

Лабораторная работа 1.

Сдать до 17.9

Реализовать в **двух вариантах**: простые типы и используя классы **BigDecimal**, **BigInteger**.
Использовать класс **Formatter**.

Общее задание

1. Разработать консольное приложение на Java.
2. Вычисление ряда выполнить в отдельном классе (не загрузочном)!
3. Функция представлена в виде своего ряда Тейлора. Вычислить приближённое значение суммы этого бесконечного ряда. Вычисления заканчивать, когда очередное слагаемое окажется по модулю меньше заданного числа ε . Вид этого числа определяется следующим условием:
 $\varepsilon = 10^{-k}$, где k – натуральное число.
4. Значения x и k ввести с клавиатуры.
5. Факториал и степень в слагаемых отдельно не считать!
6. Сравнить полученный результат со значением, вычисленным через стандартные функции.
7. Используя класс **Formatter** реализовать :
 - a. Выводить некоторые целые значения (округлить) в восьмеричном и шестнадцатеричном виде
 - b. Для чисел с плавающей точкой использовать *спецификатор минимальной ширины поля и спецификатор точности (количество знаков после запятой = $k+1$)*
 - c. Использовать *флаги (flags) форматирования: 0, +) #*
8. Использовать для вывода *спецификатор минимальной ширины поля и спецификатор точности* в методе `System.out.printf()`.

Варианты:

1. $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots; \quad \text{где } x \in (-\infty, +\infty)$
2. $\frac{1}{(1+x)^3} = 1 - \frac{2*3}{2}x + \frac{3*4}{2}x^2 - \frac{4*5}{2}x^3 + \dots; \quad \text{где } x \in (-1, +1)$
3. $\ln(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \dots; \quad \text{где } x \in [-1, +1)$
- 4.

$$\sqrt{1+x} = (1+x)^{\frac{1}{2}} = 1 + \frac{x}{2} + \frac{\frac{1}{2}\left(\frac{1}{2}-1\right)}{2!}x^2 + \frac{\frac{1}{2}\left(\frac{1}{2}-1\right)\left(\frac{1}{2}-2\right)}{3!}x^3 + \dots =$$

$$= 1 + \frac{x}{2} - \frac{1 \cdot x^2}{2^2 2!} + \frac{1 \cdot 3 \cdot x^3}{2^3 3!} - \frac{1 \cdot 3 \cdot 5 \cdot x^4}{2^4 4!} + \dots + (-1)^{n+1} \frac{1 \cdot 3 \cdot 5 \cdot (2n-3) x^n}{2^n n!}.$$

$$\sqrt{1+x} = 1 + \frac{1}{2}x - \frac{1}{2^2 * 2!}x^2 + \frac{1*3}{2^3 * 3!}x^3 + \dots + (-1)^{n+1} \frac{1*3*5*...*(2n-3)}{2^n * n!}x^n \dots; \quad \text{где } x \in (-1, +1)$$

$$5. \quad \frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2}x + \frac{1*3}{2*4}x^2 - \frac{1*3*5}{2*4*6}x^3 \dots; \quad \text{где } x \in (-1, +1)$$

$$6. \quad \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots; \quad \text{где } x \in (-\infty, +\infty)$$

$$7. \quad \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \dots; \quad \text{где } x \in (-\infty, +\infty)$$

$$8. \quad \arcsin x = x + \frac{1}{2} * \frac{x^3}{3} + \frac{1*3}{2*4} * \frac{x^5}{5} + \frac{1*3*5}{2*4*6} * \frac{x^7}{7} + \dots; \quad \text{где } x \in (-1, +1)$$

$$9. \quad \frac{\sin x}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \dots; \quad \text{где } x \in (-\infty, +\infty)$$

Лабораторная работа № 2

UNIT-тесты

Общее задание:

- Использовать UNIT-тесты для тестирования лабораторной 1.
- Запись кода и кода тестов на GitHub

Лабораторная № 3

Сдать до 8.10

Использование стандартные методы обработки строк из классов **String**, **StringBuffer****StringBuilder** (более новые), **StringTokenizer**, Использование классов-оболочек (**Integer**, **Double** и т.д.) для простых типов, класс **Arrays**, классы **NumberFormat**, **DecimalFormat(Formatter)**. Для заполнения элементов массивов использовать класс **Random** или метод **random** из **Math**, **Matcher**, **Pattern**, **PatternSyntaxException**, **FileReader**, **FileWriter**, **BufferedReader**, **BufferedWriter**.

