Program Structures and Algorithms

Fall 2024

Group members: Akshit Saxena, Sanskruti Manoria   
Nuids: 002475083, 002643300  
GitHub link: <https://github.com/saxena-akshit/WebKrawler-INFO6205>  
App name : WebKrawler  
Task: Build a web crawler that uses parallel/asynchronous processing to find web pages, given a set of starting pages.

***Contents***  
**1.Abstract**

**2.Web-Crawler Implementation**

**3.Code & Output**

**4.Unit Test Cases**

**5.Benchmarking**

**6.Conclusion**

**7. References**

**4. Test Cases  
  
1. WebCrawlerApplicationTests**

**A screen shot of a computer

Description automatically generated**

**2. TestControllerTest  
A computer screen shot of a computer

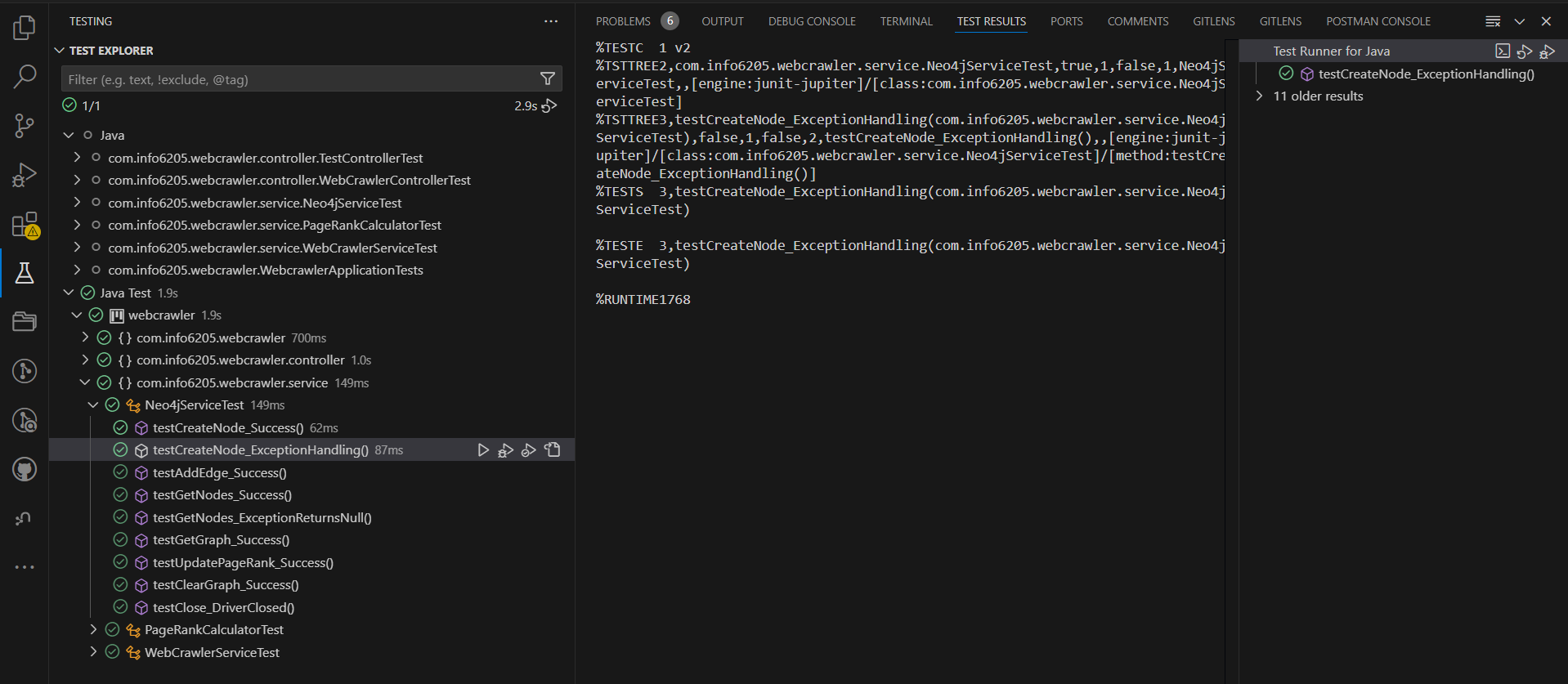
Description automatically generated  
  
3. WebCrwaler Controller Test  
A screenshot of a computer

Description automatically generated**

**4. WebCrawler Services**

**4.1 Neo4j Tests**

4.1.1 TestCreateNode()A screenshot of a computer

Description automatically generated **4.1.2** createNodeExceptionHandling()****

4.1.3 TestAddEdge\_Sucess()A screenshot of a computer

Description automatically generated  
4.1.4 TestGetNodes\_Sucess()A screenshot of a computer

Description automatically generated

4.1.5 TestGetNodesExceptionHandlingNull()A screenshot of a computer

Description automatically generated

4.1.6 TestGetGraph\_Success()A screenshot of a computer

Description automatically generated

4.1.7 TestUpdatePagerank\_Success()

A screenshot of a computer

Description automatically generated

4.1.8 TestClearGraph\_Success()A screenshot of a computer

Description automatically generated

4.1.9 TestClose\_DriverClosed()A screenshot of a computer

Description automatically generated

**4.2.PageRankCalculatorTest**

4.2.1 TestComputePageRank()A screenshot of a computer

Description automatically generated

4.2.2 TestCreateNeighbors()A screenshot of a computer

Description automatically generated

4.2.3 TestGetOutDegree()A screenshot of a computer

Description automatically generated  
  
**4.3. WebCrawlerServices**  
4.3.1 TestStartCrawl()A screenshot of a computer

Description automatically generated

4.3.2 TestCalculatePrioroty()A screenshot of a computer

Description automatically generated

4.3.3 TestProcessURL()A screenshot of a computer

Description automatically generated

4.3.4 TestBlackListPriority()A screenshot of a computer

Description automatically generated

4.3.5 TestLowPriorityForImages()A screenshot of a computer

Description automatically generated

4.3.6 TestPriorityForGovURLs A screenshot of a computer

Description automatically generated

4.3.7 TestPriorityForResearchURLs A screenshot of a computer

Description automatically generated

4.3.8 TestOnionURLs()A screenshot of a computer

Description automatically generated

4.3.9 TestProcessURLWithValidLinks()A screenshot of a computer

Description automatically generated

4.3.10 TestPageRankResponseStructure()A screenshot of a computer

Description automatically generated

4.3.11TestNoDuplicateBlackListedEntries()

A screenshot of a computer

Description automatically generated

4.3.12 TestThreadPoolShutDown()A screenshot of a computer

Description automatically generated

Overview of Test cases:

**A screenshot of a computer

Description automatically generated**