

Description for Assignment 4

Author: Xiaonan Liu

To reuse the code, I design a class named DNAGenerator, it has servals fields include the thread count, when we instance the class, we could define the thread count to be one or five, then we could use thread poll to control how many threads we should use, and don't need to write two repeat program to generate the DNA strings.

In DNAGenerator class, we only has a random rate, total count we need to generate the DNA string, also in every thread how many String we need to generate, and only a container for the result of our generated string use ArrayList.

DNAGenerator
<ul style="list-style-type: none">+ random : Random+ GENOME: final char[]{'A','T','G','C'}+ totalGemomeCount : int, 100+ threadCount : int+ genomeSizePreThread: final int+ excutor: EXecutorService+ genomeStrings : List<String>
<ul style="list-style-type: none">+ Constructor+ initialThreadPoolAndRun(): void+ getGenomes(): List<String>

I define two constructors in case we need to generate more strings or customize the count, one use default total count, just as we asked by prof, 100, another one is creating this instance by the user's input. So, in the user interface, use could choose default value or customize it.

The most important function in this class, is `initializeThreadPoolAndRun`, it will create the number of threads that is asked by user and generate the Strings base on the requirements. I use thread pool to manage the threads, and add results to the list. The `getGenomes` will continue call this function while the size of the results does not match the count we needed.

```
//according to the thread count to create the threads to run the generation
//append the char in the stringBuilder and add to the list
1 usage
private void initializeThreadPoolAndRun() {
    IntStream.range(0, threadCount).forEach(item -> executor.execute(
        new Thread(() -> {
            for (int x = 0; x < genomeSizePerThread; x++) {
                StringBuilder sb = new StringBuilder();
                for (int y = 0; y < 10; y++) {
                    sb.append(GENOME[RANDOM.nextInt( bound: 4)]);
                }
                genomeStrings.add(sb.toString());
            }
        })
    ));
    executor.shutdown();
    try {
        if (!executor.awaitTermination( timeout: 3, TimeUnit.SECONDS)) {
            executor.shutdownNow();
        }
    } catch (InterruptedException e) {
        System.out.println(e.getMessage());
    }
}

// if the size is not match the totalCount,continue run previous function

3 usages
public List<String> getGenomes() {
    if (genomeStrings.size() != totalGenomeCount) {
        initializeThreadPoolAndRun();
    }
    return this.genomeStrings;
}
}
```

In the main function, we could read the user's input and get the results which is needed. In the test, I test all the functions, and have make sure all functions work properly.

```

/Users/liuxiaonan/Library/Java/JavaVirtualMachines/corretto
How many threads need to run (q/Q to exit)?
1
How many strings do you need to generate (Y/y use default count)?
100
Finished, DNAs list size: 100, execute time: 28
How many threads need to run (q/Q to exit)?
1
How many strings do you need to generate (Y/y use default count)?
200
Finished, DNAs list size: 200, execute time: 1
How many threads need to run (q/Q to exit)?
5
How many strings do you need to generate (Y/y use default count)?
100
Finished, DNAs list size: 100, execute time: 1
How many threads need to run (q/Q to exit)?
5
How many strings do you need to generate (Y/y use default count)?
200
Finished, DNAs list size: 200, execute time: 1
How many threads need to run (q/Q to exit)?
|

```

This is my output for my program, based on the results, it is not very clear that does multi threads more effective than one thread, so I tried to generate more examples to test.

```

How many strings do you need to generate (Y/y use default count)?
50000
Finished, DNAs list size: 49836, execute time: 53
How many threads need to run (q/Q to exit)?
1
How many strings do you need to generate (Y/y use default count)?
500000
Finished, DNAs list size: 500000, execute time: 98
How many threads need to run (q/Q to exit)?

```

Also

```

How many threads need to run (q/Q to exit)?
1
How many strings do you need to generate (Y/y use default count)?
10000000Finished, DNAs list size: 4214048, execute time: 4070
How many threads need to run (q/Q to exit)?
5
How many strings do you need to generate (Y/y use default count)?
10000000
Finished, DNAs list size: 2002081, execute time: 3003
How many threads need to run (q/Q to exit)?

```

I think when the more data we are requesting, we could tell that, multi threads did be more effective than single thread.