Objectives

To write a Java program that manipulates strings of DNA by appropriately using methods from the String and Random classes. To gain exposure to the field of Bioinformatics that lies at the intersection of Biology, Math and Computer Science. To continue the practice of using Scanner class to obtain user's input through the terminal.

Special Notes

This is your first team-based assignment. You have to work in groups of two (or three) for this assignment. You may select your own partner.

This is a longer laboratory assignment than your previous assignments. Please plan your time this week accordingly and work on it incrementally. Remember, divide and conquer!

General Guidelines for Labs

- **Work on the Alden Hall computers.** If you want to work on a different machine, be sure to transfer your programs to the Alden machines and re-run them before submitting.

- **Update your repository often!** You should add, commit, and push your updated files each time you work on them. I will not grade your programs until the due date has passed.

- **Review the Honor Code policy.** You may discuss programs with others, but programs that are nearly identical to others will be taken as evidence of violating the Honor Code.

Reading Assignment

To learn more about Java strings and random numbers, review Sections 3.1–3.4 in your textbook. Also carefully study the sample program described in the section called A Sample Program in this lab assignment document.

Create a new directory and a Java program

In your own cs111f2016-<your user name> repository inside labs directory, create a directory called lab5 by typing “mkdir lab5”. Type “cd lab5” to move to the new directory. Then using gvim create a Java program named DnaManipulation.java. See the section called DNA Manipulation in this document for the specifics on what you need to do in your DnaManipulation.java program.
Study a Sample Program

Go to the shared course repository and pull StringDemo.java. Copy this program into lab5 directory inside your own cs111f2016-<your user name> repository. Open this program and examine it to see what it does. Then run it a few times with different input strings.

Discuss the Following Questions with Your Group Member:

At this point, you should create a report text document and type your responses to the questions below in your report, after you discuss them with your group member.

- Where is the random number generator declared?
- Suppose you name the random number generator wilbur instead of r. What other lines in the program need to be changed?
- Where is the scanner declared?
- Suppose you name the scanner orville instead of scan. What other lines in the program need to be changed?
- Suppose the line:
  \[
  \text{s1 = s1.toUpperCase();}
  \]
  is changed to:
  \[
  \text{s1.toUpperCase();}
  \]
  Will the program still correctly display the upper-case version of s1?
- Do the statements s1.toUpperCase(); and s1.toLowerCase(); change the contents of s1? (See previous question!)
- Suppose the user types abcde when asked to enter a string.
  - Write this down with numbers to indicate the positions (indices) of each character.
  - How many different ways are there to insert an “x” into the string abcde? List all the position numbers where this x could be placed. (Don’t forget the beginning and the end.)
  - According to the book, thenextInt(num) method of the Random class returns a random number in the range 0 to num - 1. List all the values that could be returned by “r.nextInt(5)”.
- On paper, write down the string ABCDEFG, with position numbers between the letters.
  - What is the portion corresponding to the expression:
    \[
    "ABCDEFG".substring(0,3)
    \]
– What is the portion corresponding to the expression:

"ABCDEFG".substring(3)

– What string do we get if we evaluate the expression:

"ABCDEFG".substring(0,3) + "ABCDEFG".substring(3)

Explain, in English words, what the following statement does:

\[ s2 = s1.substring(0,\text{location}) + 'x' + s1.substring(\text{location}); \]

• What is the value of the expression "PQRST".charAt(0) ?

• What is the value of the expression "PQRST".charAt(2) ?

• What is the value of the expression "PQRST".charAt(4) ?

• List all possible values that can be returned by the expression \( r.nextInt(5) \)

• Explain, in English words, what the following statement does:

\[ c = "PQRST".charAt(r.nextInt(5)); \]

Don’t continue with the assignment until you understand the answers to all the questions above! Please feel free to discuss these with your classmates, teaching assistants and your instructor!

