

## Лабораторная работа 1.

Сдать до 12.9

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

### Общее задание

Разработать консольное приложение на Java.

**Ряд программировать в отдельном классе (не загрузочном)!**

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  $\varepsilon$ . Вид этого числа определяется следующим условием:  $\varepsilon = 10^{-k}$ , где  $k$  – натуральное число.

Сравнить полученный результат со значением, вычисленным через стандартные функции для double значения.

Значения  $x$  и  $k$  ввести с клавиатуры.

Вывод результата осуществить с  $k+1$  знаками после десятичной точки.

### Реализация с BigDecimal, BigInteger:

Выбрать вариант из 3-х пунктов (влияет на количество баллов за лабораторную):

1. Переделать всю лабораторную (9-10 баллов);
2. Вычисление только нескольких математических операций.вычислений (6-8 баллов);
3. Перевод из(в) простых типов в(из) BigInteger(BigDecimal) (4-5 балла).

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

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.

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

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

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

5.  $\frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2}x + \frac{1*3}{2*4}x^2 - \frac{1*3*5}{2*4*6}x^3 \dots; \quad \text{где } x \in (-1, +1)$
6.  $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots; \quad \text{где } x \in (-\infty, +\infty)$
7.  $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \dots; \quad \text{где } x \in (-\infty, +\infty)$
8.  $\arcsin x = x + \frac{1}{2} * \frac{x^3}{3} + \frac{1*3}{2*4} * \frac{x^5}{5} + \frac{1*3*5}{2*4*6} * \frac{x^7}{7} + \dots; \quad \text{где } x \in (-1, +1)$
9.  $\frac{\sin x}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \dots; \quad \text{где } x \in (-\infty, +\infty)$
10.  $\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots; \quad \text{где } x \in (-1, +1)$
11.  $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots; \quad \text{где } x \in (-1, +1]$
12.  $\ln\left(\frac{1+x}{1-x}\right) = 2\left(x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots\right); \quad \text{где } x \in (-1, +1)$
13.  $\sinh x = \frac{e^x - e^{-x}}{2} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots; \quad \text{где } x \in (-\infty, +\infty)$
14.  $\cosh x = \frac{e^x + e^{-x}}{2} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots; \quad \text{где } x \in (-\infty, +\infty)$
15.  $a^x = 1 + \frac{x \ln a}{1!} + \frac{(x \ln a)^2}{2!} + \frac{(x \ln a)^3}{3!} + \dots; \quad \text{где } x \in (-\infty, +\infty)$
16.  $\ln x = 2 * \left( \frac{(x-1)}{x+1} + \frac{1}{3} \left( \frac{x-1}{x+1} \right)^3 - \frac{1}{5} \left( \frac{x-1}{x+1} \right)^5 + \dots \right); \quad \text{где } x > 0$

## Лабораторная работа 2.

Сдать до 19.9

**Для 1 лабораторной использовать класс: Formatter**

**Общее задание для Formatter:**

1. Выводить целые значения в восьмеричном и шестнадцатеричном виде
2. Выводить значения с плавающей точкой
3. Использовать *спецификатор минимальной ширины поля*
4. Использовать *спецификатор точности*
5. Использовать *флаги (flags) форматирования: 0, +) #*
6. Использование *порядкового номера аргумента*

**Примечание: меню не использовать!**

### Лабораторная работа 3.

Сдать до 3.10

#### Обработка двумерных массивов.

Использование **классов-оболочек**(Integer, Double и т.д.) для простых типов, класс **Arrays**, классы **NumberFormat**, **DecimalFormat**. Для заполнения элементов массивов использовать класс **Random** или метод **random** из **Math**.

#### Общее задание:

1. В вариантах, где необходимо менять строки (столбцы) матрицы местами - циклы для переноса всех элементов не использовать, а работать как с указателями на массивы!
2. Для целых значений таблицы , выводить данные используя NumberFormat;
3. Для действительных данных таблицы, выводить данные DecimalFormat.
4. Две любые строки строки (или столбцы) матрицы выводить в числовом формате,% и денежных единицах.
5. Для некоторых вариантов вводить матрицу(ы) с консоли .
6. Коллекции (Collections)– не использовать!
7. Создать объект класса Local либо с текущими настройками либо новые (страна и язык).

1. Даны две действительные квадратные матрицы порядка  $n$ . Получить новую матрицу умножением элементов каждой строки первой матрицы на наибольшее из значений элементов соответствующей строки второй матрицы. Отсортировать в последней строке элементы по убыванию, используя Comparator и реализовать бинарный поиск элемента в строке(стандартный метод).

2. Даны две действительные квадратные матрицы порядка  $n$ . Получить новую матрицу путем прибавления к элементам каждого столбца первой матрицы произведения всех элементов соответствующих строк второй матрицы. Произведения вывести на консоль. Частично отсортировать первую строку по убыванию с индекса  $i$  по  $j$ , используя Comparator. Частично скопировать строку в другой массив.

