Assignment 1

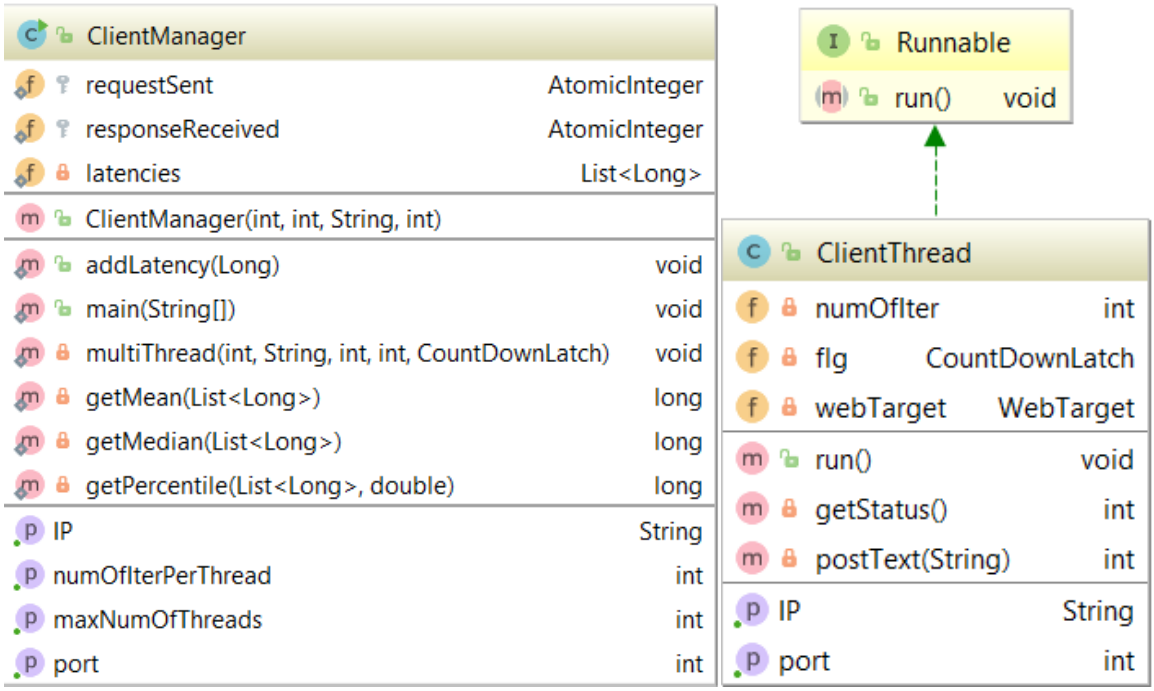
Zhenyuan Xi

https://github.com/Zhenyuan-Xi/BSDS

1. I have two packages for server and client, respectively.

   In the server package, I have three classes: 1). main function, 2). get function, 3). post function

   In the client package, I have two classes: 1). main function and some util functions to calculate the results, in the main(), I create the four parts for Warm up, Loading, Peak and Cool down. Then calculate the duration for each part and output the results. 2). a helper class for Thread which implements Runnable and override run() inside.



2. github link: https://github.com/Zhenyuan-Xi/BSDS

3. Step4: 20/100 EC2

```
Server starting .....: http://ec2-34-205-125-134.compute-1.amazonaws.com:8080/myapp/myget
Client starting .....Time: 2018-09-30 15:29:04
Warmup phase: All threads running ....
Warmup phase complete: Time 19 seconds
Loading phase: All threads running ....
Loading phase complete: Time 23 seconds
Peak phase: All threads running ....
Peak phase complete: Time 29 seconds
Cooldown phase: All threads running ....
Cooldown phase complete: Time 19 seconds
==========================================
Total number of requests sent: 7400
Total number of Successful responses: 7400
Test Wall Time: 91 seconds
```

Step4: 100/100 EC2

```
Server starting .....: http://ec2-34-205-125-134.compute-1.amazonaws.com:8080/myapp/myget
Client starting .....Time: 2018-09-30 15:24:04
Warmup phase: All threads running ....
Warmup phase complete: Time 25 seconds
Loading phase: All threads running ....
Loading phase complete: Time 55 seconds
Peak phase: All threads running ....
Peak phase complete: Time 112 seconds
Cooldown phase: All threads running ....
Cooldown phase complete: Time 33 seconds
==========================================
Total number of requests sent: 37000
Total number of Successful responses: 37000
Test Wall Time: 227 seconds
```

