During the last sprint, metrics **Eo_Lcom1,Eo_Lcom2** and Eo**_Lcom3** were implemented and integrated to the **Jpeek** project:
https://github.com/HSE-Eolang/jpeek/tree/master/src/eo

The tests for the metrics are the following:
Overall score is an arithmetic average of all metric scores from the "Score" column in the table. "E/C" shows the total amount of classes measured by this metric, in most cases the number will be the same for all metrics, and the elements our statistics pays attention to (we ignore max and min). Mathematical average of all measurements, which doesn't give too much information about the quality of code, but is visible here for statistical purposes; we recommend to pay attention to the "Score" column. "Green," "Yellow," and "Red" columns show how many classes received that colors on their measurements; obviously, red classes are those which have lower quality; to see them just click the name of the metric in the first column. "Score" is a weighted average of the numbers from the "Green," "Yellow," and "Red" columns; the weight of green classes is 1.0, yellow ones get 0.25, and red ones get 0.05; if the color of the number is green, the quality is high enough, if it's orange the quality is average, if it's red, the quality is too low, for this particular metric; in small font below the score value you can see how big is the difference between this metric score and the average score for the entire code base."Defects" is the percentage of classes that fall out of one-sigma standard deviation interval.

1) *test project*: **Eclipse Deeplearning4J Examples**

https://github.com/eclipse/deeplearning4j-examples/tree/master/dl4j-examples

Description: This project contains a set of examples that demonstrate use of the high level DL4J API to build a variety of neural networks. The DL4J ecosystem also allows users to build neural networks with SameDiff (part of the ND4J library) with a more fine grained API.

the result:

Overall score[1] is **8.24** out of 10. Here is the matrix.

8.2417

Metric	E/C[2]	μ[3]	σ[3]	Min	Max	Green[4]	Yellow[4]	Red[4]	Score[5]	Defects[6]	Graph
EO_LCOM1	30/123 0%	17.33	37.89	1.00	117.00	30 24%	0 0%	0 0%	10.00 21%	13%	
EO_LCOM2	30/123 0%	0.69	0.17	0.33	0.95	19 15%	10 8%	1 1%	7.18 -13%	50%	
EO_LCOM3	26/123 0%	0.92	0.13	0.62	1.20	26 21%	0 0%	0 0%	10.00 21%	27%	
LCOM	30/123 0%	17.33	37.89	1.00	117.00	13 11%	7 6%	10 8%	5.08 -38%	13%	
LCOM2	30/123 0%	0.69	0.17	0.33	0.95	19 15%	10 8%	1 1%	7.18 -13%	50%	
LCOM3	26/123 0%	0.92	0.13	0.62	1.20	26 21%	0 0%	0 0%	10.00 21%	27%	

EO_LCOM1,EO_LCOM2,EO_LCOM3 - implemented **EO** metrics
LCOM,LCOM2,LCOM3 - original **Java** metrics

2) *test project*: **JavaParser**

https://github.com/javaparser/javaparser

Description: This project contains a set of libraries implementing a Java 1.0 - Java 15 Parser with advanced analysis functionalities.

the result:

Overall score[1] is **8.33** out of 10. Here is the matrix.

8.3333

Metric	E/C[2]	μ[3]	σ[3]	Min	Max	Green[4]	Yellow[4]	Red[4]	Score[5]	Defects[6]	Graph
EO_LCOM1	3/32 0%	5.50	4.50	1.00	10.00	3 9%	0 0%	0 0%	10.00 20%	0%	
EO_LCOM2	2/32 0%	0.68	0.18	0.50	0.87	1 3%	1 3%	0 0%	6.25 -25%	0%	
EO_LCOM3	2/32 0%	1.08	0.08	1.00	1.17	2 6%	0 0%	0 0%	10.00 20%	0%	
LCOM	3/32 0%	5.50	4.50	1.00	10.00	2 6%	1 3%	0 0%	7.50 -10%	0%	
LCOM2	2/32 0%	0.68	0.18	0.50	0.87	1 3%	1 3%	0 0%	6.25 -25%	0%	
LCOM3	2/32 0%	1.08	0.08	1.00	1.17	2 6%	0 0%	0 0%	10.00 20%	0%	

EO_LCOM1,EO_LCOM2,EO_LCOM3 - implemented **EO** metrics
LCOM,LCOM2,LCOM3 - original **Java** metrics

3) *test project:*Json-iterator

https://github.com/json-iterator/java

Description:jsoniter (json-iterator) is fast and flexible JSON parser available in Java

the result:

Overall score[1] is **7.28** out of 10. Here is the matrix.

7.2802

Metric	E/C[2]	μ[3]	σ[3]	Min	Max	Green[4]	Yellow[4]	Red[4]	Score[5]	Defects[6]	Graph
EO_LCOM1	56/151 0%	60.79	200.18	1.00	885.00	56 37%	0 0%	0 0%	10.00 37%	6%	
EO_LCOM2	41/151 0%	0.57	0.25	0.08	0.95	20 13%	12 8%	9 6%	5.72 -21%	37%	
EO_LCOM3	44/151 0%	0.77	0.31	0.09	1.33	33 22%	6 4%	5 3%	7.90 8%	25%	
LCOM	56/151 0%	60.79	200.18	1.00	885.00	34 23%	5 3%	17 11%	6.45 -11%	6%	
LCOM2	41/151 0%	0.57	0.25	0.08	0.95	20 13%	12 8%	9 6%	5.72 -21%	37%	
LCOM3	44/151 0%	0.77	0.31	0.09	1.33	33 22%	6 4%	5 3%	7.90 8%	25%	

EO_LCOM1,EO_LCOM2,EO_LCOM3 - implemented **EO** metrics
LCOM,LCOM2,LCOM3 - original **Java** metrics

Overall:
All implemented metrics produce the same result as the original solution, in the 3-rd test the score of EO_LCOM_1 is higher than original java LCOM. Other metrics share the same score result.