Лабораторная работа 1.

Сдать до 18.9

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 ε. 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. $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots; \quad где\ x \in (-\infty, +\infty)$
2. $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \ldots; \quad где\ x \in (-\infty, +\infty)$
3. $\arcsin x = x + \frac{1}{2} * \frac{x^3}{3} + \frac{1*3}{2*4} * \frac{x^5}{5} + \frac{1*3*5}{2*4*6} * \frac{x^7}{7} + \ldots; \quad где\ x \in (-1, +1)$
4. $\frac{\sin x}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \ldots; \quad где\ x \in (-\infty, +\infty)$
5. $\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots; \quad где\ x \in (-1, +1)$
6. $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots; \quad где\ x \in (-1, +1]$

7. $\ln(\frac{1+x}{1-x}) = 2(x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + ...); \quad где\, x \in (-1, +1)$

8. $\sinh x = \frac{e^x - e^{-x}}{2} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + ...; \quad где\, x \in (-\infty, +\infty)$

9. $\cosh x = \frac{e^x + e^{-x}}{2} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + ...; \quad где\, x \in (-\infty, +\infty)$

10. $a^x = 1 + \frac{x \ln a}{1!} + \frac{(x \ \ln a)^2}{2!} + \frac{(x \ln a)^3}{3!} + ...; \quad где\, x \in (-\infty, +\infty)$

11. $\ln x = 2 * \left(\frac{(x-1)}{x+1} + \frac{1}{3}\left(\frac{x-1}{x+1}\right)^3 - \frac{1}{5}\left(\frac{x-1}{x+1}\right)^5 + ... \right); \quad где\, x > 0$

Лабораторная работа 2.

Сдать до 25.9

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 9.10

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;
Для действительных данных таблицы, выводить данные DecimalFormat.
Две любые строки строки (или столбцы) матрицы выводить в числовом формате,% и денежных единицах.

4. Для некоторых вариантов вводить матрицу(ы) с консоли .

5. Коллекции (Collections)– не использовать!

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.

Лабораторная № 4
Сдать до 23.10
Обработка строк.
Использование стандартные методы обработки строк из классов String, StringBuffer (StringBuilder более новый), StringTokenizer и метод String.format.

Общее задание

1. Запрашивает две строки и целое число Р- 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. **Использовать Collections. Добавить лексемы в ArrayLis. Отсортировать лексемы, используя лямбда-выражение(выбрать индивидуальный вариант):**
   1) **По наличию в палиндроме среднего одиночного символа**
   2) **По длине строки**
   3) **По первому элементу строки**
   4) **По последнему элементу строки**
   5) **Количеству цифр**
   6) **По количеству латинских символов**
   7) **По наличию в палиндроме среднего одиночного символа**
   8) **По длине строки**
   9) **По первому элементу строки**
   10) **По последнему элементу строки**
   11) **Количеству цифр**
   12) **По количеству латинских символов**

7. Обработку регулярных выражений можно использовать (дополнительный бал).

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 Р(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку случайное число до числа Р или в начало строки(если нет Р). 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 Р(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 P(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 P(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 P(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 P(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку число равное -P, добавить его в начало строки. Последнее целое число =числу P - 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 P(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку число равное -P, добавить его после числа P 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 P(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 P(если есть, то должно совпадать с лексемой) , вывести позицию в изначальной строке. Добавить в строку квадрат числа P , поместить это число после первой лексемы строки. Последенее число 2-й с\с - удалить из строки. Все результаты сформировать в строки и с помощью String.format и вывести.

Лабораторная № 5.a
Сдать (прислать ссылку)

Общее задание:

1. Создать репозитарий на GitHub
2. Сохранить все предыдущие лабораторные
3. Добавить coommits

Лабораторная № 5.b
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.
3. Использовать классы (чтение\запись в файл) : FileReader, FileWriter, BufferedReader, BufferedWriter.
4. Можно использовать коллекции (класс ArrayList).

1. Составить регулярное выражение, является ли заданная строчка IP адресом, записанным в десятичном виде. Max число -255.
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 экпоненциальном 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, кроме Минска и 5 городов- областных центров. 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
BM 3523213 Октябрьский РУВД Витебск
HB3523213 Советский РУВД Гомель
KH3523213 Московский РУВД Гродно
MP3523213 Московский РУВД Минск
MC3523213 РУВД Слуцк
KB3523213 Ленинский РУВД Могилев
PP8908213 Фрунзенский РУВД Минск

Пример неправильных выражений:
AB123456 Московский РУВД Брест
BM1234567 Октябрьский РУВД
HB12345678 Советский Гомель
KH1234567 Московский РУВД
MM1234567 Московский РУВД Минск
MC1234567 РУВД
KB1234567 РУВД Могилев
PP1234567 Минск
BM1234567 РУВД Гомель
K1234567 Московский РУВД Гродно
МФ1234567 Московский РУВД Минск

6. Написать регулярное выражение определяющее является ли данная строчка датой в формате dd/mm/yyyy - dd mmm 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 соответсвующем 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? с необязательным слешем в конце. 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. Специальные символы не используются. 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/

http://zcontest.com/
https://zcontest.ru
https://sub.zcontest-ru.com:8080
http://zcontest.ru/dir%201/dir_2/
zcon.com/index.html#bookmark
Пример неправильных выражений:
Just Text.
http://a.com
http://www.domain-.com

Лабораторная работа № 6а

Сдать до 20.11

Обработка коллекций.XML.JSON.UNIT-тестирование.

Общее задание:

- *Чтение данных из текстового файла (необязательно, только для изначальной записи в map)*
- *Использовать итераторы для вывода.*
- *Чтение из XML- файла, запись в XML файл.*
- *Чтение из JSON- файла, запись в JSON файл.*
- *Использовать лямбда-выражения, например, для вывода* List.
- *Использовать UNIT-тесты для тестирования нескольких методов*
- *Запись кода и кода тестов на GitHub*

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 ввывод 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 коэфициента, 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 работа № 6b

Сдать до 27.11

Шифрование. Архивация .

Общее задание:

- *Для лабораторной 6 реализовать:*
    1. *Шифрование данных.*
    2. *Архивация (jar, zip, rar).*
3. *Запись кода и кода тестов на GitHub*

Лабораторная работа № 7

UNIT-тесты

Общее задание:

- *Использовать UNIT-тесты для тестирования лабораторных1-5.*
- *Запись кода и кода тестов на GitHub*

Лабораторная работа № 8

Сдать до 18.12

Паттерны

Общее задание:

- *Реализовать два варианта с Design Patterns (Decorator и Builder).*
- *Реализовать Pattern Singleton для доступа к базе данных (файлу данных)*
- *Использовать UNIT-тесты для тестирования.*
- *Запись кода и кода тестов 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 стиральных машин.
5. Кафе, разработка индивидуального меню.
6. Фабрика по сборке различных домов.
7. Фабрика по сборке различных домашних комбайнов.
8. Фабрика по пошиву различной одежды.
9. Фабрика по изготовлению мебельных гарнитуров.
10. Кафе, изготовление различных тортов.

Лабораторная работа № 9*(бонусная)

Графика.

Общее задание:

- *Создать графическое приложение для одной из лабораторных: 4,5, 6, 8.*
- *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*