

Сертифікат: <https://coursera.org/account/accomplishments/verify/BL7W6PJN4VU2>

Практичні завдання: https://github.com/Akhanrok/java_coursera

Середовище розробки: <https://www.dukelearntoprogram.com//downloads/bluej.php>

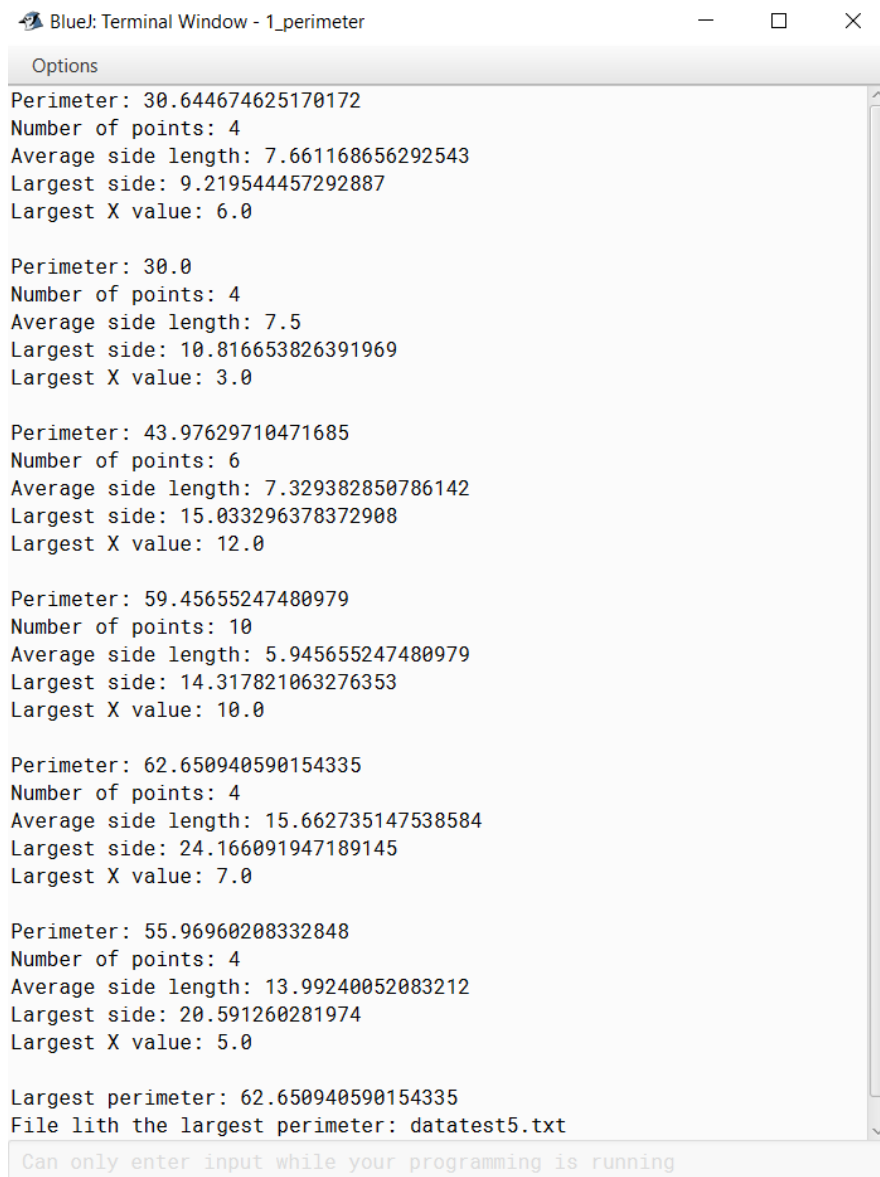
Курс: <https://www.coursera.org/learn/java-programming>

Завдання 1

Створити програму, що обчислює периметр фігури, середню довжину сторони, довжину найбільшої сторони та виводить найбільше значення X, кількість точок фігури в обраному користувачем файлі та назву файлу з найбільшим периметром серед усіх обраних файлів і його значення.

Код програми: [PerimeterAssignmentRunner.java](#)

Результати виконання програми:



```
BlueJ: Terminal Window - 1_perimeter
Options
Perimeter: 30.644674625170172
Number of points: 4
Average side length: 7.661168656292543
Largest side: 9.219544457292887
Largest X value: 6.0

Perimeter: 30.0
Number of points: 4
Average side length: 7.5
Largest side: 10.816653826391969
Largest X value: 3.0

Perimeter: 43.97629710471685
Number of points: 6
Average side length: 7.329382850786142
Largest side: 15.033296378372908
Largest X value: 12.0

Perimeter: 59.45655247480979
Number of points: 10
Average side length: 5.945655247480979
Largest side: 14.317821063276353
Largest X value: 10.0

Perimeter: 62.650940590154335
Number of points: 4
Average side length: 15.662735147538584
Largest side: 24.166091947189145
Largest X value: 7.0

Perimeter: 55.96960208332848
Number of points: 4
Average side length: 13.99240052083212
Largest side: 20.591260281974
Largest X value: 5.0

Largest perimeter: 62.650940590154335
File with the largest perimeter: datatest5.txt

Can only enter input while your programming is running
```

