OOP with Java
Course project 9
Investigation of Gene sequences

Acceptable Programming Languages:	**Java**
Deadline:	**Февруари *(on the final exam date)***
Instructor	**Dr. Evgeny Krustev**

Problem Statement:

It is not uncommon nowadays, especially with the large number of genomes being sequenced, that a researcher comes across a novel DNA or protein sequence for which no functional data is available. Some basic information on the sequence is necessary before a molecular biologist can take the new sequence into the laboratory and perform meaningful experiments with it. It would, for example, make the task of deciphering the biological function of a piece of DNA much easier if it were known that the new sequence encoded a metabolic enzyme or, indeed, a protein that is a putative member of a superfamily such as an immunoglobulin, a kinase, etc. Conversely, if the sequence was a Repetitive DNA Element, it would need an entirely different approach for its study

This is where the power of database searching comes in handy. The principle aim of database searching, is to reveal the existence of similarity between an input sequence (called 'query sequence') that a user wants to find more information about and other sequences (called 'target' sequences) that are stored in a biological database. This is usually the first step a researcher takes in determining the biological significance of an unknown sequence.

The purpose of this project is compute several basic properties of a given sequence.

One such property is the type of the sequence and you have to develop a software application that determines if a given sequence is nucleotide (DNA or RNA) or protein. Note that RNA, like DNA is a polymer composed of four nucleotides. The difference between RNA and DNA is the nature of the sugar moiety: RNA has the ribose sugar, while DNA has the deoxyribose sugar. RNA has the same purine bases as DNA: adenine (A) and guanine (G) and the same pyrimidine cytosine (C), but instead of thymine (T), it uses the pyrimidine uracil (U).

Determination of sequence type is done with an algorithm that takes into account information on the natural composition of nucleotide and protein sequences. According to the algorithm, if:

1. Total number of nuculeotides (that is, sum of A, T, G and C's) divided by the total length of the sequence is greater that 0.85, **it is a DNA sequence**
2. Total number of A, T, G, C and U's divided by the total length of the sequence is greater that 0.85, **it is an RNA sequence**

If neither of these two conditions is met, the sequence is assumed to be a protein sequence. Note that we are not using the extended DNA/RNA alphabet that includes symbols for sequence ambiguity as defined in the International Union of Pure and Applied Chemistry (IUPAC)

and International Union of Biochemistry (IUB) nucleotide and amino acid nomenclature. Instead, we are assuming the DNA alphabet to be composed of the four bases A (adenine), T (thymine), G (guanine), C (cytosine) and N, the RNA alphabet to be composed of A (adenine), U (uridine), G (guanine), C (cytosine) and N (where N is any nucleotide base) and the amino acid alphabet to be composed of A (alanine), C (cysteine), D (aspartate), E (glutamic acid), F (phenylalanine), G (glycine), H (histidine), I (isoleucine), K (lysine), L (leucine), M (methionine), N (asparagine), P (proline), Q (glutamine), R (arginine), S (serine), T (threonine), V (valine), W (tryptophan) and Y (tyrosine).

Project tasks:

A. Theory

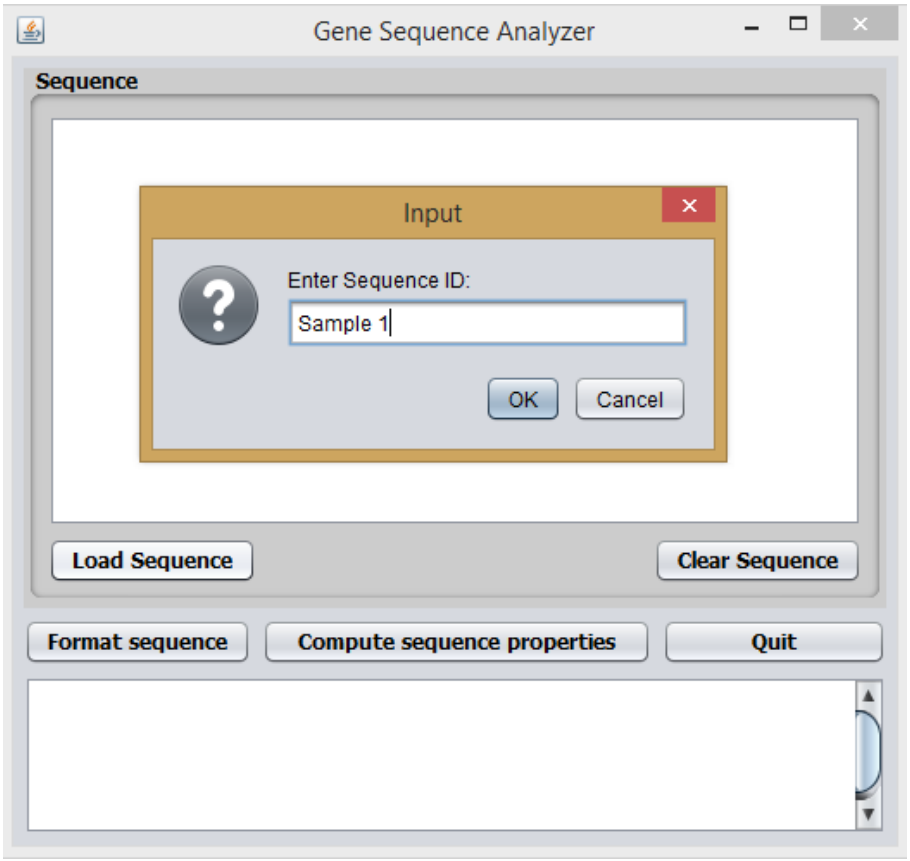
1. Research Internet resources and create a list of at least four biological sequence databases. Prepare a short report including information about:
   a) The location of the database
   b) The services provided
   Provide examples about how database searches have helped making discoveries in biology and medicine.
   Submit a MS Word document with your findings

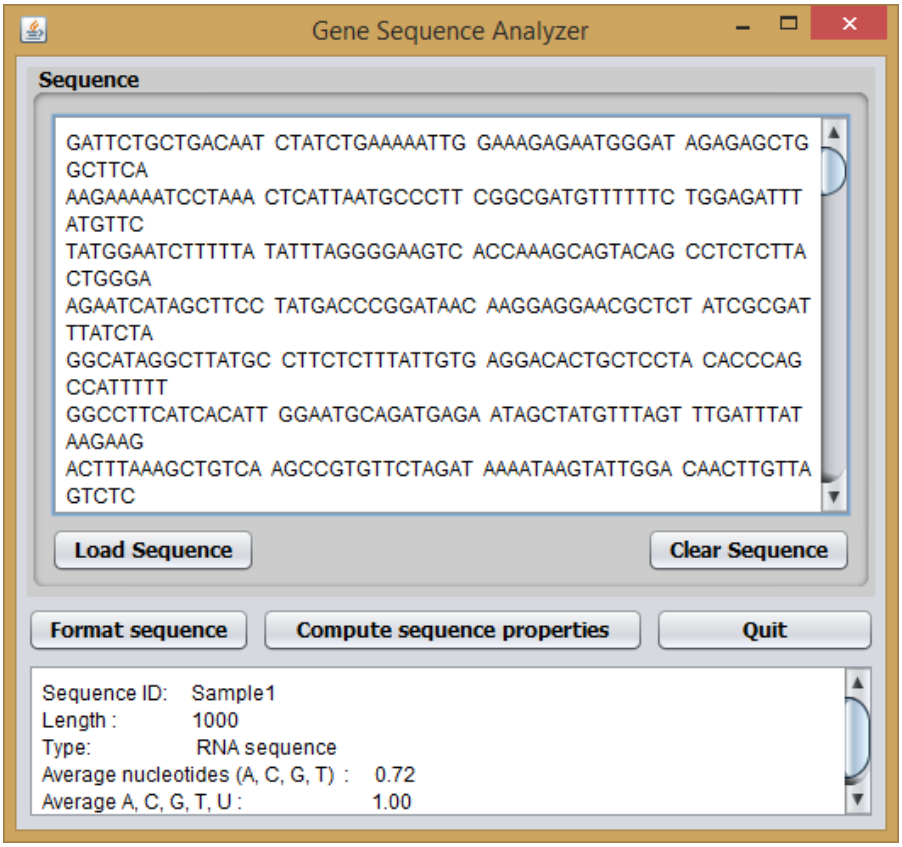
B. Practice

1. Write a SOAP web service application with a web method that gets as an argument the name a Gene sequence sample, reads the appropriate text file for this sample (sample gene sequences are provided with this project) and returns the String with the gene sequence read from the appropriate text file
2. Write a JavaFX client application for the SOAP web service with the following design