3. Дана целочисленная матрица порядка  $n*m$ . Переставляя ее строки и столбцы, добиться, чтобы наибольший элемент (один из них) оказался в верхнем левом углу. Вывести на экран полученную матрицу. Отсортировать в последней строке элементы по убыванию, используя Comparator. Частично скопировать строку в другой массив.

4. Дана целочисленная матрица порядка  $n*m$ . Переставляя ее строки и столбцы, добиться, чтобы наименьший элемент (один из них) оказался в нижнем правом углу. Вывести на экран полученную матрицу. Частично отсортировать первую строку по убыванию с индекса  $i$  по  $j$ , используя Comparator и реализовать бинарный поиск элемента в строке(стандартный метод).

5. В данной действительной квадратной матрице порядка  $n$  найти  $\max$  по модулю элемент. Получить квадратную матрицу порядка  $n-1$  путем выбрасывания из исходной какой-либо строки и столбца, на пересечении которых расположен элемент с найденным значением. Отсортировать в последней строке элементы по убыванию, используя Comparator и реализовать бинарный поиск элемента в строке(стандартный метод).

6. В данной действительной квадратной матрице порядка  $n$  найти  $\min$  элемент. Получить квадратную матрицу порядка  $n+1$  путем добавления к исходной строки и столбца, на пересечении которых расположен элемент с найденным значением. Заполнить строку и столбец элементом  $\min$ . Частично отсортировать первую строку по убыванию с индекса  $i$  по  $j$ , используя Comparator Частично скопировать строку в другой массив.

7. Характеристикой столбца целочисленной матрицы назовем сумму модулей его отрицательных нечетных элементов. Переставляя столбцы заданной матрицы, расположить их в соответствии с ростом характеристик. Характеристики вывести на консоль. Отсортировать в последней строке элементы по убыванию, используя Comparator. Частично скопировать строку в другой массив.
8. Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик. Характеристики вывести на консоль. Частично отсортировать первую строку по убыванию с индекса  $i$  по  $j$ , используя Comparator и реализовать бинарный поиск элемента в строке(стандартный метод).
9. Для двух заданных действительной матриц одинакового размера проверить, можно ли получить вторую матрицу из первой применением (конечного числа раз) операций транспонирования относительно главной и побочной диагоналей. Отсортировать в последней строке элементы по убыванию, используя Comparator и реализовать бинарный поиск элемента в строке(стандартный метод).
10. Взаимно однозначное отображение элементов матрицы на себя можно задать с помощью двух целочисленных матриц: в первой указывать номер строки, куда переходит данный элемент, а во второй матрице — номер столбца, если элемент остаётся на месте - помечаем 0. Построить три матрицы, задающие отражение каждого элемента матрицы на симметричный ему относительно главной диагонали. Частично отсортировать первую строку по убыванию с индекса  $i$  по  $j$ , используя Comparator. Скопировать в другой массив.
11. Взаимно однозначное отображение элементов матрицы на себя можно задать с помощью двух целочисленных матриц: в первой указывать номер строки, куда переходит данный элемент, а во второй — номер столбца. Построить две матрицы, задающие отражение каждого элемента матрицы на симметричный ему относительно побочной диагонали. Отсортировать в последней строке элементы по убыванию, используя Comparator. Скопировать строку в другой массив.
12. Выведите номера столбцов, все элементы, которых четны. Для каждого столбца с отрицательным элементом на главной диагонали вывести его номер и сумму всех элементов этого столбца. Частично отсортировать первую строку по убыванию с индекса  $i$  по  $j$ , используя Comparator и реализовать бинарный поиск элемента в строке(стандартный метод).
13. Выведите номера строк, с максимальным количеством повторяющихся элементов. Частично отсортировать первую строку по убыванию с индекса  $i$  по  $j$ , используя Comparator и реализовать бинарный поиск элемента в строке(стандартный метод).
14. Дана целочисленная матрица порядка  $n*m$ . Переставляя ее строки и столбцы, добиться, чтобы наибольший элемент (один из них) оказался в верхнем правом углу. Вывести на экран полученную матрицу. Отсортировать в последней строке элементы по возрастанию, используя Comparator и реализовать бинарный поиск элемента в строке(стандартный метод).
15. Дана целочисленная матрица порядка  $n*m$ . Переставляя ее строки и столбцы, добиться, чтобы наименьший элемент (один из них) оказался в нижнем левом углу. Вывести на экран полученную матрицу. Частично отсортировать среднюю строку (или одну из двух средних ) по возрастанию с индекса  $i$  по  $j$ , используя Comparator. Частично скопировать строку в другой массив.

#### **Дополнительно:**

Создать объект класса Local либо с текущими настройками либо новые (страна и язык).