Завдання 2

Створити метод `findSimpleGene`, що виконує наступне:

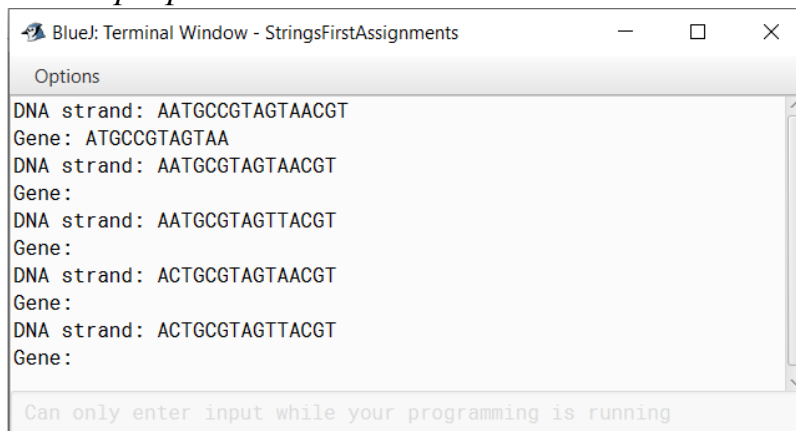
- знаходить позицію індексу стартового кодону «ATG». Якщо «ATG» немає, повертає порожній рядок;
- знаходить позицію індексу першого стоп-кодону «TAA», що з'являється після знайденого «ATG». Якщо «TAA» немає, повертає порожній рядок;
- якщо довжина рядка між «ATG» і «TAA» кратна 3, програма виводить цей рядок, включаючи кодони «ATG» і «TAA».

Створити метод `testSimpleGene`:

- створити п'ять ланцюгів ДНК:
 - ДНК без «ATG»;
 - ДНК без «TAA»;
 - ДНК без «ATG» і «TAA»;
 - ланцюг ДНК з «ATG», «TAA», що кратний 3;
 - ланцюг ДНК з «ATG», «TAA», що не кратний 3;
- вивести кожен ланцюг ДНК, перевірити, чи є ген, викликавши `findSimpleGene` з цим рядком як параметром. Якщо ген існує, вивести ген, інакше – порожній рядок.

Код програми: [Part1.java](#)

Результати виконання програми:



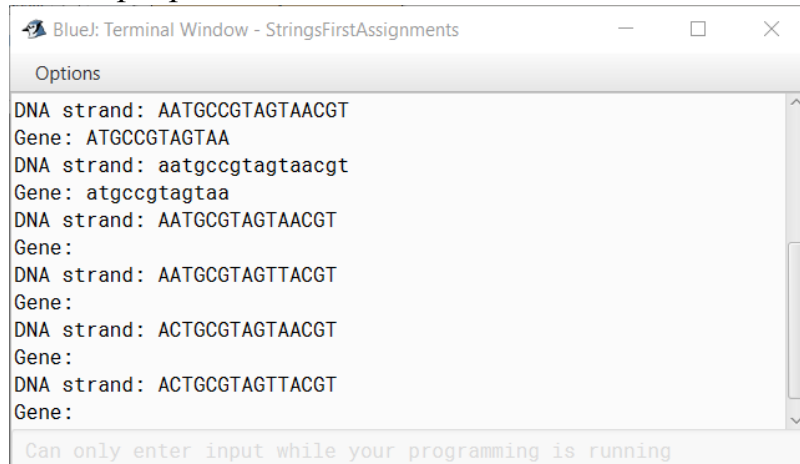
```
BlueJ: Terminal Window - StringsFirstAssignments
Options
DNA strand: AATGCCGTAGTAACGT
Gene: ATGCCGTAGTAA
DNA strand: AATGCGTAGTAACGT
Gene:
DNA strand: AATGCGTAGTTACGT
Gene:
DNA strand: ACTGCGTAGTAACGT
Gene:
DNA strand: ACTGCGTAGTTACGT
Gene:
Can only enter input while your programming is running
```

Завдання 3

Змінити метод `findSimpleGene` з попереднього завдання, додавши три параметри: для рядка ДНК, для стартового кодону та для стоп-кодону. Додати до методу `testSimpleGene` ланцюг ДНК що складається з малих літер. Ген має виводитися в тому ж регістрі, що й ланцюг ДНК.

Код програми: [Part2.java](#)

Результати виконання програми:



```
BlueJ: Terminal Window - StringsFirstAssignments
Options
DNA strand: AATGCCGTAGTAACGT
Gene: ATGCCGTAGTAA
DNA strand: aatgccgtagtaacgt
Gene: atgccgtagtaa
DNA strand: AATGCCGTAGTAACGT
Gene:
DNA strand: AATGCCGTAGTTACGT
Gene:
DNA strand: ACTGCCGTAGTAACGT
Gene:
DNA strand: ACTGCCGTAGTTACGT
Gene:
Can only enter input while your programming is running
```