a) when the user clicks **Load sequence** button , the selected gene sequence is read from the web service method and loaded in the Sequence text area (use a Task<Void>)
b) when the user clicks **Clear sequence** button , Sequence text area gets cleared
c) when the user clicks **Format sequence** button, remove all the spaces and the new line characters inside the text displayed in the Sequence text area and all the letters change to lower case (use **regular expressions and the Stream API**)
d) when the user clicks **Compute sequence properties** button, the following properties of the loaded gene sequence get computed (match the given sample):
   - The Sequence ID name
   - The length of the sequence (the total characters of the DNA alphabet found in the given sequence)
   - The average of the total occurrence of A, T, G and C's in the gene sequence
   - The longest sequences of A's, T's, G's and C's in the gene sequence and how many times each one of these sequences appear in the gene sequence.
   - The average of the total occurrence of A, T, G, C and U's in the gene sequence
   - The type of the sequence (DNA, RNA or protein) as per the algorithm described above

3. Research the Internet and test the application with gene sequence of your own. Try for instance, http://www.genet.sickkids.on.ca/GenomicDnaSequencePage,form.direct
4. **Submit** in a single archive the application (**Netbeans project**) and a **short report** of the work complete to execute the above Practice tasks


Gene Sequence Analyzer
Sequence
Input
Enter Sequence ID:
Sample 1
OK
Cancel
Load Sequence
Clear Sequence
Format sequence
Compute sequence properties
Quit


Gene Sequence Analyzer
Sequence
GATTCTGCTGACAAT CTATCTGAAAAATTG GAAAGAGAATGGGAT AGAGAGCTG
GCTTCA
AAGAAAAATCCTAAA CTCATTAATGCCCTT CGGCGATGTTTTTTC TGGAGATTT
ATGTTC
TATGGAATCTTTTTA TATTTAGGGGAAGTC ACCAAAGCAGTACAG CCTCTCTTA
CTGGGA
AGAATCATAGCTTCC TATGACCCGGATAAC AAGGAGGAACGCTCT ATCGCGAT
TTATCTA
GGCATAGGCTTATGC CTTCTCTTTATTGTG AGGACACTGCTCCTA CACCCAG
CCATTTTT
GGCCTTCATCACATT GGAATGCAGATGAGA ATAGCTATGTTTAGT TTGATTTAT
AAGAAG
ACTTTAAAGCTGTCA AGCCGTGTTCTAGAT AAAATAAGTATTGGA CAACTTGTTA
GTCTC
Load Sequence
Clear Sequence
Format sequence
Compute sequence properties
Quit
Sequence ID: Sample1
Length : 1000
Type: RNA sequence
Average nucleotides (A, C, G, T) : 0.72
Average A, C, G, T, U : 1.00

Evaluation:

Your project will be evaluated on the following **general points**:

- **Sophistication/complexity/originality** of the problem being solved/investigated and of the **solution(s)/approaches** considered.
- **Demonstrated ability to extract/analyze** concurrency-related problems/issues from a general problem/area of interest.
- **Clarity of explanations, and for implementations programming skill/quality**. Your report **(in Bulgarian!)** should be well written and free of grammatical and spelling errors. **Programs** must be **well commented and in a professional style**.
- **Awareness of related work.** Others have considered the same or similar problems before you. Your work does not have to be novel, but you should be able to contextualize your approach. Be sure to explain how each referenced work is *related* to your work. Note that a 5-minute *Google* search will not be adequate; if you are unfamiliar with the required textbooks for the course:
- **Completeness** of the project.

Deliverables: *The files with* :

1. the source code
2. the executable code
3. the instructions for compiling your source code
4. the report explaining the data structures and the algorithm implementation, describe things such as how your code has been tested, limitations of your code, problems encountered, and problems remaining
5. any files used to test the implementation of the program with an explanation about it included in the report.

References:

[1] **P. Deitel, H. Deitel, "**Java How to Program (early objects)**",** Prentice Hall 9 ed. **2012**, ISBN-10: 0132575663, ISBN-13: 978-0-13-257566-9 (**основна)**

[2] **Y. Daniel Liang, "**Introduction to Java Programming**", 7th** ed.,Prentice Hall **2009** ISBN-13: 978-0-13-601267-2

[3] **Bruce Eckel** "Thinking in Java", 4th ed., Prentice Hall 2006 или българското й издание "Да мислим на Java" том 1 и 2, SoftPress, **2001**