## Лабораторная № 4

Сдать до 17.10

### Обработка строк.

**Использование стандартные методы обработки строк из классов `String`, `StringBuffer`, `StringTokenizer` и метод `String.format`.**

#### Общее задание

1. Запрашивает две строки и целое число `P`- для поиска.

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 разделители (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`** –если только всего один разделитель и по одному разделителю между лексемами.

3. Вывод чисел, осуществлять с помощью метода **`String.format`**.

4. **Использовать методы :**

- Для строк **`String`**: `format`, `charAt`, `getChars`, `Split`, `Remove`, `Substring`, `Replace`, `IndexOf`, `LastIndexOf`
- Для строк **`StringBuffer`**: `delete`, `insert`, `reverse`

5. Для поиска чисел не из 10с\с использовать перегруженный метод `Integer.parseInt`.

6. **Обработку регулярных выражений – не использовать.**

7. **Коллекции (`Collections`)– не использовать!**

#### Индивидуальные задания

1. 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 лексемы являющиеся палиндромами. Найти число `P` (если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку случайное число до числа `P` или в начало строки(если нет `P`). Подстроку, заключенные в круглые скобки - удалить из строки. Все результаты сформировать в строки и с помощью `String.format` и вывести.

2. 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 лексемы не являющиеся палиндромами. Найти число `P` (если есть, то должно совпадать с лексемой), вывести позицию в изначальной строке. Добавить в строку случайное число после числа `P` или в середину строки(если нет `P`). Подстроку (с самой маленькой длиной), начинающуюся цифрой - удалить из строки . Все результаты сформировать в строки и с помощью `String.format` 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 лексемы с хотя бы одной парой одинаковых символов и чётным количеством символов. Найти число `P` (если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку случайное число до числа `P` или в начало строки(если нет `P`). Подстроку из цифр (с самой маленькой длиной), - удалить из строки . Все результаты сформировать в строки и с помощью `String.format` и вывести.

4. Разбить первую строку на лексемы (используя разделители из второй строки из второй строки), определить в ней целые числа 10-й с\с. Числа записать в новый отдельный массив. Среди лексем не являющихся числами, найти лексемы состоящие только из одинаковых символов. Найти число

Р(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке.

Продублировать в строке одно из чисел, добавить его после числа. Любую подстроку из знаков препинания - удалить из строки. Все результаты сформировать в строки и с помощью String.format и вывести.

5. 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 являющихся числами, найти лексемы не являющиеся палиндромами. Найти число Р(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Продублировать в строке самое большое число, добавить его в начало строки. Первую лексему с латинскими буквами и цифрами - удалить из строки. Все результаты сформировать в строки и с помощью String.format и вывести.

6. 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 состоящие только из символов латинского . Найти число Р(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку одно из чисел с “-“, добавить его в середину строки. Предпоследнее целое число - удалить из строки. Все результаты сформировать в строки и с помощью String.format и вывести.

7. 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 лексемы являющиеся палиндромами. Найти число Р(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку число равное -Р, добавить его в начало строки. Последнее целое число =числу Р - удалить из строки. Все результаты сформировать в строки и с помощью String.format и вывести.

8. 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 лексемы состоящие только из символов русского алфавита. Найти число Р(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку число равное -Р, добавить его после числа Р или после первого целого числа строки. Удалить любую лексему состоящие только из символов русского алфавита . Все результаты сформировать в строки и с помощью String.format и вывести.

9. 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 являющихся числами 16-й с\с , найти лексемы состоящие большего количества ‘1’ чем ‘0’. Добавить в строку случайное число, удалить первую лексему с латинскими буквами . Найти число Р(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Первое число 16-й с\с - удалить из строки. Все результаты сформировать в строки и с помощью String.format и вывести.

10. 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, найти лексемы, состоящие из цифр и из символов латинского алфавита. Найти число Р(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку квадрат числа Р , поместить это число после первой лексемы строки. Последнее число 2-й с\с - удалить из строки. Все результаты сформировать в строки и с помощью String.format и вывести.

11. 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 лексемы состоящие только из знаков препинания. Найти число Р (если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку случайное число, добавить его после числа Р или перед первым целым числом строки. Последнее

целое число < числа P - удалить из строки. Все результаты сформировать в строки и с помощью String.format и вывести.

12. Разбить первую строку на лексемы (используя разделители из второй строки), определить в ней целые числа 16-й с\с. Числа записать в новый отдельный массив. Среди лексем найти лексемы, состоящие не из цифр и не из символов латинского и не русского алфавита, удалить одну из таких лексем из строки. Найти число P(если есть, то должно совпадать с лексемой), вывести позицию в изначальной строке. Продублировать в строке самое большое число, добавить его в начало строки. Все результаты сформировать в строки и с помощью String.format и вывести.

13. Разбить первую строку на лексемы (используя разделители из второй строки), определить в ней целые числа 2-й с\с. Числа записать в новый отдельный массив. Среди лексем являющихся числами, являющихся 2 с\с - найти четные, удалить одну из таких лексем из строки. Найти число P(если есть, то должно совпадать с лексемой), вывести позицию в изначальной строке. Добавить модуль числа P в начало строки. Все результаты сформировать в строки и с помощью String.format и вывести.