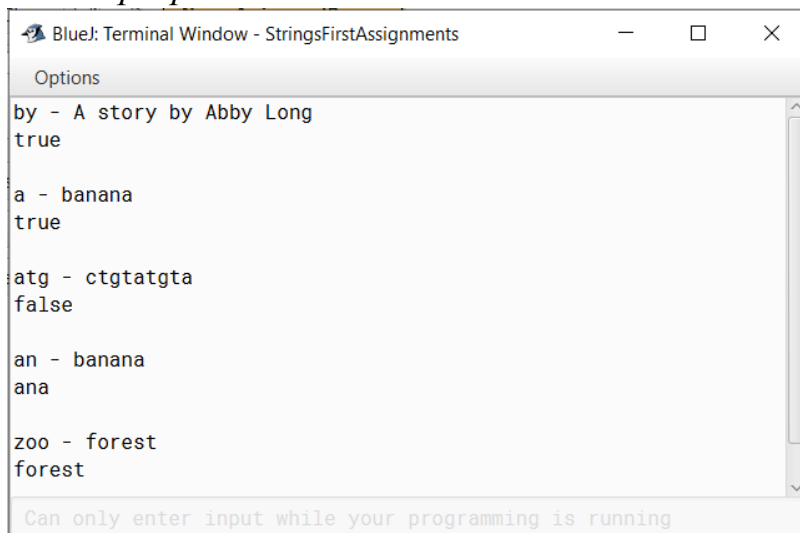
Завдання 4

Створити метод `twoOccurrences`, що має два параметри `String` під назвою `stringa` та `stringb`. Цей метод повертає `true`, якщо `stringa` з'являється принаймні двічі в `stringb`, інакше він повертає `false`.

Створити метод `lastPart`, який має два параметри `String` під назвою `stringa` та `stringb`. Цей метод знаходить першу появу `stringa` в `stringb` і повертає частину `stringb`, яка йде за `stringa`. Якщо `stringa` не зустрічається в `stringb`, повертається `stringb`.

Код програми: [Part3.java](#)

Результати виконання програми:



```
BlueJ: Terminal Window - StringsFirstAssignments
Options
by - A story by Abby Long
true

a - banana
true

atg - ctgtatgta
false

an - banana
ana

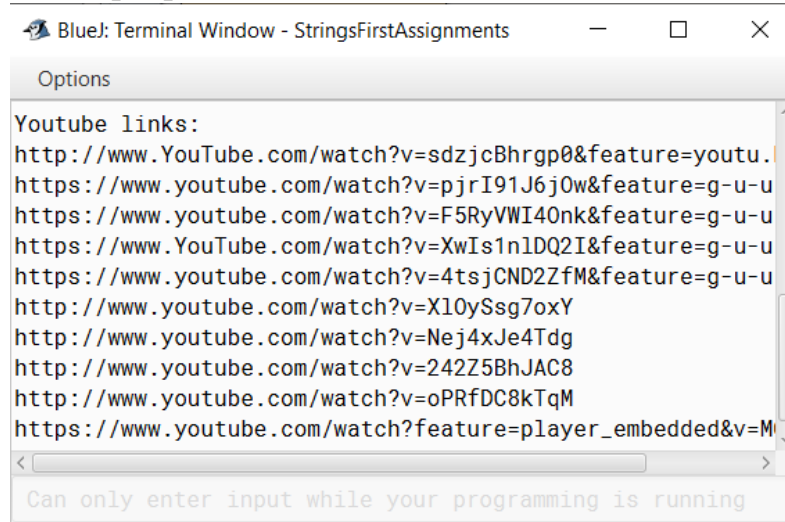
zoo - forest
forest
Can only enter input while your programming is running
```

Завдання 5

Створити програму, яка виводить кожну URL-адресу на сторінці <http://www.dukelearntoprogram.com/course2/data/manylinks.html>, що є посиланням на youtube.com.

Код програми: [Part4.java](#)

Результати виконання програми:



```
BlueJ: Terminal Window - StringsFirstAssignments
Options
Youtube links:
http://www.YouTube.com/watch?v=sdzjcBhrp0&feature=youtu.
https://www.youtube.com/watch?v=pjrI91J6j0w&feature=g-u-u
https://www.youtube.com/watch?v=F5RyVWI40nk&feature=g-u-u
https://www.YouTube.com/watch?v=XwIs1n1DQ2I&feature=g-u-u
https://www.youtube.com/watch?v=4tsjCND2ZfM&feature=g-u-u
http://www.youtube.com/watch?v=X10ySsg7oxY
http://www.youtube.com/watch?v=Nej4xJe4Tdg
http://www.youtube.com/watch?v=242Z5BhJAC8
http://www.youtube.com/watch?v=oPRfDC8kTqM
https://www.youtube.com/watch?feature=player_embedded&v=M
Can only enter input while your programming is running
```

Завдання 6

Створити метод `findStopCodon`, який має три параметри: `String dna`, `int startIndex`, `String stopCodon`. Цей метод повертає індекс першої появи `stopCodon`, який з'являється після `startIndex`, а відстань між ними кратна 3. Якщо такого стоп-кодону немає, цей метод повертає довжину ланцюга ДНК.

