

Задание 1

Цели домашнего задания

Научиться работать с методами `indexOf()`, `lastIndexOf()`, `substring()` и `trim()` класса `String`.

Что необходимо сделать

- Выполните задание в классе `main/java/practice/strings/TotalEarningsCalculator.java`
- Напишите код, который считает сумму заработка всех друзей. Используйте методы `indexOf()`, `lastIndexOf()`, `substring()` и `trim()`. Использование регулярных выражений в данном задании не допускается.

Задание 2

Цели домашнего задания

Научиться работать с методами `indexOf()`, `lastIndexOf()`, `substring()` и `trim()` класса `String`.

Что необходимо сделать

- Выполните задание в классе `main/java/practice/strings/FullNameFormatter.java`
- 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 ФИО, должна содержать три слова, состоящих из кириллических символов, разделённых пробелом, может содержать дефис. Если строка не

соответствует формату, то в консоль выводится: Введенная строка не является ФИО.

- Использование регулярных выражений в данном задании не допускается.
- Строго соблюдайте формат вывода результата.

Работа с регулярными выражениями

Задание 3

Цели домашнего задания

Научиться писать и использовать несложные регулярные выражения.

Что необходимо сделать

- Выполните задание в классе `main/java/practice/TotalEarningsCalculatorRegex.java`
- Реализуйте получение суммы заработка каждого человека из текста регулярным выражением, чтобы в конце программы рассчитывалась и распечатывалась общая сумма заработка людей. В метод возможна передача любого сочетания имён и суммы зарплат в строке.

Задание 4

Цели домашнего задания

Научиться писать и использовать несложные регулярные выражения.

Что необходимо сделать

- Выполните задание в классе `main/java/practice/strings/SplitText.java`
- Реализуйте метод, который принимает в качестве аргумента строку, содержащую английский текст (не менее 100 слов). Метод должен

разделить текст на слова и вернуть строку, состоящую из полученных слов, каждое слово на отдельной строке.

- Знаки препинания и числа не являются словами.

Задание 5

Цели домашнего задания

Научиться писать и использовать несложные регулярные выражения.

Что необходимо сделать

- Выполните задание в классе
`Main main/java/practice/regex/FullNameFormatterRegex.java`
- Напишите программу (аналог задания 2), 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 ФИО, должна содержать три слова, состоящих из кириллических символов, разделённых пробелом, может содержать дефис. Если строка не соответствует формату, то в консоль выводится: Введенная строка не является ФИО.
- Используйте регулярные выражения для определения валидности строки и её деления на фрагменты.

Задание 6

Цели домашнего задания

Научиться писать и использовать несложные регулярные выражения.

Что необходимо сделать

- Выполните задание в классе `main/java/practice/regex/PhoneCleanerRegex.java`
- 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.

Пример ввода номеров и результата вывода программы

Ввод пользователя	Вывод программы в консоль	Примечание
+7 909 123-45-67	79091234567	В номере 11 символов, код страны 7 - верный
+7 (909) 1234567	79091234567	В номере 11 символов, код страны 7 - верный
8-905-1234567	79051234567	В номере 11 символов, первый символ (выхода на мобильный номер) заменяется на код страны 7 — номер верный

9-453-1234567	Неверный формат номера	В номере 11 символов, первый символ очистки 9, это не 7 и не 8 — формат неверный
8-905-123	Неверный формат номера	В номере 7 символов — номер неверный
905-1234567	79051234567	В номере 10 символов после очистки значит, приводим к формату номера с
8-905-12345672342	Неверный формат номера	В номере более 11 символов — номер неверный