14. Разбить первую строку на лексемы (используя разделители из второй строки), определить в ней целые числа 16-й с\с. Числа записать в новый отдельный массив. Среди лексем найти лексемы, состоящие из цифр и из символов русского алфавита. Найти число P(если есть, то должно совпадать с лексемой), вывести позицию в изначальной строке. Найти в строке первое число < 0, добавить его в середину строки. Первое целое число < числа P - удалить из строки. Все результаты сформировать в строки и с помощью String.format и вывести.

### Лабораторная № 5

Сдать до 30.10

#### Регулярные выражения.

Использовать классы : Matcher, Pattern, PatternSyntaxException.

#### Общее задание:

1. Данные считать из файла ( > =10 тестовых строк).
2. Записать результаты в файл. Использовать классы : FileReader, FileWriter, BufferedReader, BufferedWriter.
3. Можно использовать коллекции (класс ArrayList).

1. Составить регулярное выражение, является ли заданная строчка IP адресом, записанным в десятичном виде

Пример правильных выражений:

127.0.0.1  
255.255.255.0  
192.168.0.1

Пример неправильных выражений:

1300.6.7.8  
abc.def.gha.bcd  
254.hzf.bar.10

2. Написать регулярное выражение определяющее является ли данная строчка GUID с или без скобок. Где GUID это строчка, состоящая из 8, 4, 4, 4, 12 шестнадцатеричных цифр разделенных тире.

Пример правильных выражений:

{e02fa0e4-01ad-090A-c130-0d05a0008ba0}  
e02fd0e4-00fd-090A-ca30-0d00a0038ba0

Пример неправильных выражений:

02fa0e4-01ad-090A-c130-0d05a0008ba0}  
e02fd0e400fd090Aca300d00a0038ba0



3. Написать регулярное выражение определяющее является ли заданная строка правильным MAC-адресом( числа в 16 с\с).

Пример правильных выражений:

01:32:54:67:89:AB

aE:dC:cA:5A:76:54

Пример неправильных выражений:

01:33:47:65:89:ab:cd

01:23:45:67:89:Az

4. Проверить является ли заданная строка числом, не обязательно целым или может быть в экспоненциальном виде, записанным в десятичной системе счисления (может быть с нулями в старших разрядах).

Пример правильных выражений:

123456

234567

000333

0333.1121

2424,423423

-1.17E-08

-1.17E-08

1.17E8

1.17e8

Пример неправильных выражений:

113..444

16. 12345

23232312?33

1.17E+08

1.17EE+08

1.17Ee+08

1.17Ee++8

\*90875

!>>№;%?\*

5. Написать регулярное выражение определяющее является ли данная строчка номером паспорта и местом выдачи. Проверять название на существование такового - не нужно. Городам соответствуют буквы в номере : Минск – МР, Брест – АВ, и т.д.; областные города –МС. КВ, РР - паспорта, выданные Министерством иностранных дел (в случае проживания за рубежом).

Пример правильных выражений:

AB1234567 Московский РУВД Брест

BM3523213 Октябрьский РУВД Витебск

HB3523213 Советский РУВД Гомель

KN3523213 Московский РУВД Гродно

MP3523213 Московский РУВД Минск

MS3523213 РУВД Слуцк

KB3523213 Ленинский РУВД Могилев

PP8908213 Фрунзенский РУВД Минск

Пример неправильных выражений:

AB123456 Московский РУВД Брест

BM1234567 Октябрьский РУВД

HB12345678 Советский Гомель

KN1234567 Московский РУВД

MM1234567 Московский РУВД Минск

MS1234567 РУВД

KB1234567 РУВД Могилев

PP1234567 Минск

BM1234567 РУВД Гомель

K1234567 Московский РУВД Гродно

MФ1234567 Московский РУВД Минск

6. Написать регулярное выражение определяющее является ли данная строчка датой в формате dd/mm/yyyy . Начиная с 0001 года до 1000 года



Пример правильных выражений:

29/02/0899

30/04/0999

01/01/0055

Пример неправильных выражений:

29/02/2001

30-04-2003

1/1/1899

**7.** Написать регулярное выражение определяющее является ли данная строчка адресом в соответствующем формате : город, улица\проспект\переулок, дом, квартира. Проверять название на существование такового - не нужно.

Пример правильных выражений:

г. Минск, ул. Притыцкого, д.15, кв.311

г. Минск, пр. Независимости, д.10, кв.22

г. Минск, пер. Велосипедный, д.15, кв.1

г. Москва, пер. Велосипедный, д.15

г. Вапвап, пер. Пваппрврвапр, д.15

Пример неправильных выражений:

гг. Минск, пер. Велосипедный, д.15, кв. 1

гМосква, пер. Велосипедный, д.15

г. Минск, Притыцкого, д.15, кв. 311

г.Москва, пер. Велосипедный, д.15

г. Минск, пр. Независимости

г. Минск, п. Велосипедный, д.15, кв.1

г. Минск, пр. Велосипедный, д.15, кв.