Створити метод `findGene`, що виконує наступне:

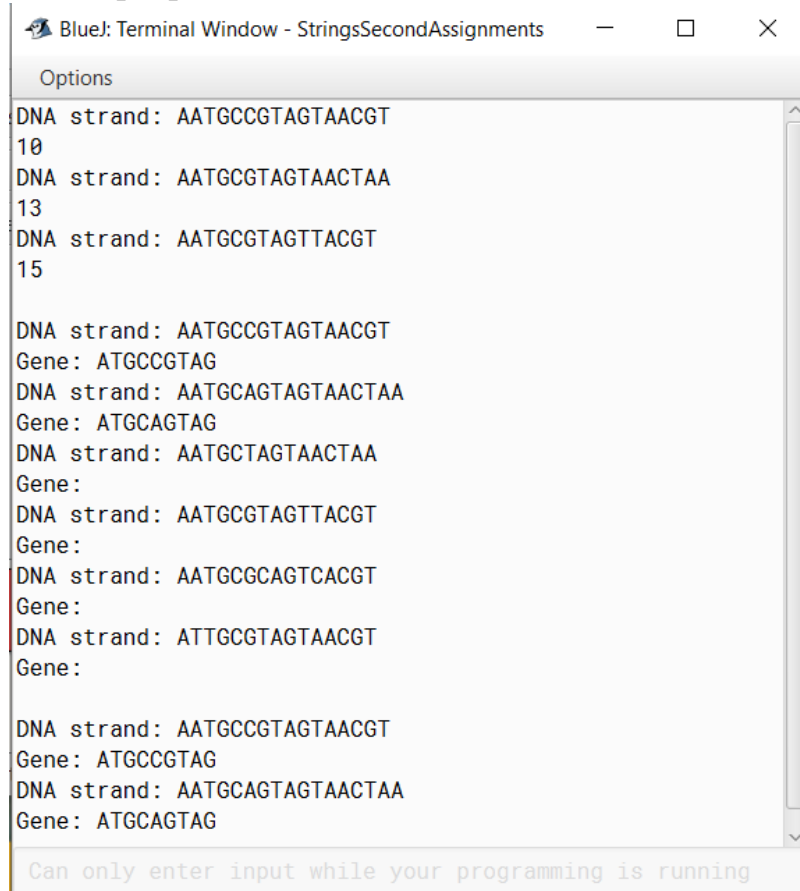
- знаходить індекс першого стартового кодону «ATG». Якщо «ATG» немає, повертає порожній рядок;
- знаходить індекс першого стоп-кодону «TAA»/«TAG»/«TGA» після першого «ATG», який кратний трьом від «ATG».
- повертає ген, утворений з «ATG» і найближчого стоп-кодону, кратного трьом. Якщо немає гена, повернути порожній рядок.

Створити метод `printAllGenes`, який кілька разів знаходить гени та друкує кожен, доки гени не закінчаться.

Створити методи `testFindStopCodon`, `testFindGene`, `testPrintAllGenes`

Код програми: [Part1.java](#)

Результати виконання програми:



```
BlueJ: Terminal Window - StringsSecondAssignments
Options
DNA strand: AATGCCGTAGTAACGT
10
DNA strand: AATGCCGTAGTAACTAA
13
DNA strand: AATGCCGTAGTTACGT
15

DNA strand: AATGCCGTAGTAACGT
Gene: ATGCCGTAG
DNA strand: AATGCAGTAGTAACTAA
Gene: ATGCAGTAG
DNA strand: AATGCTAGTAACTAA
Gene:
DNA strand: AATGCCGTAGTTACGT
Gene:
DNA strand: AATGCGCAGTCACGT
Gene:
DNA strand: ATTGCCGTAGTAACGT
Gene:

DNA strand: AATGCCGTAGTAACGT
Gene: ATGCCGTAG
DNA strand: AATGCAGTAGTAACTAA
Gene: ATGCAGTAG

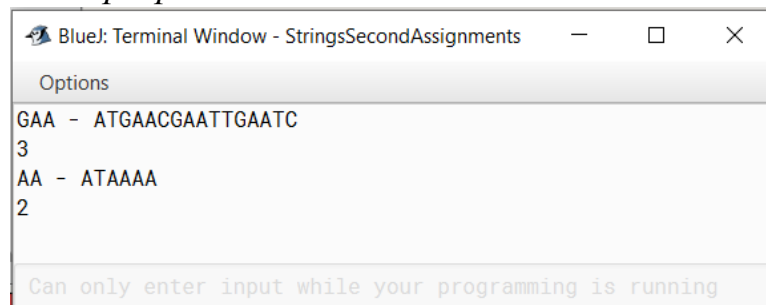
Can only enter input while your programming is running
```

Завдання 7

Створити метод `howMany` з двома параметрами `String` з назвами `stringa` та `stringb`. Цей метод повертає ціле число, яке вказує, скільки разів `stringa` з'являється в `stringb`. Кожна нова поява `stringa` не повинна перетинатися з іншими його появами. Наприклад, `howMany("AA", "ATAAAA")` повертає 2.