Общее задание

1. Запрашивает две строки.

Первая строка содержит лексемы, состоящие из любых символов, которые можно ввести с клавиатуры, например, числа (см. индивидуальные варианты, 2-й или 8-й или 10-й или 16-й систем счисления).

Вторая строка содержит символы разделителя (1 разделитель - 1 символ, но между лексемами может стоять >1 разделителя и разные разделители) .

2. **StringTokenizer** использовать, если между лексемами стоит >1 разделителя и разные разделители). **String.Split** (без регулярного выражения, если только всего один разделитель и по одному разделителю между лексемами или с регулярным выражением).
3. **Использовать методы :**

- Для строк **String**: format, charAt, getChars, Split, Remove, Substring, Replace, IndexOf, LastIndexOf
 - Для строк **StringBuffer\StringBuilder (обязательно)**: delete, insert, reverse
4. Для поиска чисел не из 10с\с использовать перегруженный метод Integer.parseInt.
 5. Считать\записать данные\результаты из\в файл. Использовать классы : FileReader, FileWriter, BufferedReader, BufferedWriter.
 6. Обработку дат и времени осуществлять при помощи регулярных выражений (количество дней во всех месяцах проверять не нужно).
 7. Для валидации дат и времени использовать классы Date, (Calendar), SimpleDateFormat (метод setLenient). Для вывода даты и времени использовать классы Format, DateFormat или Formatter.
- * Записать в Arrays, отсортировать список лексем даты\времени с помощью Comparator.

Индивидуальные задания:

1. Разбить первую строку на лексемы (используя разделители из второй строки), определить в ней целые числа 2-й с\с. Числа записать в новый отдельные массив. Среди лексем не являющихся числами, найти время (ММ\ЧЧ\СС). Добавить в строку случайное число после лексем-времени или в середину строки(если нет) . Подстроку (с самой маленькой длиной), начинающуюся цифрой - удалить из строки. Все результаты сформировать в строки и вывести.
2. Разбить первую строку на лексемы (используя разделители из второй строки), определить в ней целые числа 10-й с\с. Числа записать в новый отдельные массив. Среди лексем не являющихся числами, найти время (ЧЧ-ММ- СС). Добавить в строку случайное число до числа 10й с\с или в начало строки(если нет). Подстроку, заключенные в круглые скобки - удалить из строки. Все результаты вывести. Все результаты сформировать в строки и вывести.
3. Разбить первую строку на лексемы (используя разделители из второй строки), определить в ней целые числа 8-й с\с. Числа записать в новый отдельные массив. Среди лексем не являющихся числами, найти даты (ДД:ММ:ГГ). Добавить в строку случайное число после лексем-даты или в середину строки(если нет) . Подстроку (с самой маленькой длиной), заканчивающуюся цифрой - удалить из строки. Все результаты сформировать в строки и вывести.
4. Разбить первую строку на лексемы (используя разделители из второй строки), определить в ней целые числа 16-й с\с. Числа записать в новый отдельные массив. Среди лексем не являющихся числами, найти время (ММ-ЧЧ-СС). Добавить в строку случайное число после лексем-времени или в середину строки(если нет) . Подстроку (с самой маленькой длиной), начинающуюся с символа - удалить из строки. Все результаты сформировать в строки и вывести.
5. Разбить первую строку на лексемы (используя разделители из второй строки). Лексем записать в новый отдельные массив. Среди лексем, найти лексем состоящие только из символов латинского алфавита и цифр. Среди лексем не являющихся числами, найти время (ММ-ЧЧ-СС).Добавить в строку случайное число после лексем-даты или в середину строки(если нет) . Подстроку (самой большой лексемой) - удалить из строки. Все результаты сформировать в строки и вывести.
6. Разбить первую строку на лексемы (используя разделители из второй строки), определить в ней целые числа 2-й с\с. Числа записать в новый отдельные массив. Среди лексем не являющихся числами, найти время (ММ\ЧЧ\СС). Добавить в строку случайное число после числа 2-й с\с или в середину строки(если нет) . Подстроку (с самой маленькой длиной), начинающуюся цифрой - удалить из строки. Все результаты сформировать в строки и вывести.
7. Разбить первую строку на лексемы (используя разделители из второй строки), определить в ней вещественные числа . Числа записать в новый отдельные массив. Среди лексем не являющихся

числами, найти время (ЧЧ-ММ). Добавить в строку случайное число после вещественного числа или в середину строки(если нет) . Подстроку (с самой маленькой длиной), заканчивающуюся символом-удалить из строки. Все результаты сформировать в строки и вывести.