**DNA Manipulation Program**

Bioinformatics is the study of biological phenomena by the use of biology, mathematics and computer science. The field of bioinformatics studies and develops solutions for critical problems such as disease prediction, tracking infectious diseases, forensic science, drug discovery, improving food supply and personalizing medicine. One of the most important data used in the bioinformatics studies concerns DNA. Deoxyribonucleic acid (DNA) is a molecule that encodes the genetic instructions (genes) which are used by all known living organisms and many viruses to build the proteins required to sustain existence. The genes of DNA are written in the nucleotides; guanine (G), adenine (A), thymine (T), and cytosine (C), (chemical compounds) which serve as the alphabet of the genetic language. Essentially, a DNA string is a string consisting of only the letters A, C, G, and T, for instance, “CAATGTCAC”. These strings encode various genetic traits such as hair color, eye color, and many others.

Each DNA string has a complement formed by replacing each code letter by its complementary code. A and T are complements; so are G and C. Thus, the complement to the string “CAATGTCAC” is “GTTACAGTG”. In many cases, DNA needs to be manipulated in order to achieve the desired task. That is, in order to enhance or modify the characteristics of an individual organism, DNA of that organism maybe altered in some way. For example, DNA manipulation is used to produce a type of insulin from yeast and bacteria that can be used on diabetic patients. DNA may sometimes undergo a mutation, or a self-imposed manipulation as well (e.g., inherited from a parent). There are three types of mutation/manipulation that may occur: insertion of a new letter somewhere in the string; removal of a letter from the string; and replacement of one letter by another. The table below shows the examples of the replacement of letters and the complement of the given sequence.
Strand | Sequence
---|---
$S$ | ACGTGCTCTGGTAC
$A \rightarrow T$ | TCGTGCTCTGGTTC
$T \rightarrow A$ | TCGAGCCGCAAAGATC
$C \rightarrow G$ | TGGGGGAGAGAGATG
$G \rightarrow C$ | TGCACGGGAAACCACG
$S_{complementary}$ | TGCACGGGAAACCACG

For this laboratory assignment, you will create a Java program that manipulates a DNA string by finding its complement and performing different types of mutation. Note that all changes are made to the original input string; they are not cumulative. Your program should satisfy the following requirements:

1. Get the DNA string as an input from the user, and save it into a variable called “dnaString” (or another appropriate variable name).

2. Obtain the complement of dnaString, appropriately labeled. (HINT: you can use several applications of the String class's “replace” method. But remember, that replace method will replace all instances of the specified character. How can you circumvent this?)

NOTE: The next three parts of your program will also use Java's Random class. You may want to read about the Random class at the bottom of this document to review the concepts behind the Random class.

3. Perform a random mutation consisting of inserting a randomly-chosen extra letter into a randomly-chosen position in a dnaString; it must be one of the four allowed letters. Print this, appropriately labeled and identifying the position of the insertion and the letter inserted.

4. Perform a random mutation consisting of removing a randomly-chosen letter from a randomly-chosen position in a dnaString; print this, appropriately labeled and identifying the position of the insertion and the letter removed.

5. Perform a random mutation consisting of altering a single letter from a randomly-chosen position in a dnaString; it must be changed to a randomly-chosen letter from the set of allowed letters. Print it, appropriately labeled and identifying the position of the replacement, the new letter, and the letter it replaces.

Two sample runs are provided below. You will get different values, of course, since the changes are random, but the structure of the output will be the same:

```
aldenv5:lab5 jjumadinova$ java DnaManipulation
Janyl Jumadinova
Lab 5
Thu Sep 28 12:51:58 EDT 2016

Enter a string containing only C, G, T, and A: actg
Complement of ACTG is TGAC
```
Inserting T at position 0 gives TACTG
Deleting from position 1 gives ATG
Changing position 2 gives ACGG

aldenv5:lab5 jjumadinova$ java DnaManipulation
Janyl Jumadinova
Lab 5
Thu Sep 28 12:52:58 EDT 2016

Enter a string containing only C, G, T, and A: actg
Complement of ACTG is TGAC
Inserting G at position 0 gives GACTG
Deleting from position 0 gives CTG
Changing position 0 gives ACTG

In the second example, nothing was changed in the last line—the program randomly replace the
letter “A” with the letter “A”! This behavior is ok.

Additional Program Requirements

• Make sure your program prints the names of all team members, the lab number, and the date
  as the first few output lines.

• Make sure your program contains the comment header with the honor pledge, the names of
  all team members, lab number, date, and the purpose of the program.

• Make sure you document your program properly, by using comments throughout your program
  whenever appropriate.

• Make sure your output is neat (no missing spaces, etc.) and that your program is neat
  (indenting, etc.).

Required Deliverables

For this assignment you are invited to submit electronic versions of the following deliverables
through the Bitbucket repository. As you complete this step, you should make sure that you created
a lab5/ directory within the Git repository. Then, you can save all of the required deliverables in
the lab5/ directory—please see the course instructor or a teaching assistant if you are not able to
create your directory properly.

1. A completed, properly commented and formatted DnaManipulation.java program. Only
   one submission from a team is required for this document.

2. A report document containing: (1) your answers to the questions in the first part of the
   lab, (2) at least three different outputs obtained after running DnaManipulation in the
   terminal window at least three times, (3) description of the strategy your team developed for
   completing this assignment. Only one submission from a team is required for this document.
3. A reflection document explaining the work each team member has completed and if there were any challenges you faced while working with your team member. A submission from each team member is required.

Share your program, the output file and the document describing your team work with me through your Git repository by correctly using “git add”, “git commit”, and “git push” commands. When you are done, please ensure that the Bitbucket Web site has a lab5/ directory in your repository with the three files called DnaManipulation.java, output and report. You should see the instructor if you have questions about assignment submission.

A Quick Tutorial on Random Number Generation

To generate random numbers, we need an object of the Random class. You can name this variable anything you want—“rand” or “random” are good names. To create a new random number generator named rand, be sure to import the Random class: import java.util.Random;

and then create an instance of this class: Random rand = new Random();

The three most useful methods in the Random class are nextInt, nextFloat and nextDouble. If rand is the name of our random number generator (it can be called something else) and n is a positive integer, rand.nextInt(n) produces an int in the range 0, ..., n-1 and rand.nextDouble() and rand.nextFloat() produce a double and a float respectively in the range from 0 to 1 (not including 1).

By being clever, we can get different ranges—here are a few examples. In the Java code, assume i is an int variable, d is a double variable, and rand is an object of the Random class:

<table>
<thead>
<tr>
<th>Desired Range</th>
<th>Java Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1, 2, 3, 4, 5</td>
<td>i = rand.nextInt(6);</td>
</tr>
<tr>
<td>10, 11, ..., 19, 20</td>
<td>i = rand.nextInt(11) + 10;</td>
</tr>
<tr>
<td>-5, -4, -3, ..., 4, 5</td>
<td>i = rand.nextInt(11) - 5;</td>
</tr>
<tr>
<td>0, 3, 6, 9, 12</td>
<td>i = 3 * rand.nextInt(5);</td>
</tr>
<tr>
<td>-1, 1</td>
<td>i = 2 * rand.nextInt(2) - 1;</td>
</tr>
<tr>
<td>0 ≤ d &lt; 1</td>
<td>d = rand.nextDouble();</td>
</tr>
<tr>
<td>0 ≤ d &lt; 10</td>
<td>d = 10 * rand.nextDouble();</td>
</tr>
<tr>
<td>−5 ≤ d &lt; 5</td>
<td>d = 10 * rand.nextDouble() - 5;</td>
</tr>
</tbody>
</table>

You can also use random numbers to do other creative things, for example pick out characters randomly. Let’s say I want to pick out a random letter out of my first name, I can first create and initialize a variable String myName = "Janyl"; Then I can generate a random number between 0 and 4 and save it into a variable of type int as int randomNumber = rand.nextInt(5). Finally, I use charAt method to select the character at a position created by the random number generator as myName.charAt(randomNumber).