

Simple Tasks

Тема. Робота з циклами, масивами та рядками в Java.

****зверніть увагу на 7-му лабораторну роботу***

Завдання. Відповідно до номера у загальному списку слухачів дисципліни створити консольний додаток (список з гугл таблиці всіх слухачів, а не зі своєї групи: 11-й у списку бере 1-й варіант, 12-й – 2-й, ..., 21-й – 1-й, і т.д.):

1. Визначити ті рядки, довжина яких менша (більша) середньої. На вхід поступає масив String. На виході – масив String.
2. Визначити слово, в якому кількість різних символів мінімальна. Якщо таких слів декілька, то визначити перше з таких. На вхід поступає рядок із словами. Словом вважається послідовність символів між пробілами та символами «white space». На виході – масив String.
3. Знайти ті слова, які містять тільки символи латинського алфавіту. Серед них знайти ті слова, які містять рівну кількість голосних та приголосних. На вхід поступає рядок із словами. На виході – масив String.
4. Знайти слова, символи в яких йдуть в порядку зростання їх кодів. На вхід поступає рядок із словами. На виході – масив String.
5. Знайти слова, які складаються тільки з різних символів. На виході – масив String.
6. Серед простих чисел, які не перевищують заданий n, знайти таке, в двійковій формі якого максимальна кількість одиниць. Просте число – це натуральне число, яке ділиться на 1 та на себе.
7. Серед простих чисел, які не перевищують заданий n, знайти таке, в двійковій формі якого максимальна кількість нулів. Просте число – це натуральне число, яке ділиться на 1 та на себе.
8. Розробити функцію, яка визначає кількість надпростих чисел в натуральному ряді. Функція приймає на вхід натуральне число не більше 1000. Надпростим називається число, якщо воно є простим та число, яке отримане з нього записом цифр у зворотному порядку, теж є простим.

9. Знайти всі досконалі числа в діапазоні від 1 до деякого заданого позитивного числа (досконале число – це число, яке дорівнює сумі всіх своїх дільників, крім самого себе. Наприклад, $6 = 1+2+3$).
10. Розробити функцію (не використовувати готові методи класів обгортки), яка визначає ціле позитивне число в десятковій системі, потім перетворює його в двійкову / вісімкову/ шістнадцяткову системи числення.

Hard Task

Тема. Засвоєння структури JVM. Написання власного завантажувача класів

Завдання. Розробити власний class loader, який «на льоту» підвантажує зміни, зроблені в java-файлі.

Даний клас:

```
public class TestModule {  
    @Override  
    public String toString() {  
        return "TestModule, version 1!";  
    }  
}
```

В іншому класі створюється екземпляр класу TestModule і викликається System.out.println(t), при цьому програма продовжує висіти в пам'яті.

Реалізувати, щоб при зміні класу TestModule.java він автоматично перекомпілювався, і його байт-код перевантажувався в JVM, і виводилося повідомлення нового змісту.

Додаткові відомості (*посиланням вже погано):

<http://publib.boulder.ibm.com/infocenter/javasdk/v1r4m2/index.jsp?topic=%2Fcom.ibm.java.diagnostics.142j9%2Fhtml%2Fid1020.html>

<http://www.ibm.com/developerworks/java/tutorials/j-classloader/j-classloader-pdf.pdf>

<http://samag.ru/archive/article/28>

<http://samag.ru/archive/article/68>

<http://samag.ru/archive/article/87>

«Просунуті» відомості про class loader:

<http://zeroturnaround.com/blog/reloading-objects-classes-classloaders/>