г. Москва, пер. Велосипедный, д.

г. Минск, , д.15, кв. 1

пер. Велосипедный, д.15, кв. 1

Минск, ул. Притыцкого, д.15, кв. 311

**8.** Написать регулярное выражение определяющее является ли данная строчка номером телефона в формате : сотовая связь (три оператора) и городская связь. Пример правильных выражений:

+375291234567

+375441234567

+375331234567

+375251234567

375291234567

80171234567

Пример неправильных выражений:

+3752944123456

+37528441234

+375204012345

801631234569

8016123456

**9.** Проверить, надежно ли составлен пароль. Пароль считается надежным, если он состоит из 8 или более символов. Где символом может быть английская буква, цифра и знак подчеркивания. Пароль должен содержать хотя бы одну заглавную букву, одну маленькую букву и одну цифру.

Пример правильных выражений:

C00l\_Pass

SupperPas1

Пример неправильных выражений:

Cool\_pass

C00l

**10.** Написать регулярное выражение определяющее является ли данная строчка валидным URL адресом. В данной задаче правильным URL считаются адреса http и https, явное указание протокола также может отсутствовать. Учитываются только адреса, состоящие из символов, т.е. IP адреса в качестве URL не присутствуют при проверке. Допускаются поддомены, указание порта доступа

через двоеточие, GET запросы с передачей параметров, доступ к подпапкам на домене, допускается наличие якоря через решетку. Однобуквенные домены считаются запрещенными. Запрещены спецсимволы, например «-» в начале и конце имени домена. Запрещен символ «\_» и пробел в имени домена. При составлении регулярного выражения ориентируйтесь на список правильных и неправильных выражений заданных ниже.

Пример правильных выражений:

<http://www.zcontest.ru>  
<http://zcontest.ru>  
<http://zcontest.com>  
<https://zcontest.ru>  
<https://sub.zcontest-ru.com:8080>  
[http://zcontest.ru/dir%201/dir\\_2/](http://zcontest.ru/dir%201/dir_2/)  
[zcon.com/index.html#bookmark](http://zcon.com/index.html#bookmark)

Пример неправильных выражений:

Just Text.  
<http://a.com>  
<http://www.domain-.com>

**11.** Написать регулярное выражение определяющее является ли данная строка временем в формате ss/mm/hh, ss:mm:hh или ss-mm-hh

Пример правильных выражений:

01/01/12  
59/59/23  
00-00-24

Пример неправильных выражений:

61/01/12  
01/61/12  
01/33/25  
1/33/22  
59/59-23  
19/09-23  
00-00-223

**12.** Написать регулярное выражение определяющее является ли данная строка валидным E-mail адресом: >2 домена<=3 (глубина), количество любых конкретных регионов>=5(ru,by,ua,pl,com и т.д), в названии может быть тире (>=1), точка, латинские символы и цифры.

Пример правильных выражений:

[mail@mail.ru](mailto:mail@mail.ru)  
[valid@megapochta.com](mailto:valid@megapochta.com)  
[aa@aa.info](mailto:aa@aa.info)

Пример неправильных выражений:

[bug@@@com.ru](mailto:bug@@@com.ru)  
[@val.ru](mailto:@val.ru)  
Just Text2  
[val@val](mailto:val@val)  
[val@val.a.a.a](mailto:val@val.a.a.a)  
[12323123@111\[\] \[\]](mailto:12323123@111[] [])

**13.** Написать регулярное выражение определяющее является ли данная строка индивидуальным номером и местом выдачи паспорта. Проверять название на существование такового - не нужно. Городам соответствуют буквы в номере А – Минск, В – Минская область, С – Брестская, К – Гродненская, Е – Витебская, М – Могилевская, Н – Гомельская. Три последующие цифры – это порядковый номер человека, обратившегося за паспортом, из числа родившихся с ним в один день и год. Еще две буквы – признак гражданства (РВ – Республика Беларусь, ВА – лицо без гражданства, постоянно проживающее в республике, ВІ – иностранный гражданин, постоянно проживающий в стране).