4. Step5: 20/100 EC2

```
Server starting .....: http://ec2-34-205-125-134.compute-1.amazonaws.com:8080/myapp/myget
Client starting .....Time: 2018-09-30 15:29:04
Warmup phase: All threads running ....
Warmup phase complete: Time 19 seconds
Loading phase: All threads running ....
Loading phase complete: Time 23 seconds
Peak phase: All threads running ....
Peak phase complete: Time 29 seconds
Cooldown phase: All threads running ....
Cooldown phase complete: Time 19 seconds
==========================================
Total number of requests sent: 7400
Total number of Successful responses: 7400
Test Wall Time: 91 seconds
Overall throughput across all phases: 81 seconds
Mean latency for all requests: 126 milliseconds
Median latency for all requests: 94 milliseconds
99th percentile latency: 359 milliseconds
95th percentile latency: 250 milliseconds
```

Step5: 100/100 EC2

```
Server starting .....: http://ec2-34-205-125-134.compute-1.amazonaws.com:8080/myapp/myget
Client starting .....Time: 2018-09-30 15:24:04
Warmup phase: All threads running ....
Warmup phase complete: Time 25 seconds
Loading phase: All threads running ....
Loading phase complete: Time 55 seconds
Peak phase: All threads running ....
Peak phase complete: Time 112 seconds
Cooldown phase: All threads running ....
Cooldown phase complete: Time 33 seconds
==========================================
Total number of requests sent: 37000
Total number of Successful responses: 37000
Test Wall Time: 227 seconds
Overall throughput across all phases: 162 seconds
Mean latency for all requests: 398 milliseconds
Median latency for all requests: 312 milliseconds
99th percentile latency: 1265 milliseconds
95th percentile latency: 953 milliseconds
```

5. 20/100 Lambda

```
Client starting .....Time: 2018-10-02 17:48:33
Warmup phase: All threads running ....
Warmup phase complete: Time 90 seconds
Loading phase: All threads running ....
Loading phase complete: Time 103 seconds
Peak phase: All threads running ....
Peak phase complete: Time 139 seconds
Cooldown phase: All threads running ....
Cooldown phase complete: Time 96 seconds
==========================================
Total number of requests sent: 7400
Total number of Successful responses: 3688
Test Wall Time: 431 seconds
Overall throughput across all phases: 17 seconds
Mean latency for all requests: 596 milliseconds
Median latency for all requests: 534 milliseconds
99th percentile latency: 1483 milliseconds
95th percentile latency: 1026 milliseconds
```

100/100 Lambda

```
Client starting .....Time: 2018-10-02 17:28:08
Warmup phase: All threads running ....
Warmup phase complete: Time 111 seconds
Loading phase: All threads running ....
Loading phase complete: Time 209 seconds
Peak phase: All threads running ....
Peak phase complete: Time 421 seconds
Cooldown phase: All threads running ....
Cooldown phase complete: Time 146 seconds
==========================================
Total number of requests sent: 37000
Total number of Successful responses: 18479
Test Wall Time: 888 seconds
Overall throughput across all phases: 41 seconds
Mean latency for all requests: 1490 milliseconds
Median latency for all requests: 1041 milliseconds
99th percentile latency: 5648 milliseconds
95th percentile latency: 4045 milliseconds
```

6. Keep increasing the number of threads until I get an exception when I set the number of threads as 1000 and the number of iterations is 100.

   Because a total of 200000 GET and POST requests should be made but an error happened before that.

```
Server starting .....: http://ec2-34-205-125-134.compute-1.amazonaws.com:8080/myapp/myget
Client starting .....Time: 2018-10-02 13:01:06
Warmup phase: All threads running ....
Warmup phase complete: Time 129 seconds
Loading phase: All threads running ....
Loading phase complete: Time 1164 seconds
Peak phase: All threads running ....
Exception in thread "Thread-1545" javax.ws.rs.ProcessingException: Java heap space
    at org.glassfish.jersey.client.ClientRuntime.invoke(ClientRuntime.java:287)
    at org.glassfish.jersey.client.JerseyInvocation.lambda$invoke$0(JerseyInvocation.java:753)
    at org.glassfish.jersey.internal.Errors.process(Errors.java:316)
    at org.glassfish.jersey.internal.Errors.process(Errors.java:298)
    at org.glassfish.jersey.internal.Errors.process(Errors.java:229)
    at org.glassfish.jersey.process.internal.RequestScope.runInScope(RequestScope.java:414)
    at org.glassfish.jersey.client.JerseyInvocation.invoke(JerseyInvocation.java:752)
    at org.glassfish.jersey.client.JerseyInvocation$Builder.method(JerseyInvocation.java:419)
    at org.glassfish.jersey.client.JerseyInvocation$Builder.get(JerseyInvocation.java:319)
    at BSDS.ClientThread.getStatus(ClientThread.java:66)
    at BSDS.ClientThread.run(ClientThread.java:40)
    at java.lang.Thread.run(Thread.java:748)
```

7. output the list of latencies in my class to a csv file, and plot it to see the distribution of latency, where the latency is endtime – starttime, which I use System.currentTimeMillis() to record.

   Take 20/100 as an example.