из

консоли, с помощью методов:

 прямым перебором по ArrayList, (array.contains())

 бинарным поиском по сортированному ArrayList, (Collections.binarySearch())

 поиском в HashSet, (setHash.contains())

 поиском в TreeSet. (setTree.contains())

2. Измерьте и сравните длительность каждого метода поиска.

Формат вывода результатов поиска:

Поиск перебором: номер найден/не найден, поиск занял 34нс

Бинарный поиск: номер найден/не найден, поиск занял 34нс

Поиск в HashSet: номер найден/не найден, поиск занял 34нс

Поиск в TreeSet: номер найден/не найден, поиск занял 34нс

Рекомендации

 Сортировка не входит в учёт времени для бинарного поиска.

 Для детального сравнения методов поиска используйте время в наносекундах:

     System.nanoTime()

 Перед использованием Collections.binarySearch() не забудьте использовать

Collections.sorted() для сортировки основного ArrayList