Код програми: [Part2.java](#)

Результати виконання програми:



```
BlueJ: Terminal Window - StringsSecondAssignments
Options
GAA - ATGAACGAATTGAATC
3
AA - ATAAAA
2

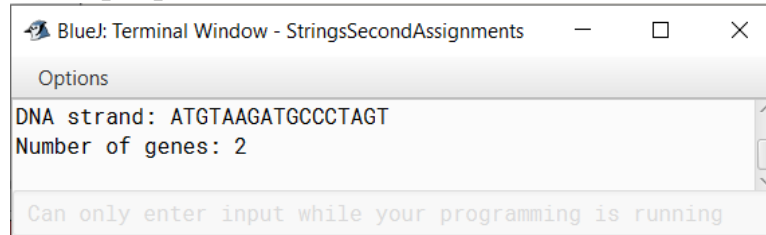
Can only enter input while your programming is running
```

Завдання 8

Створити програму для підрахунку кількості генів у ланцюзі ДНК, використавши методи з завдання 6 та створивши метод countGenes, що повертає кількість генів, знайдених у ДНК.

Код програми: [Part3.java](#)

Результати виконання програми:



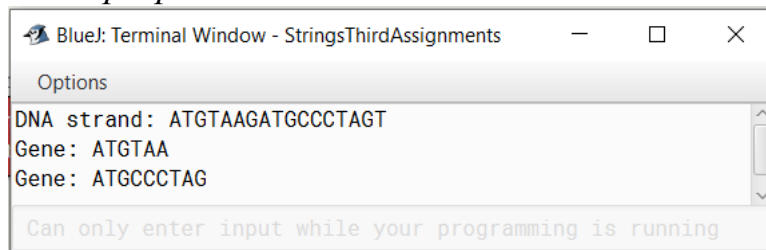
```
BlueJ: Terminal Window - StringsSecondAssignments
Options
DNA strand: ATGTAAGATGCCCTAGT
Number of genes: 2
Can only enter input while your programming is running
```

Завдання 9

Створити програму для пошуку генів у ланцюзі ДНК, використавши методи з завдання 6 та створивши метод printAllGenes, що має повернути знайдені гени за допомогою StorageResource.

Код програми: [Part1.java](#)

Результати виконання програми:



```
BlueJ: Terminal Window - StringsThirdAssignments
Options
DNA strand: ATGTAAGATGCCCTAGT
Gene: ATGTAA
Gene: ATGCCCTAG
Can only enter input while your programming is running
```

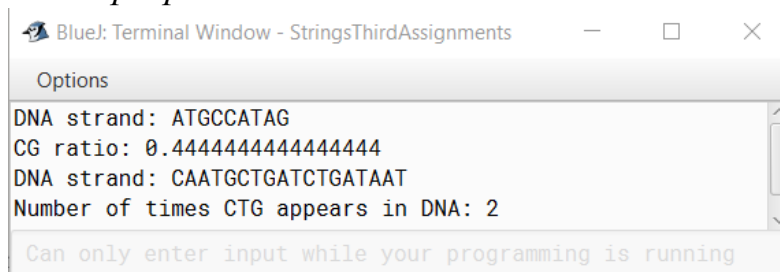
Завдання 10

Створити метод cgRatio, який повертає співвідношення С і G у ДНК як частку всього ланцюга ДНК.

Створити метод countCTG, який повертає кількість разів, коли кодон CTG з'являється в ланцюзі ДНК.

Код програми: [Part2.java](#)

Результати виконання програми:



```
BlueJ: Terminal Window - StringsThirdAssignments
Options
DNA strand: ATGCCATAG
CG ratio: 0.4444444444444444
DNA strand: CAATGCTGATCTGATAAT
Number of times CTG appears in DNA: 2
Can only enter input while your programming is running
```

Завдання 11

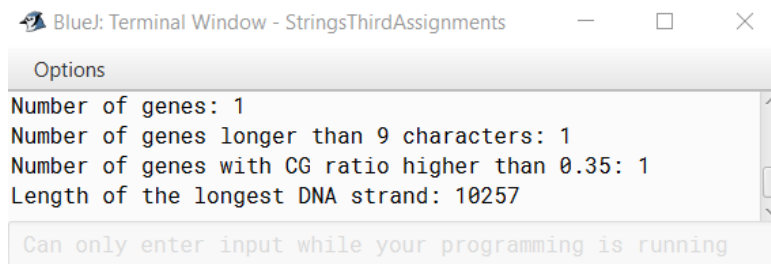
Створити програму, використавши методи з завдання 6 та створивши метод `processGenes`, який має один параметр `sr` (`StorageResource`). Цей метод обробляє всі рядки в `sr`, щоб дізнатися інформацію про них. Зокрема, він має:

- вивести кількість генів
- вивести кількість генів, довших за 9 символів;
- вивести кількість генів, CG-співвідношення яких перевищує 0,35;
- вивести довжину найдовшого гена.