8. Разбить первую строку на лексемы (используя разделители из второй строки). Лексемы записать в новый отдельные массив. Среди лексем не являющихся числами, найти лексемы состоящие только из символов латинского алфавита. Среди лексем не являющихся числами, найти даты (ММ:ДД:ГГ). Добавить в строку случайное число после лексемы на латинском языке или в середину строки(если нет). Подстроку (с любой лексемой) - удалить из строки. Все результаты сформировать в строки и вывести.

9. Разбить первую строку на лексемы (используя разделители из второй строки). Лексемы записать в новый отдельные массив. Среди лексем не являющихся числами, найти лексемы состоящие только из символов русского алфавита. Среди лексем не являющихся числами, найти даты (ММ ДД ГГ).Добавить в строку случайное число после лексемы на русском или в середину строки(если нет) . Подстроку (самой большой лексемой) - удалить из строки. Все результаты сформировать в строки и вывести.

10. Разбить первую строку на лексемы (используя разделители из второй строки), определить в ней вещественные числа. Числа записать в новый отдельные массив. Среди лексем не являющихся числами, найти даты (ДД\ММ\ГГ). Добавить в строку случайное число до вещественного числа или в конец строки(если нет) . Подстроку (с самой маленькой длиной), заканчивающуюся символом-удалить из строки. Все результаты сформировать в строки и вывести.

Лабораторная работа № 4

Сдать до 5.11 (часть : пункты 1-9)

Сдать до 12.11

Обработка коллекций.XML.JSON.UNIT-тестирование.

Общее задание:

1. Создать абстрактные классы.
2. Создать классы для хранения списков объектов, *List* и *Map* (наследники абстрактного класса).
3. Создать классы для чтения\записи данных в\из файл(а)
4. Создать класс (наследник абстрактного) для хранения полей объекта(согласно индивидуальному проекту), например, автомобиль содержит поля: **id, тип, модель, мощность двигателя, максимальная скорость дата выпуска, цена** и т.д. Создать методы *Get* , *Set...* (автогенерация), переопределить метод *toString()*.
5. Проверка входных данных (данные, числа и т.д.) на корректность(данные изначально считать из текстового файла).
6. Запись объектов данных в *List*, *Map*.
7. Ввод с консоли новых данных для объекта и добавление\обновление данных в *List*, *Map* (и обновление файлов)
8. Использовать итераторы для вывода.
9. Сортировка данных по выбранным полям, используя *Collections*, *Comparator*. Лямбда-выражения.
10. Чтение из XML- файла, запись в XML файл (*DOM* –технология)
11. Чтение из JSON- файла, запись в JSON файл.
12. Шифрование данных.
13. Архивация (*jar*, *zip*, *rar*).
14. Использовать *UNIT*-тесты для тестирования.
15. Запись на *GitHub*

Индивидуальные задания:

1. Фабрика по сборке различных грузовых автомобилей.
2. Фабрика по сборке различных мобильных телефонов.
3. Фабрика по сборке различных кофеварок.
4. Фабрика по сборке различных мультиварок.
5. Фабрика по сборке различных стиральных машин.
6. Кафе, разработка индивидуального меню.
7. Фабрика по сборке различных домов.
8. Фабрика по сборке различных домашних комбайнов.

Лабораторная работа № 5 (проект)

Сдать до 26.11

Паттерны

Общее задание:

Для предыдущей лабораторной реализовать паттерны:

1. Реализовать два варианта с *Design Patterns* (*Decorator* и *Builder*).
 2. Реализовать *Pattern Singleton* для доступа к базе данных (файлу данных)
- Использовать *UNIT*-тесты для тестирования.
 - Запись кода и кода тестов на *GitHub*

Лабораторная работа № 6

Графика и потоки Thread.

Общее задание:

- Создать графическое приложение для предыдущей лабораторной.
- Создать поток(и) *Thread* для параллельной записи\чтения данных в файлы *JSON\XML*.
- Использовать элементы для добавления новой информации и вывода информации:
 1. Использовать диалоговые окна.
 2. Использовать текстовые поля.
 3. Использовать списки.
 4. Использовать таблицы.
 5. И т.д.
- Запись кода и кода тестов на *GitHub*