Пример правильных выражений:

0000456C001PB3 Московский РУВД Брест  
7123456E001PB9 Октябрьский РУВД Витебск  
1123456H001PB7 Советский РУВД Гомель  
2020456K001PB3 Московский РУВД Гродно

1100056A001PB5 Московский РУВД Минск  
7123456B001PB3 РУВД Слуцк  
2123456M001PB2 Ленинский РУВД Могилев  
7123456A001PB4 Фрунзенский РУВД Минск  
7123456H001PB1 Ленинский РУВД Гомель  
7123456K001 BA3 Московский РУВД Гродно  
7123456A001 BI1 Московский РУВД Минск

Пример неправильных выражений:

0000456C001PB3 Московский РУВД Брест  
7123456E001PB9 Октябрьский РУВД  
1123456H001PB7 Советский Гомель  
2020456K001PC3 Московский РУВД Гродно  
1100056A001PB5 Московский РУВД Минск  
7123456B001PB3 РУВД  
2123456M001PB2 РУВД Могилев  
7123456A001PB4 Минск  
57123456H001PB1 РУВД Гомель  
7123456K001 BA Московский РУВД Гродно  
7123456A001 I1 Московский РУВД Минск

**14.** Написать регулярное выражение определяющее является ли данная строка шестнадцатичным идентификатором цвета в HTML. Где #FFFFFF для белого, #000000 для черного, #FF0000 для красного и.т.д.

Пример правильных выражений:

#FFFFFF  
#FF3421  
#00ff00

Пример неправильных выражений:

232323  
f#fddee  
#fd2

### Дополнительное задание:

1. С помощью метода find (class Matcher) найти в строке с тегами HTML-документа все теги , строки между тегами, атрибуты. Результаты вывести.
2. С помощью регулярного выражения заменить слова в тексте на новые.

### Лабораторная работа № 6

#### Форматирование даты и времени.

Сдать до 7.11

Использовать Классы **Date**, **Calendar**, **DateFormat**, **Formatter**, **SimpleDateFormat**

#### Общее задание:

1. Ввод, вывод(преобразование из строки в **Date**, **Calendar** и наоборот).
2. Проверить данные с неправильными : месяцами(>12), номерами дней месяца (>31), часами, минутами, секундами (стандартный метод форматирующего класса )
3. Для **Calendar** изменять данные , использовать методы :settime, add,roll.
4. Получить текущую дату.

#### Индивидуальные задания

Для **Calendar** вывести и изменять:

1. Дата, эру (до нашей эры или после), год
2. эру (до нашей эры или после), года, месяц
3. год, месяц, час
4. месяц, час, минуты

5. час, минуты , секунды
6. минуты , порядковый номер дня в неделе, миллисекунды
7. секунды ,миллисекунды, день в году
8. миллисекунды, возвратит день в году ,день месяца
9. возвратит день в году ,день месяца ,порядковый номер дня в неделе
10. день месяца ,порядковый номер дня в неделе, порядковый номер недели в текущем месяце
11. секунды, порядковый номер дня недели в текущем месяце, номер недели в месяце
12. порядковый номер дня недели в текущем месяце, номер недели в месяце, номер недели в году
13. номер недели в месяце, номер недели в году, индикатор до обеда/после обеда

### Для SimpleDateFormat вывести:

#### Вариант

- 13,1,8,5 **эра** (в английской локализации - AD и BC)
- 14,2, 9 **год** (4-х значное число)
- 3,10 **год** (последние 2 цифры)
- 4,1 **год** (4-х значное число)
- 14,5,2 **номер месяца** без лидирующих нулей
- 6,3 **номер месяца** (с лидирующими нулями если номер месяца < 10)
- 7,4 **четырёх буквенное сокращение месяца** в русской локализации (трех буквенное - в английской : Feb)
- 15,9,6 **полное название месяца** (в английской локализации - February)
- 13,10,7 **неделя в году** без лидирующих нулей
- 11,8 **неделя в году** с лидирующими нулями
- 14,12,9 **неделя в месяце** без лидирующих нулей
- 1,10 **неделя в месяце** с лидирующим нулем (если это необходимо)
- 14,2,11 **день в году**
- 13,3,12 **день месяца** без лидирующих нулей
- 15,4,11 **день месяца** с лидирующими нулями
- 5,12 **день недели** в месяце без лидирующих нулей
- 15,6,3 **день недели** в месяце с лидирующими нулями
- 7,10 **день недели** (сокращение)
- 13,8,2 **день недели** (полностью)
- 9,1 **АМ/РМ** указатель
- 14,10,2 **часы** в 24-часовом формате без лидирующих нулей
- 15,11,3 **часы** в 24-часовом формате с лидирующим нулем
- 12,1,5 **Час** в 24-часовом формате
- 4,2,6 **время** в 12-часовом формате без лидирующих нулей
- 13,3,7 **время** в 12-часовом формате с лидирующим нулем
- 14,4,8 **минуты** без лидирующих нулей
- 15,5,9 **минуты** с лидирующим нулем
- 15,6,10 **секунды** без лидирующих нулей
- 7,11 **секунды** с лидирующим нулем
- 13,8,12 **миллисекунды**
- 9,1 **часовой пояс**

### Для Formatter вывести:

#### Вариант

- 1,3,11 **Сокращенное название дня недели**
- 2,4,9 **Полное название дня недели**
- 3,5, 14 **Сокращенное название месяца**
- 4,6,13 **Полное название месяца**
- 5,7,14 **Стандартное представление в виде: день, месяц чч:мм:сс часовой пояс, год**
- 6,8, 14 **Первые две цифры года**
- 7,9,13 **День месяца как десятичное целое (от 01 до 31)**
- 8,10,15 **месяц/день/год**

11,4	День месяца как десятичное целое (от 1 до 31)
10,12	год-месяц-день
12,2, 14	Час (от 00 до 23)
1,6	Час (от 1 до 12)
2,7,3	Номер дня в году как десятичное целое (от 001 до 366)
5,10	Миллисекунды (от 000 до 999)
6,11, 15	Десятичный номер месяца (от 01 до 13)
7,12	Минуты как десятичное целое (от 00 до 59)
8,1, 15	Наносекунды (от 000000000 до 999999999)
9,3,8, 15	Региональный эквивалент АМ или РМ заглавными буквами
10,4	Региональный эквивалент АМ или РМ строчными буквами
13,11,	Миллисекунды от 1/1/1970
1, 13	чч:мм (12-часовой формат)
2,9	чч:мм (24-часовой формат)
12,4	Секунды (от 00 до 60)

## Лабораторная работа № 7

Сдать до 14.11

### Обработка коллекций.

#### Общее задание:

- *Использовать любой из контейнеров: ArrayList(хранить массивы и матрицы), LinkedList.*
- *Использовать итераторы для вывода.*
- *чтение из текстового файла, запись в текстовый файл. Использовать классы : FileReader, FileWriter, BufferedReader, BufferedWriter.*
- *Использовать лямбда-выражения, например, для вывода List.*

*Примечание: значение возвращаемое hashCode, должно быть в пределах количества элементов у MAP(в этом случае удастся увидеть отсортированные объекты по соответствующему полю в объекте ключа).*

### Индивидуальные задания

1. Дана матрица из целых чисел. Найти в ней прямоугольные подматрицы, состоящие из одинаковых элементов. Использовать класс TreeMap.
2. На плоскости задано N точек. Вывести в файл описания всех прямых, которые проходят более чем через одну точку из заданных. Для каждой прямой указать, через сколько точек она проходит. Использовать класс HashMap. Отсортировать по количеству точек, т.е. переопределить hash код для объекта ключа в HashMap.
3. На клетчатой бумаге нарисован круг, во входном файле: размерность листа N\*M клеток координаты центра и радиус круга. Вывести в файл описания всех клеток, целиком лежащих внутри круга, в порядке возрастания расстояния от клетки до центра круга. Фигуру хранить в матрице. Использовать класс TreeMap.
4. На плоскости задано N отрезков. Найти точки пересечения двух отрезков. Использовать класс TreeMap.
5. На прямой гоночной трассе стоит N автомобилей, для каждого из которых известны начальное положение и скорость. Вывести время и названия машин первых K обгонов. Использовать класс HashTable. Отсортировать по времени обгонов для конкретного авто, т.е. переопределить hash код для объекта ключа в HashTable.

6. Реализовать класс "черный ящик", хранящий группы различных множеств чисел :целых, вещественных, дробных (числитель и знаменатель), комплексных и имеющую внутренний счетчик K, изначально равный нулю. Класс должен поддерживать операции добавления числа в множество и самих множеств и вывод данных. Вывести множества с минимальным количеством чисел K. Использовать класс TreeMap.
7. Задан файл с текстом на английском языке. Выделить слова. Определить номера строк, столбцов и номер предложения для каждого слова. Использовать класс HashMap. Отсортировать по номеру предложения, т.е. переопределить hash код для объекта ключа в HashMap.
8. Во входном файле хранится информация о системе главных автодорог, связывающих города Беларуси. Используя эту информацию, постройте граф, отображающее систему дорог республики, а затем, продвигаясь по графу, определить минимальный по длине путь из г.Минска в другой заданный город. Предусмотреть возможность для последующего сохранения дерева в виртуальной памяти. Используйте класс TreeMap.
9. Во входном файле хранится телефонная книга. Организовать поиск по фамилии абонентов, адресу абонентов. Используйте класс HashTable. Отсортировать по телефону, т.е. переопределить hash код для объекта ключа в HashTable.
10. На клетчатом листе бумаги закрашена часть клеток. Выделить все различные фигуры, которые образовались при этом. Фигуру хранить в матрице . Фигурой считается набор закрашенных клеток, достижимых друг из друга при движении в четырёх направлениях. Две фигуры являются различными, если их нельзя совместить поворотом на угол, кратный 90 градусам, и параллельным переносом. Используйте класс HashTable. Отсортировать по количеству клеток в фигуре, т.е. переопределить hash код для объекта ключа в HashTable.
11. Задан файл с текстом на английском языке. Выделить все различные слова. Для каждого слова подсчитать частоту его встречаемости. Слова, отличающиеся регистром букв, считать различными. Использовать класс HashMap. Отсортировать по слову, т.е. переопределить hash код для объекта ключа в HashMap.
12. Сложить два многочлена заданной степени и вычислить, коэффициенты многочленов хранятся в объекте. Использовать HashTable. Отсортировать по степени у коэффициента, т.е. переопределить hash код для объекта ключа в HashTable.
13. Задан файл с фамилиями игроков и их результатами на соревнованиях. Организовать поиск и вывести в файл: победителей, выбывших, у которых результаты находятся в заданном диапазоне. Используйте класс TreeMap.

### Лабораторная № 8.1 (проект 1-я часть)

Сдать до 6.12

#### Обработка Базы данных

#### Общее задание

#### Создать проект:

1. Создать файлы для хранения данных , в каждом - не меньше 10 записей (предметов, людей и т.д). Для каждого поля типа String в файле должно быть выделено по максимуму одинаковое количество байт во всех строках, например:

23 байта	8 байт	4 байта	10 байт	4 байта
название товара	вид	цена	дата поступления	количество

*товары – Блокнот					
Файл	Правка	Формат	Вид	Справка	
ASUS TUF Gaming FX505DT-HN540	ноутбук	2250	01.01.2000	5	
Apple iPad 10.2	планшет	1299	05.01.2000	3	
POCO X3 NFC	смартфон	899	07.01.2000	10	

- Данные для коллекций (List или Map). читаются из текстового файла.
- Создать абстрактные классы, создать классы-наследники, которые переопределяет некоторые методы (хотя бы 1 метод).
- Разработать собственные интерфейсы и реализовать их методы .
- Использовать для частичного чтения и частичного изменения информации в файле класс RandomAccessFile, использовать метод seek.
- Обеспечить возможность шифрования паролей (встроенный стандартный алгоритм): доступ к информации зарегистрированных пользователей.
- Использовать сериализацию объектов в(из) бинарный формат (файл).
- Использовать сериализацию объектов в(из) XML формат(а) .Использовать XStream или XMLEncoder/XMLDecoder.
- Использовать запись в XML формат с помощью DOM .
- Общая схема проекта (не вся):

```

abstract class Abstractclass
{
public ...
}
class BD extends Abstractclass
{
    List    // или Map. Сюда добавляются объекты класса Element
    ...
}
class Element //согласно индивидуальному варианту
{
...string name;
...string ...;
    string ...
}

```

### Индивидуальные задания:

#### 1. Туристическое агентство.

Агентство должно иметь сведения:

*об экскурсионных маршрутах:* название маршрута, начальный пункт, конечный пункт, протяженность пути, дата проведения

#### 2. Аэропорт.

Управление аэропортом должно иметь сведения:

*о самолётах:* номер, марка, дата выпуска, грузоподъемность, количество пассажиров;

#### 3. Автосалон.

Управление автосалона должно иметь сведения:

*об автомобилях:* марка, пробег на момент приобретения, стоимость, объем двигателя, дата выпуска



#### **4. Ипподром**

Управление ипподромом должно иметь сведения:

*о лошадях:* кличка лошади, возраст, стаж участия в соревнованиях, владелец, дата рождения

#### **5. Автопарк**

Управление автопарком должно иметь сведения:

*о водителях:* табельный номер, фамилия водителя, категория, стаж, адрес, дата рождения

#### **6. Компания по продаже компьютеров**

Дирекция компании должна иметь сведения:

*о товаре:* название товара, вид, цена единицы товара на момент поступления, цена единицы товара на данный момент, дата поступления

#### **7. Атттракционы**

Городская служба должна иметь сведения:

*об аттракционах:* номер аттракциона, название аттракциона, дата приобретения, срок службы, номер площадки;

#### **8. Автомастерские**

Городская служба должна иметь сведения:

*о мастерских:* номер автомастерской, адрес, перечень марок ремонтируемых машин, список мастеров, дата открытия

#### **9. Морской круиз**

Управление должно иметь сведения:

*о теплоходах:* название теплохода, регистрационный номер, дата выпуска, предельный срок эксплуатации,

#### **10. Реклама на ТВ**

Генеральный директор должен иметь сведения:

*о передачах:* название передачи, номер канала, дата и время выхода в эфир, стоимость 1 мин рекламного времени, кол-во пауз, фамилия ведущего;

#### **11. Оптовый магазин по продаже CD и DVD дисков**

Управление магазином должно иметь сведения:

*о маклерах:* фамилия маклера, адрес, дата рождения, процент с продажи;

#### **12. Юридическая компания**

Юридическая компания должны иметь сведения:

*о юристах:* фамилия, категория, стаж, адрес, дата рождения, постоянная зарплата, процент за квалификацию и стаж работы юриста;

#### **13. Вокзал**

Управление вокзалом должно иметь сведения:

*о поездах:* номер, дата выпуска, количество вагонов, количество пассажиров, количество билетов (плацкарт, купе и т.д.), маршрут;