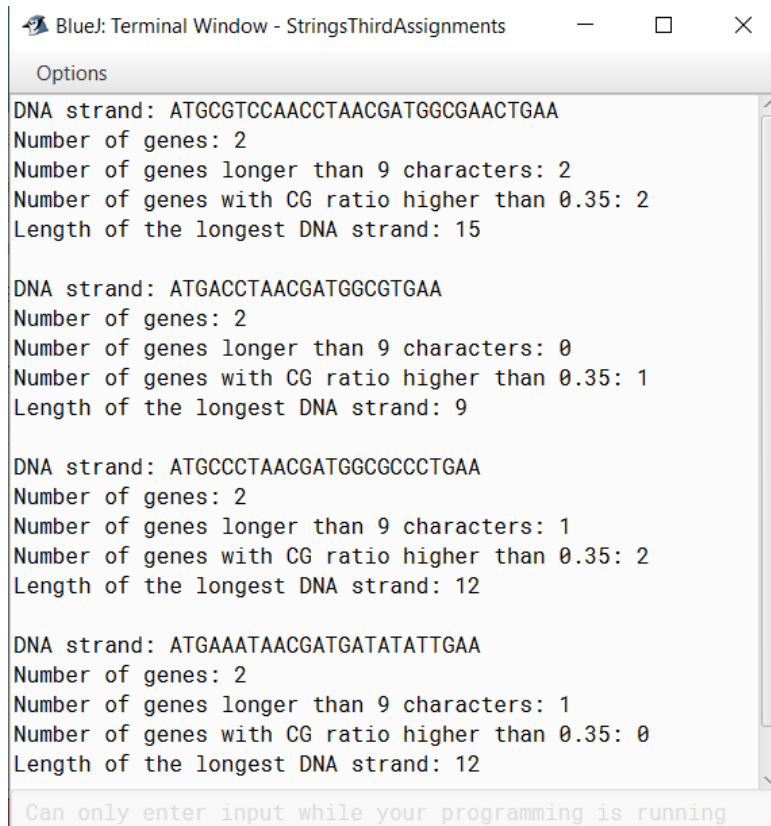
Створити метод `testProcessGenes`. Для тестування використати наданий файл `brcalline.txt` та ланцюги ДНК з генами різної довжини та різними CG-співвідношеннями.

Код програми: [Part3.java](#)

Результати виконання програми:



```
BlueJ: Terminal Window - StringsThirdAssignments
Options
Number of genes: 1
Number of genes longer than 9 characters: 1
Number of genes with CG ratio higher than 0.35: 1
Length of the longest DNA strand: 10257
Can only enter input while your programming is running
```



```
BlueJ: Terminal Window - StringsThirdAssignments
Options
DNA strand: ATGCGTCCAACCTAACGATGGCGAACTGAA
Number of genes: 2
Number of genes longer than 9 characters: 2
Number of genes with CG ratio higher than 0.35: 2
Length of the longest DNA strand: 15

DNA strand: ATGACCTAACGATGGCGTGAA
Number of genes: 2
Number of genes longer than 9 characters: 0
Number of genes with CG ratio higher than 0.35: 1
Length of the longest DNA strand: 9

DNA strand: ATGCCCTAACGATGGCGCCCTGAA
Number of genes: 2
Number of genes longer than 9 characters: 1
Number of genes with CG ratio higher than 0.35: 2
Length of the longest DNA strand: 12

DNA strand: ATGAAATAACGATGATATATTGAA
Number of genes: 2
Number of genes longer than 9 characters: 1
Number of genes with CG ratio higher than 0.35: 0
Length of the longest DNA strand: 12
Can only enter input while your programming is running
```

Завдання 12

Створити метод `countryInfo`, який повертає рядок інформації про країну або повертає «no result», якщо інформація про країну відсутня.

Створити метод `listExportersTwoProducts`, який має три параметри: `CSVParser`, `exportItem1`, `exportItem2`. Цей метод виводить назви всіх країн, які мають як елементи експорту `exportItem1`, так і `exportItem2`.

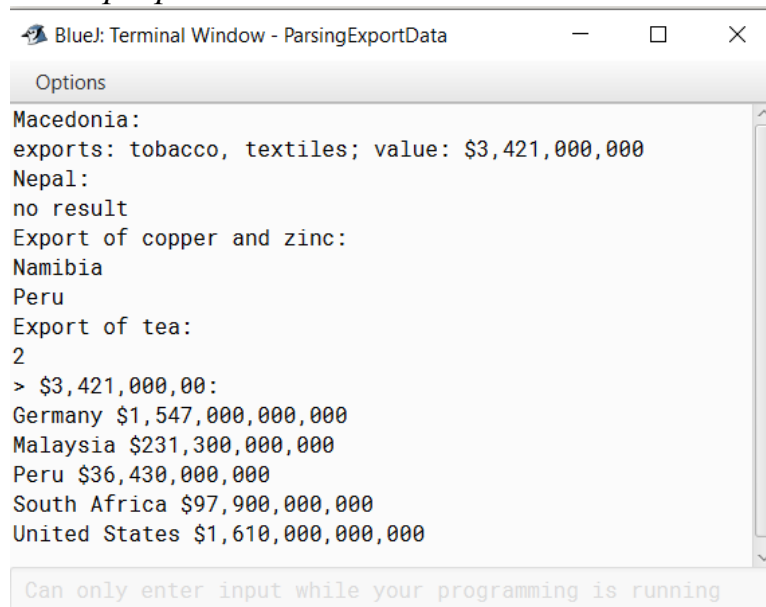
Створити метод `numberOfExporters`, який має два параметри: `CSVParser` та `exportItem`. Цей метод повертає кількість країн, які експортують `exportItem`.

Створити метод `bigExporters`, що має два параметри: `CSVParser` та `amount`. Цей метод виводить назви країн і вартість для всіх країн, у яких рядок вартості довший за рядок `amount`.

Створити метод `tester`. Для тестування використати надані файли.

