

Exercise 4.1

All answers/completed tasks can be found in code as it made more sense to leave them there

Exercise 4.2

All answers/completed tasks can be found in code as it made more sense to leave them there

Exercise 4.3

All other answers/completed tasks can be found in code as it made more sense to leave them there

Part 6.

Paralles is few ms faster, no significant difference

Part 12.

Slower - sequential = 13 000 ms, parallel = 14 000 ms +-

Exercise 4.4

Part 1.

Code

```
double result = ns.mapToDouble(n -> n).map(n -> 1 / n).sum();
```

Results for first run can be seen below

```
Time elapsed = 5872 ms  
Sum = 21,3004815003479420
```

Part 2.

Code

```
double result = ns.parallel().mapToDouble(n -> n).map(n -> 1 / n).sum();
```

Results for second run with `parallel()`

```
Time elapsed = 718 ms
Sum = 21,3004815003479420
```

Part 3.

Code

```
double result = 0;

for(double i = 1; i <= N; i++) {
    result += 1 / i;
}
```

Results for for loop sum

```
Time elapsed = 2538 ms
Sum = 21,3004815013485500
```

Difference between `sum()` and `for` loop is 0,000000001000608

Part 4.

Code

```
double result = DoubleStream.generate(new DoubleSupplier() {
    double N = 0;
    @Override
    public double getAsDouble() {
        N += 1;

        return 1 / N;
    }
}).limit(N).sum();
```

Results

```
Time elapsed = 5331 ms
Sum = 21,3004815013479420
```

Part 5.

Code

```
double result = DoubleStream.generate(new DoubleSupplier() {
```

```
double N = 0;
@Override
public double getAsDouble() {
    N += 1;

    return 1 / N;
}
}).limit(N).parallel().sum();
```

Results

```
Time elapsed = 3395 ms
Sum = 103,4281533341543200

Time elapsed = 3741 ms
Sum = 94,2123312579626300

Time elapsed = 4367 ms
Sum = 93,5230909226233300
```

Simply because field `N` inside `DoubleSupplier` is not synchronized and accessed by multiple threads, the results differ. By synchronizing `getAsDouble()` function we get correct result however the computation is veeery slow compared to e.g. sequential run.

```
Time elapsed = 65736 ms
Sum = 21,3004815013471460
```