Книги и материалы для подготовки:

Java Complete Reference by Schildt;

Java SE 8 for the Really Impatient by Horstmann;

https://docs.oracle.com/javase/tutorial/java/index.html;

1. Темы
   1. Синтаксические конструкции schildt I:3,4,5
   2. ООП. Наследование schildt I:6,7,8
   3. Collections. Array, List, Map horst 160-162
   4. Strings schildt II:16
   5. I/O. Файловая система. schildt 689-727; horst 163-166, 183-187
   6. Streams API. schildt 965-991; horst 2
   7. Lambdas. schildt 381-412; horts 1

Задания:

**Lang.math**:

* + - 1. Euclid’s algorithm (which is over two thousand years old) computes the greatest common divisor of two numbers as gcd(a, b) = a if b is zero, and gcd(b, rem(a, b)) otherwise, where rem is the remainder. Clearly, the gcd should not be negative, even if a or b are (since its negation would then be a greater divisor). Implement the algorithm with %, floorMod, and a rem function that produces the mathematical (non-negative) remainder. Which of the three gives you the least hassle with negative values? [horst p.174 #3]

**1.** Алгоритм Евклида (которому более двух тысяч лет) вычисляет наибольший общий делитель двух чисел как gcd (a, b) = a, если b равно нулю, а gcd (b, rem (a, b)) в противном случае, где rem - остаток. Ясно, что gcd не должен быть отрицательным, даже если a или b (поскольку его отрицание тогда будет большим делителем). Реализуйте алгоритм с помощью функции **%**, **floorMod** и функции **rem**, которая производит математический (неотрицательный) остаток. Какая из трех дает вам наименьшую проблему с отрицательными значениями? [horst с.174 # 3]

**Util.collection**

**2.** Implement a bidirectional Rot13 algorithm. Encode/decode a string from the command line argument.

**3.** Write a program to convert an input hexadecimal string into its equivalent decimal number.

1. Внедрить двунаправленный алгоритм **Rot13**. Кодирование / декодирование строки из аргумента командной строки.
2. Напишите программу для преобразования входной шестнадцатеричной строки в ее эквивалентное десятичное число.

**Java.nio**

**4.** Write methods that turn a Scanner into a stream of words, lines, integers, or double values. [horst.p 175 #9]

**5.** Unzip the src.zip file from the JDK. Using Files.walk, find all Java files that contain the keywords transient and volatile; [horst.p 175 #10]

**6.** Write an example that counts the number of times a particular character, such as E, appears in a file. The character can be specified at the command line.

**7.** Write a program that reads a text file, specified by the first command line argument, into a List. The program should then print random lines from the file, the number of lines printed to be specified by the second command line argument. Write the program so that a correctly-sized collection is allocated all at once, instead of being gradually expanded as the file is read in. Hint: To determine the number of lines in the file, use [java.io.File.length](https://docs.oracle.com/javase/8/docs/api/java/io/File.html#length--) to obtain the size of the file, then divide by an assumed size of an average line.

**4.** Запишите методы, которые превращают сканер в поток слов, строк, целых чисел или чисел double. [horst.p 175 # 9]

**5**. Разархивируйте файл src.zip из JDK. Используя Files.walk, найдите все файлы Java, содержащие ключевые слова *transient* (переходные) и *volatile* (нестабильные); [horst.p 175 # 10]

**6.** Напишите пример, который подсчитывает количество раз, когда в файле появляется конкретный символ, например E. Символ может быть указан в командной строке.

**7.** Напишите программу, которая читает текстовый файл, указанный в первом аргументе командной строки, в список. Затем программа должна распечатать случайные строки из файла, количество строк, напечатанных для указания вторым аргументом командной строки. Запишите программу так, чтобы коллекция с правильным размером была распределена одновременно, а не постепенно расширялась по мере чтения файла. Совет. Чтобы определить количество строк в файле, используйте java.io.File.length для получения размер файла, затем разделить на предполагаемый размер средней строки.

**Util.stream**

**8.** Using Stream.iterate, make an infinite stream of random numbers—not by calling Math.random but by directly implementing a linear congruential generator. In such a generator, you start with x0 = seed and then produce xn + 1 = (a xn + c) % m, for appropriate values of a, c, and m. You should implement a method with parameters a, c, m, and seed that yields a Stream. Try out a = 25214903917, c = 11, and m = 2^48. [horst p. 44 #5]

**9.** Write a method public static Stream zip(Stream first, Stream second) that alternates elements from the streams first and second, stopping when one of them runs out of elements. [horst p. 44 #8]

**8.** Используя Stream.iterate, создайте бесконечный поток случайных чисел - не путем вызова Math.random, а путем непосредственного внедрения линейного конгруэнтного генератора. В таком генераторе вы начинаете с x0 = seed, а затем создаете xn + 1 = (a xn + c)% m для соответствующих значений a, c и m. Вы должны реализовать метод с параметрами a, c, m и seed, который дает Stream. Попробуйте a = 25214903917, c = 11 и m = 248. [horst p. 44 # 5]

**9**. Напишите метод public static Stream zip (first Stream, Stream second), который чередует элементы из потоков сначала и вторые, останавливаясь, когда один из них заканчивается из элементов. [horst p. 44 # 8]

**Algo**

(**Optional**) 10. Online palindrome detection. Read in characters one at a time. Report at each instant if the current string is a palindrome. Use the Rabin-Karp hashing algorithm. <http://www-igm.univ-mlv.fr/~lecroq/string/node5.html#SECTION0050>

<http://algs4.cs.princeton.edu/53substring/>

Зачет задач howto:

1. Создаем git-репозиторий на открытой площадке (GitHub/Bitbucket).
2. Делаем задачи, комитим код с комментариями в репозиторий.
3. Передаем ссылку на репо мне.

Почта: [yuriitipikin@gmail.com](mailto:yuriitipikin@gmail.com)