Код програми: [ParsingExportDataAssignmentRunner.java](#)

Результати виконання програми:



```
BlueJ: Terminal Window - ParsingExportData
Options
Macedonia:
exports: tobacco, textiles; value: $3,421,000,000
Nepal:
no result
Export of copper and zinc:
Namibia
Peru
Export of tea:
2
> $3,421,000,00:
Germany $1,547,000,000,000
Malaysia $231,300,000,000
Peru $36,430,000,000
South Africa $97,900,000,000
United States $1,610,000,000,000
Can only enter input while your programming is running
```

Завдання 13

Створити метод `coldestHourInFile`, який має один параметр, `CSVParser`. Цей метод повертає `CSVRecord` із найнижчою температурою у файлі, а отже, і всю інформацію про найнижчу температуру, наприклад, час.

Створити метод `fileWithColdestTemperature`, що повертає рядок, який є назвою файлу з найнижчою температурою серед обраних користувачем.

Створити метод `lowestHumidityInFile`, який має один параметр, `CSVParser parser`. Цей метод повертає `CSVRecord` із найнижчою вологістю.

Створити метод `lowestHumidityInManyFiles`, який повертає `CSVRecord`, що має найнижчу вологість серед усіх файлів.

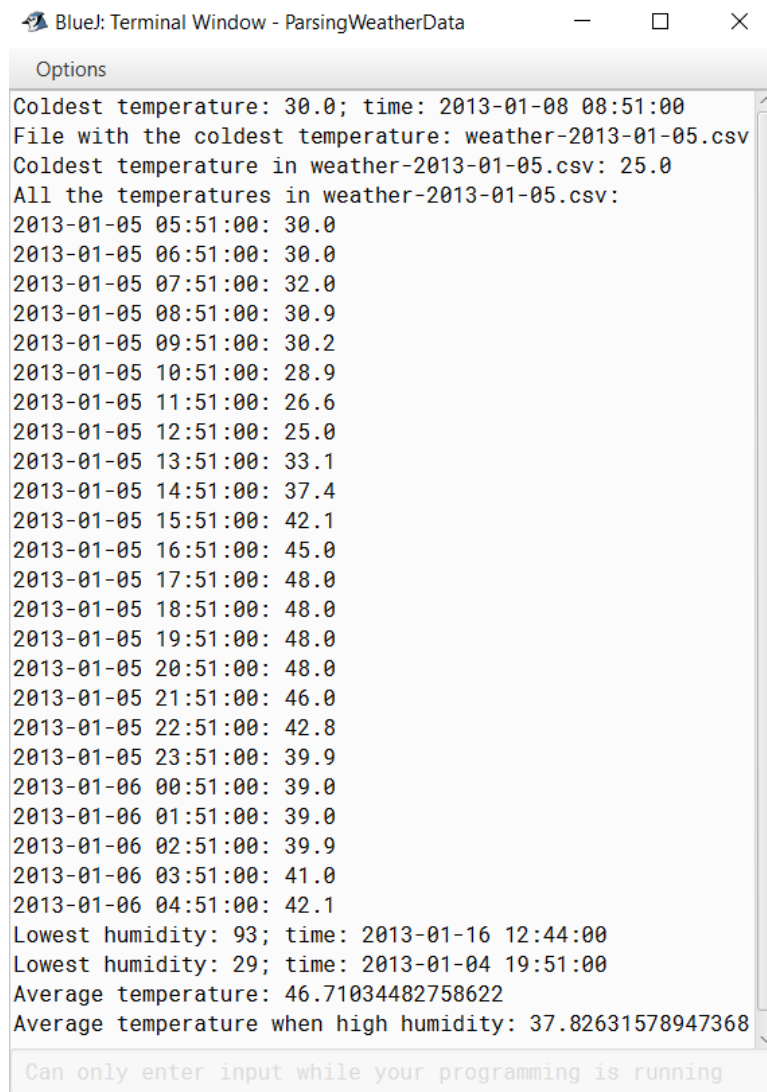
Створити метод `averageTemperatureInFile`, який має один параметр, `CSVParser parser`. Цей метод повертає середню температуру у файлі.

Створити метод `averageTemperatureWithHighHumidityInFile`, який має два параметри: `CSVParser parser` та `int value`. Цей метод повертає середню температуру лише тих температур, коли вологість була більшою або дорівнювала `value`.

Для тестування використати надані файли.

Код програми: [ParsingWeatherDataAssignmentRunner.java](#)

Результати виконання програми:



```
Options
Coldest temperature: 30.0; time: 2013-01-08 08:51:00
File with the coldest temperature: weather-2013-01-05.csv
Coldest temperature in weather-2013-01-05.csv: 25.0
All the temperatures in weather-2013-01-05.csv:
2013-01-05 05:51:00: 30.0
2013-01-05 06:51:00: 30.0
2013-01-05 07:51:00: 32.0
2013-01-05 08:51:00: 30.9
2013-01-05 09:51:00: 30.2
2013-01-05 10:51:00: 28.9
2013-01-05 11:51:00: 26.6
2013-01-05 12:51:00: 25.0
2013-01-05 13:51:00: 33.1
2013-01-05 14:51:00: 37.4
2013-01-05 15:51:00: 42.1
2013-01-05 16:51:00: 45.0
2013-01-05 17:51:00: 48.0
2013-01-05 18:51:00: 48.0
2013-01-05 19:51:00: 48.0
2013-01-05 20:51:00: 48.0
2013-01-05 21:51:00: 46.0
2013-01-05 22:51:00: 42.8
2013-01-05 23:51:00: 39.9
2013-01-06 00:51:00: 39.0
2013-01-06 01:51:00: 39.0
2013-01-06 02:51:00: 39.9
2013-01-06 03:51:00: 41.0
2013-01-06 04:51:00: 42.1
Lowest humidity: 93; time: 2013-01-16 12:44:00
Lowest humidity: 29; time: 2013-01-04 19:51:00
Average temperature: 46.71034482758622
Average temperature when high humidity: 37.82631578947368

Can only enter input while your programming is running
```

Завдання 14

Створити метод `totalBirths`, що виводить: загальну кількість імен, загальну кількість дітей, кількість хлопчиків, кількість дівчаток, кількість чоловічих імен і кількість жіночих імен.

Створити метод `getRank`, який виводить ранг імені у файлі для заданої статі (1 — ім'я з найбільшою кількістю народжених). Якщо імені немає у файлі, повертається -1.

Створити метод `getName`, який виводить ім'я з заданим рангом для заданої статі з обраного файлу. Якщо ранг не наявний у файлі, вивести «NO NAME».

Створити метод `whatIsNameInYear`, який визначає імена з однаковими рангами в обраних користувачем файлах. Тобто визначається ранг імені в першому файлі, а потім ім'я з тим самим рангом у другому файлі.

Створити метод `yearOfHighestRank`, який дозволяє користувачеві обрати діапазон файлів для обробки та виводить рік із найвищим рангом для заданих імені та статі серед цих файлів. Якщо ім'я та статя відсутні в усіх вибраних файлах, метод має повернути -1.

Створити метод `getAverageRank`, який дозволяє користувачеві обрати діапазон файлів для обробки та виводить середній ранг імені та статі для вибраних файлів. Метод має повернути -1.0, якщо ім'я не вказано в жодному з вибраних файлів.

Створити метод `getTotalBirthsRankedHigher`, який виводить загальну кількість дітей тієї самої статі й року, імена яких мають вищий рейтинг, ніж задане ім'я.

Для тестування використати надані файли.

Код програми: [NamesAssignmentRunner.java](#)

Результати виконання програми:



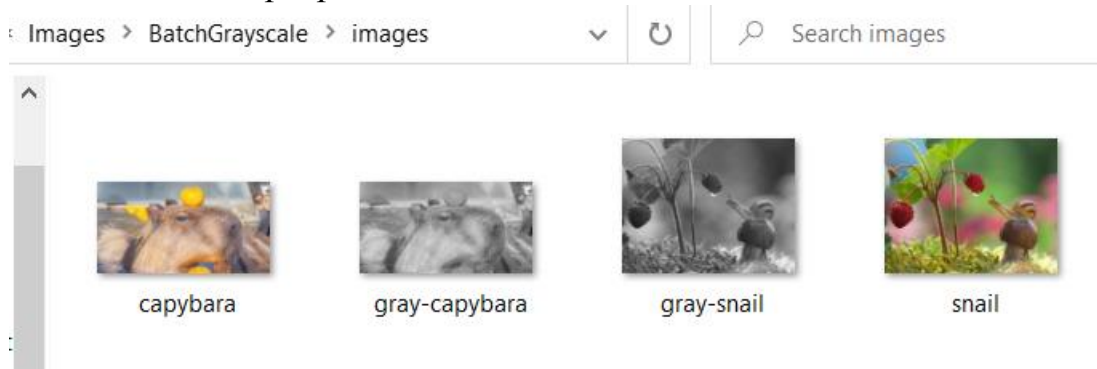
```
BlueJ: Terminal Window - Names
Options
Total girls births: 18
Total girls names: 5
Total boys births: 29
Total boys names: 5
Total births: 47
Total names: 10
Name: Mason; rank: 2
Rank: 2; name: Mason
Isabella would be Sophia
Name: Mason; year of highest rank: 2012
Name: Mason; average rank: 3.0
Name: Ethan; number of births (names ranked higher): 15
Can only enter input while your programming is running
```

Завдання 15

Створити програму, що дозволяє користувачу обирати кілька файлів та створює нові зображення, що є версіями вихідних зображень у градаціях сірого. Кожне нове зображення зберігається в новому файлі зі словом «gray-» перед назвою вихідного файлу. Наприклад, якщо вихідний файл мав назву lion.png, нове зображення в сірому кольорі матиме назву gray-lion.png.

Код програми: [BatchGrayscaleAssignmentRunner.java](#)

Результати виконання програми:



Завдання 16

Створити програму, що дозволяє користувачу обирати кілька файлів та створює нові зображення, що є інвертованими версіями вихідних зображень (потрібно змінити колір на протилежне значення в діапазоні від 0 до 255). Кожне нове зображення зберігається в новому файлі зі словом «inverted-» перед назвою вихідного файлу.

Код програми: [ImageInversionAssignmentRunner.java](#)

Результати виконання програми:

