

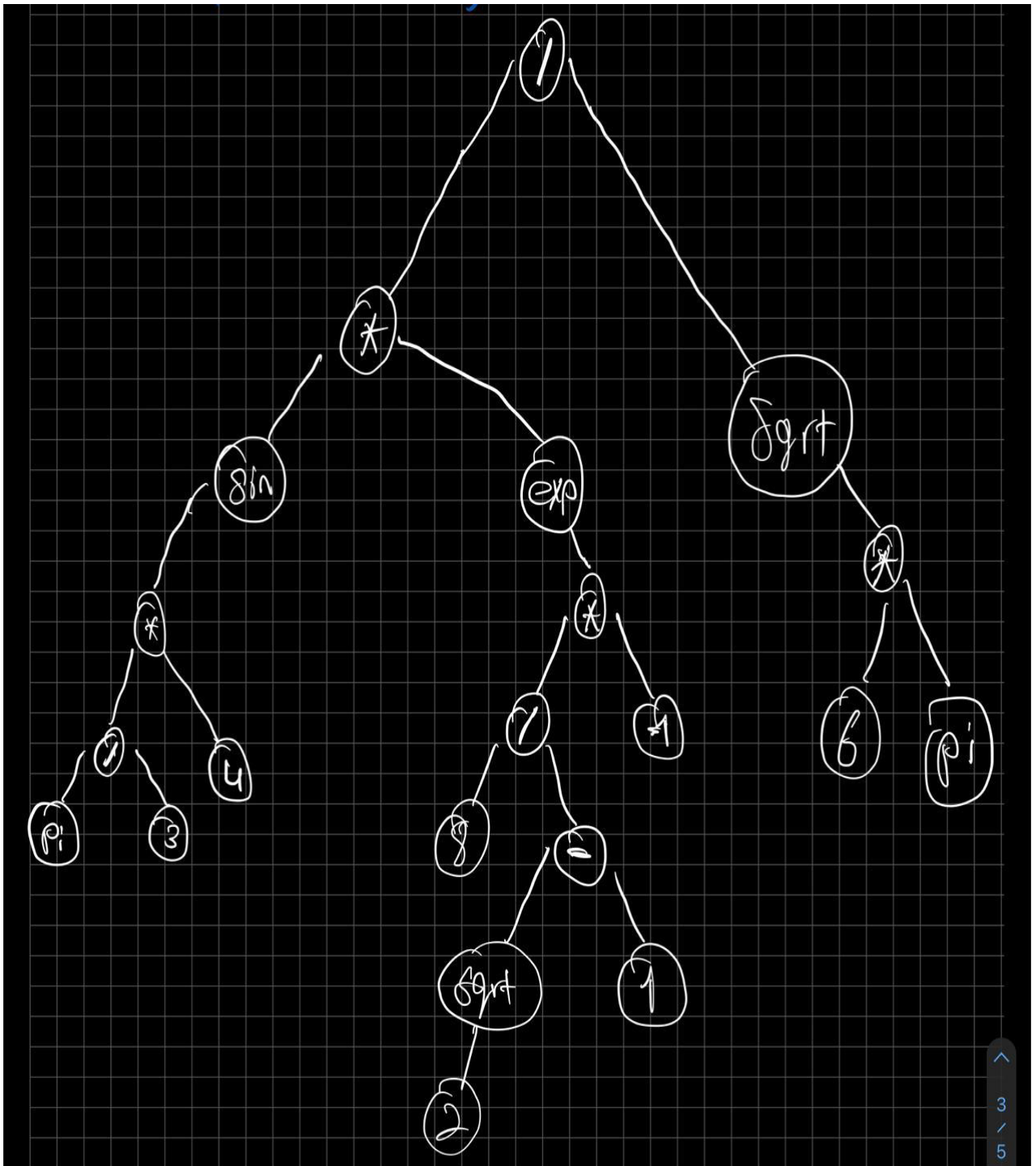
ADS LAB 2 REPORT

Elsayed, ahmed
MEMBER 2 TEAM 15

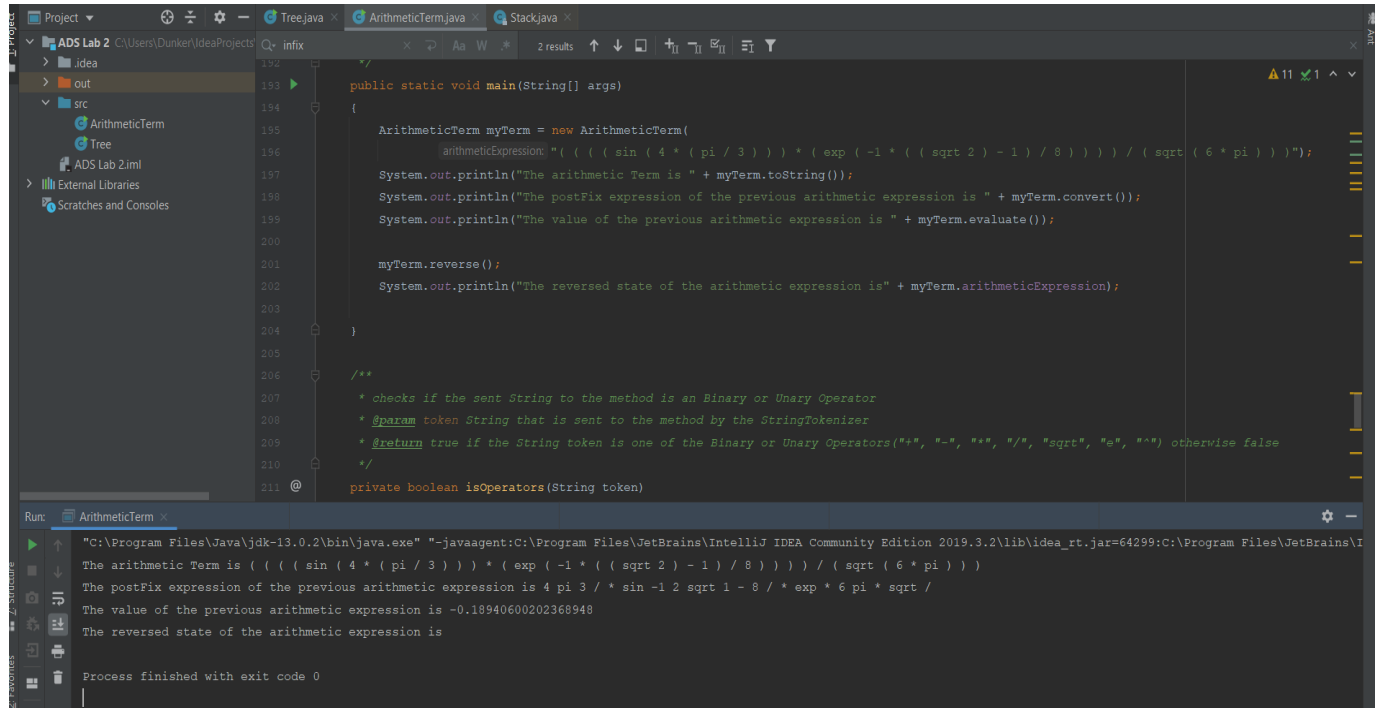
PREPARATION

Problem 5:

1. Code refactoring in ArithmeticTerm has been done based on Lab 1 feedback
2. The fully parenthesized expression of the provided arithmetic expression is:
 $((((\sin(4 * (\pi / 3))) * (\exp(-1 * ((\sqrt{2}) - 1) / 8)))) / (\sqrt{6 * \pi}))$
Tree of the previous Arithmetic expression solved in inOrderTraversal:



3. Results of conversion and evaluation of the expression:



```
192 //
193 public static void main(String[] args)
194 {
195     ArithmeticTerm myTerm = new ArithmeticTerm(
196         arithmeticExpression: "( ( ( sin ( 4 * ( pi / 3 ) ) ) * ( exp ( -1 * ( ( sqrt 2 ) - 1 ) / 8 ) ) ) ) / ( sqrt ( 6 * pi ) ) )";
197     System.out.println("The arithmetic Term is " + myTerm.toString());
198     System.out.println("The postFix expression of the previous arithmetic expression is " + myTerm.convert());
199     System.out.println("The value of the previous arithmetic expression is " + myTerm.evaluate());
200
201     myTerm.reverse();
202     System.out.println("The reversed state of the arithmetic expression is " + myTerm.arithmeticExpression);
203 }
204
205 /**
206  * checks if the sent String to the method is an Binary or Unary Operator
207  * @param token String that is sent to the method by the StringTokenizer
208  * @return true if the String token is one of the Binary or Unary Operators("+", "-", "*", "/", "sqrt", "e", "**") otherwise false
209  */
210 private boolean isOperators(String token)
211 @
```

Run: ArithmeticTerm x

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019.3.2\lib\idea_rt.jar=64299:C:\Program Files\JetBrains\I

The arithmetic Term is (((sin (4 * (pi / 3))) * (exp (-1 * ((sqrt 2) - 1) / 8)))) / (sqrt (6 * pi)))

The postFix expression of the previous arithmetic expression is 4 pi 3 / * sin -1 2 sqrt 1 - 8 / * exp * 6 pi * sqrt /

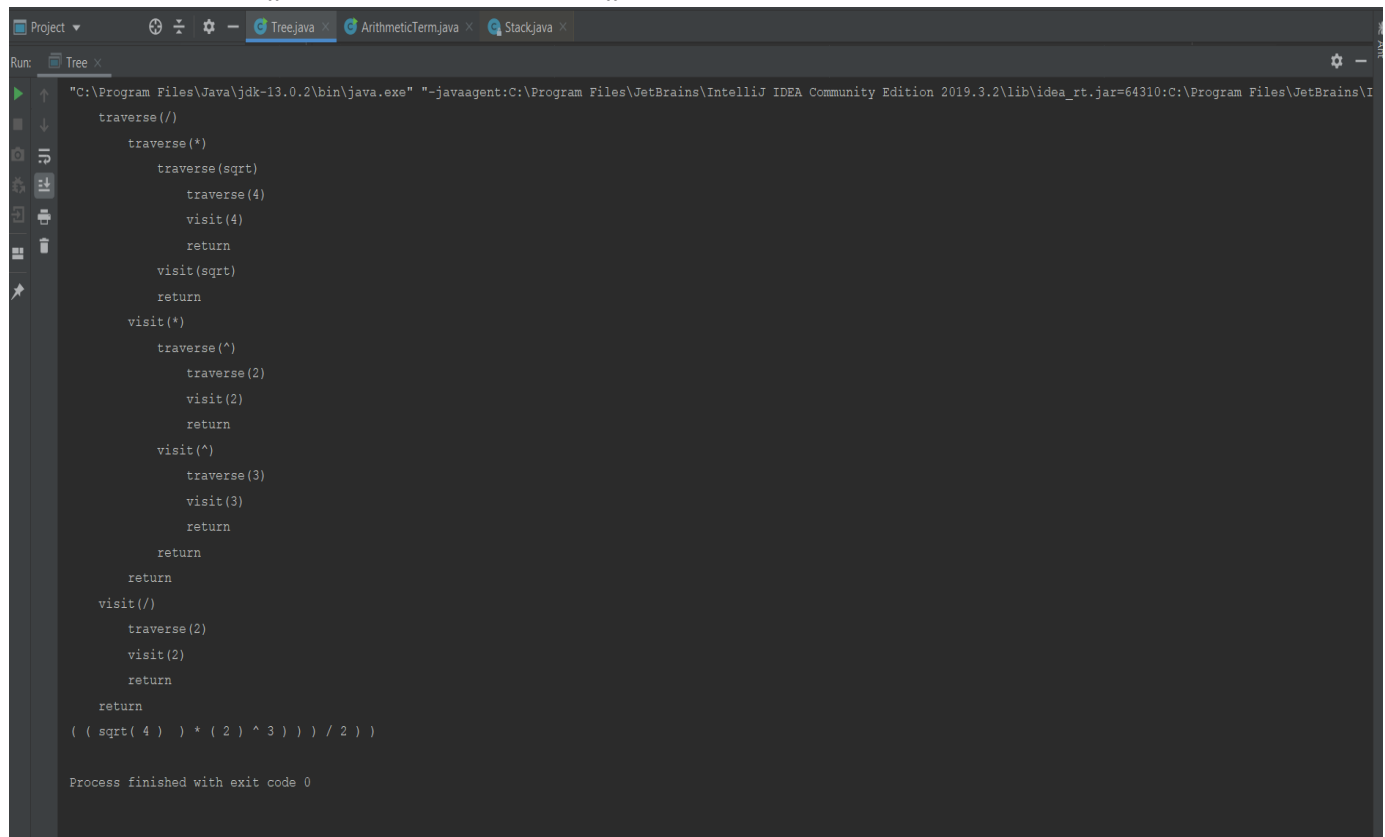
The value of the previous arithmetic expression is -0.18940600202368948

The reversed state of the arithmetic expression is

Process finished with exit code 0

4. Please check the source code for the class Tree and the inner class BiNode (had to be static in order to have objects instantiated in main method).

5. inOrderTraversal() and inFixGeneration() results:



```
1 //
2 public static void main(String[] args)
3 {
4     Tree tree = new Tree();
5     tree.inOrderTraversal();
6     tree.inFixGeneration();
7 }
8
9 /**
10  * inOrderTraversal()
11  * @return String
12  */
13 private String inOrderTraversal()
14 {
15     traverse();
16     return result;
17 }
18
19 /**
20  * traverse()
21  * @return String
22  */
23 private String traverse()
24 {
25     traverse(sqrt);
26     traverse(4);
27     visit(4);
28     return result;
29 }
30
31 /**
32  * visit()
33  * @return String
34  */
35 private String visit(String token)
36 {
37     result += token + " ";
38     return result;
39 }
40
41 /**
42  * inFixGeneration()
43  * @return String
44  */
45 private String inFixGeneration()
46 {
47     traverse();
48     return result;
49 }
50
51 /**
52  * traverse()
53  * @return String
54  */
55 private String traverse()
56 {
57     traverse(2);
58     visit(2);
59     return result;
60 }
61
62 /**
63  * visit()
64  * @return String
65  */
66 private String visit(String token)
67 {
68     result += token + " ";
69     return result;
70 }
71
72 /**
73  * inOrderTraversal()
74  * @return String
75  */
76 private String inOrderTraversal()
77 {
78     traverse();
79     return result;
80 }
81
82 /**
83  * traverse()
84  * @return String
85  */
86 private String traverse()
87 {
88     traverse(sqrt);
89     traverse(4);
90     visit(4);
91     return result;
92 }
93
94 /**
95  * visit()
96  * @return String
97  */
98 private String visit(String token)
99 {
100     result += token + " ";
101     return result;
102 }
103
104 /**
105  * inFixGeneration()
106  * @return String
107  */
108 private String inFixGeneration()
109 {
110     traverse();
111     return result;
112 }
113
114 /**
115  * traverse()
116  * @return String
117  */
118 private String traverse()
119 {
120     traverse(2);
121     visit(2);
122     return result;
123 }
124
125 /**
126  * visit()
127  * @return String
128  */
129 private String visit(String token)
130 {
131     result += token + " ";
132     return result;
133 }
134
135 /**
136  * inOrderTraversal()
137  * @return String
138  */
139 private String inOrderTraversal()
140 {
141     traverse();
142     return result;
143 }
144
145 /**
146  * traverse()
147  * @return String
148  */
149 private String traverse()
150 {
151     traverse(sqrt);
152     traverse(4);
153     visit(4);
154     return result;
155 }
156
157 /**
158  * visit()
159  * @return String
160  */
161 private String visit(String token)
162 {
163     result += token + " ";
164     return result;
165 }
166
167 /**
168  * inFixGeneration()
169  * @return String
170  */
171 private String inFixGeneration()
172 {
173     traverse();
174     return result;
175 }
176
177 /**
178  * traverse()
179  * @return String
180  */
181 private String traverse()
182 {
183     traverse(2);
184     visit(2);
185     return result;
186 }
187
188 /**
189  * visit()
190  * @return String
191  */
192 private String visit(String token)
193 {
194     result += token + " ";
195     return result;
196 }
197
198 /**
199  * inOrderTraversal()
200  * @return String
201  */
202 private String inOrderTraversal()
203 {
204     traverse();
205     return result;
206 }
207
208 /**
209  * traverse()
210  * @return String
211  */
212 private String traverse()
213 {
214     traverse(sqrt);
215     traverse(4);
216     visit(4);
217     return result;
218 }
219
220 /**
221  * visit()
222  * @return String
223  */
224 private String visit(String token)
225 {
226     result += token + " ";
227     return result;
228 }
229
230 /**
231  * inFixGeneration()
232  * @return String
233  */
234 private String inFixGeneration()
235 {
236     traverse();
237     return result;
238 }
239
240 /**
241  * traverse()
242  * @return String
243  */
244 private String traverse()
245 {
246     traverse(2);
247     visit(2);
248     return result;
249 }
250
251 /**
252  * visit()
253  * @return String
254  */
255 private String visit(String token)
256 {
257     result += token + " ";
258     return result;
259 }
260
261 /**
262  * inOrderTraversal()
263  * @return String
264  */
265 private String inOrderTraversal()
266 {
267     traverse();
268     return result;
269 }
270
271 /**
272  * traverse()
273  * @return String
274  */
275 private String traverse()
276 {
277     traverse(sqrt);
278     traverse(4);
279     visit(4);
280     return result;
281 }
282
283 /**
284  * visit()
285  * @return String
286  */
287 private String visit(String token)
288 {
289     result += token + " ";
290     return result;
291 }
292
293 /**
294  * inFixGeneration()
295  * @return String
296  */
297 private String inFixGeneration()
298 {
299     traverse();
300     return result;
301 }
302
303 /**
304  * traverse()
305  * @return String
306  */
307 private String traverse()
308 {
309     traverse(2);
310     visit(2);
311     return result;
312 }
313
314 /**
315  * visit()
316  * @return String
317  */
318 private String visit(String token)
319 {
320     result += token + " ";
321     return result;
322 }
323
324 /**
325  * inOrderTraversal()
326  * @return String
327  */
328 private String inOrderTraversal()
329 {
330     traverse();
331     return result;
332 }
333
334 /**
335  * traverse()
336  * @return String
337  */
338 private String traverse()
339 {
340     traverse(sqrt);
341     traverse(4);
342     visit(4);
343     return result;
344 }
345
346 /**
347  * visit()
348  * @return String
349  */
350 private String visit(String token)
351 {
352     result += token + " ";
353     return result;
354 }
355
356 /**
357  * inFixGeneration()
358  * @return String
359  */
360 private String inFixGeneration()
361 {
362     traverse();
363     return result;
364 }
365
366 /**
367  * traverse()
368  * @return String
369  */
370 private String traverse()
371 {
372     traverse(2);
373     visit(2);
374     return result;
375 }
376
377 /**
378  * visit()
379  * @return String
380  */
381 private String visit(String token)
382 {
383     result += token + " ";
384     return result;
385 }
386
387 /**
388  * inOrderTraversal()
389  * @return String
390  */
391 private String inOrderTraversal()
392 {
393     traverse();
394     return result;
395 }
396
397 /**
398  * traverse()
399  * @return String
400  */
401 private String traverse()
402 {
403     traverse(sqrt);
404     traverse(4);
405     visit(4);
406     return result;
407 }
408
409 /**
410  * visit()
411  * @return String
412  */
413 private String visit(String token)
414 {
415     result += token + " ";
416     return result;
417 }
418
419 /**
420  * inFixGeneration()
421  * @return String
422  */
423 private String inFixGeneration()
424 {
425     traverse();
426     return result;
427 }
428
429 /**
430  * traverse()
431  * @return String
432  */
433 private String traverse()
434 {
435     traverse(2);
436     visit(2);
437     return result;
438 }
439
440 /**
441  * visit()
442  * @return String
443  */
444 private String visit(String token)
445 {
446     result += token + " ";
447     return result;
448 }
449
450 /**
451  * inOrderTraversal()
452  * @return String
453  */
454 private String inOrderTraversal()
455 {
456     traverse();
457     return result;
458 }
459
460 /**
461  * traverse()
462  * @return String
463  */
464 private String traverse()
465 {
466     traverse(sqrt);
467     traverse(4);
468     visit(4);
469     return result;
470 }
471
472 /**
473  * visit()
474  * @return String
475  */
476 private String visit(String token)
477 {
478     result += token + " ";
479     return result;
480 }
481
482 /**
483  * inFixGeneration()
484  * @return String
485  */
486 private String inFixGeneration()
487 {
488     traverse();
489     return result;
490 }
491
492 /**
493  * traverse()
494  * @return String
495  */
496 private String traverse()
497 {
498     traverse(2);
499     visit(2);
500     return result;
501 }
502
503 /**
504  * visit()
505  * @return String
506  */
507 private String visit(String token)
508 {
509     result += token + " ";
510     return result;
511 }
512
513 /**
514  * inOrderTraversal()
515  * @return String
516  */
517 private String inOrderTraversal()
518 {
519     traverse();
520     return result;
521 }
522
523 /**
524  * traverse()
525  * @return String
526  */
527 private String traverse()
528 {
529     traverse(sqrt);
530     traverse(4);
531     visit(4);
532     return result;
533 }
534
535 /**
536  * visit()
537  * @return String
538  */
539 private String visit(String token)
540 {
541     result += token + " ";
542     return result;
543 }
544
545 /**
546  * inFixGeneration()
547  * @return String
548  */
549 private String inFixGeneration()
550 {
551     traverse();
552     return result;
553 }
554
555 /**
556  * traverse()
557  * @return String
558  */
559 private String traverse()
560 {
561     traverse(2);
562     visit(2);
563     return result;
564 }
565
566 /**
567  * visit()
568  * @return String
569  */
570 private String visit(String token)
571 {
572     result += token + " ";
573     return result;
574 }
575
576 /**
577  * inOrderTraversal()
578  * @return String
579  */
580 private String inOrderTraversal()
581 {
582     traverse();
583     return result;
584 }
585
586 /**
587  * traverse()
588  * @return String
589  */
590 private String traverse()
591 {
592     traverse(sqrt);
593     traverse(4);
594     visit(4);
595     return result;
596 }
597
598 /**
599  * visit()
600  * @return String
601  */
602 private String visit(String token)
603 {
604     result += token + " ";
605     return result;
606 }
607
608 /**
609  * inFixGeneration()
610  * @return String
611  */
612 private String inFixGeneration()
613 {
614     traverse();
615     return result;
616 }
617
618 /**
619  * traverse()
620  * @return String
621  */
622 private String traverse()
623 {
624     traverse(2);
625     visit(2);
626     return result;
627 }
628
629 /**
630  * visit()
631  * @return String
632  */
633 private String visit(String token)
634 {
635     result += token + " ";
636     return result;
637 }
638
639 /**
640  * inOrderTraversal()
641  * @return String
642  */
643 private String inOrderTraversal()
644 {
645     traverse();
646     return result;
647 }
648
649 /**
650  * traverse()
651  * @return String
652  */
653 private String traverse()
654 {
655     traverse(sqrt);
656     traverse(4);
657     visit(4);
658     return result;
659 }
660
661 /**
662  * visit()
663  * @return String
664  */
665 private String visit(String token)
666 {
667     result += token + " ";
668     return result;
669 }
670
671 /**
672  * inFixGeneration()
673  * @return String
674  */
675 private String inFixGeneration()
676 {
677     traverse();
678     return result;
679 }
680
681 /**
682  * traverse()
683  * @return String
684  */
685 private String traverse()
686 {
687     traverse(2);
688     visit(2);
689     return result;
690 }
691
692 /**
693  * visit()
694  * @return String
695  */
696 private String visit(String token)
697 {
698     result += token + " ";
699     return result;
700 }
701
702 /**
703  * inOrderTraversal()
704  * @return String
705  */
706 private String inOrderTraversal()
707 {
708     traverse();
709     return result;
710 }
711
712 /**
713  * traverse()
714  * @return String
715  */
716 private String traverse()
717 {
718     traverse(sqrt);
719     traverse(4);
720     visit(4);
721     return result;
722 }
723
724 /**
725  * visit()
726  * @return String
727  */
728 private String visit(String token)
729 {
730     result += token + " ";
731     return result;
732 }
733
734 /**
735  * inFixGeneration()
736  * @return String
737  */
738 private String inFixGeneration()
739 {
740     traverse();
741     return result;
742 }
743
744 /**
745  * traverse()
746  * @return String
747  */
748 private String traverse()
749 {
750     traverse(2);
751     visit(2);
752     return result;
753 }
754
755 /**
756  * visit()
757  * @return String
758  */
759 private String visit(String token)
760 {
761     result += token + " ";
762     return result;
763 }
764
765 /**
766  * inOrderTraversal()
767  * @return String
768  */
769 private String inOrderTraversal()
770 {
771     traverse();
772     return result;
773 }
774
775 /**
776  * traverse()
777  * @return String
778  */
779 private String traverse()
780 {
781     traverse(sqrt);
782     traverse(4);
783     visit(4);
784     return result;
785 }
786
787 /**
788  * visit()
789  * @return String
790  */
791 private String visit(String token)
792 {
793     result += token + " ";
794     return result;
795 }
796
797 /**
798  * inFixGeneration()
799  * @return String
800  */
801 private String inFixGeneration()
802 {
803     traverse();
804     return result;
805 }
806
807 /**
808  * traverse()
809  * @return String
810  */
811 private String traverse()
812 {
813     traverse(2);
814     visit(2);
815     return result;
816 }
817
818 /**
819  * visit()
820  * @return String
821  */
822 private String visit(String token)
823 {
824     result += token + " ";
825     return result;
826 }
827
828 /**
829  * inOrderTraversal()
830  * @return String
831  */
832 private String inOrderTraversal()
833 {
834     traverse();
835     return result;
836 }
837
838 /**
839  * traverse()
840  * @return String
841  */
842 private String traverse()
843 {
844     traverse(sqrt);
845     traverse(4);
846     visit(4);
847     return result;
848 }
849
850 /**
851  * visit()
852  * @return String
853  */
854 private String visit(String token)
855 {
856     result += token + " ";
857     return result;
858 }
859
860 /**
861  * inFixGeneration()
862  * @return String
863  */
864 private String inFixGeneration()
865 {
866     traverse();
867     return result;
868 }
869
870 /**
871  * traverse()
872  * @return String
873  */
874 private String traverse()
875 {
876     traverse(2);
877     visit(2);
878     return result;
879 }
880
881 /**
882  * visit()
883  * @return String
884  */
885 private String visit(String token)
886 {
887     result += token + " ";
888     return result;
889 }
890
891 /**
892  * inOrderTraversal()
893  * @return String
894  */
895 private String inOrderTraversal()
896 {
897     traverse();
898     return result;
899 }
900
901 /**
902  * traverse()
903  * @return String
904  */
905 private String traverse()
906 {
907     traverse(sqrt);
908     traverse(4);
909     visit(4);
910     return result;
911 }
912
913 /**
914  * visit()
915  * @return String
916  */
917 private String visit(String token)
918 {
919     result += token + " ";
920     return result;
921 }
922
923 /**
924  * inFixGeneration()
925  * @return String
926  */
927 private String inFixGeneration()
928 {
929     traverse();
930     return result;
931 }
932
933 /**
934  * traverse()
935  * @return String
936  */
937 private String traverse()
938 {
939     traverse(2);
940     visit(2);
941     return result;
942 }
943
944 /**
945  * visit()
946  * @return String
947  */
948 private String visit(String token)
949 {
950     result += token + " ";
951     return result;
952 }
953
954 /**
955  * inOrderTraversal()
956  * @return String
957  */
958 private String inOrderTraversal()
959 {
960     traverse();
961     return result;
962 }
963
964 /**
965  * traverse()
966  * @return String
967  */
968 private String traverse()
969 {
970     traverse(sqrt);
971     traverse(4);
972     visit(4);
973     return result;
974 }
975
976 /**
977  * visit()
978  * @return String
979  */
980 private String visit(String token)
981 {
982     result += token + " ";
983     return result;
984 }
985
986 /**
987  * inFixGeneration()
988  * @return String
989  */
990 private String inFixGeneration()
991 {
992     traverse();
993     return result;
994 }
995
996 /**
997  * traverse()
998  * @return String
999  */
1000 private String traverse()
1001 {
1002     traverse(2);
1003     visit(2);
1004     return result;
1005 }
1006
1007 /**
1008  * visit()
1009  * @return String
1010  */
1011 private String visit(String token)
1012 {
1013     result += token + " ";
1014     return result;
1015 }
1016
1017 /**
1018  * inOrderTraversal()
1019  * @return String
1020  */
1021 private String inOrderTraversal()
1022 {
1023     traverse();
1024     return result;
1025 }
1026
1027 /**
1028  * traverse()
1029  * @return String
1030  */
1031 private String traverse()
1032 {
1033     traverse(sqrt);
1034     traverse(4);
1035     visit(4);
1036     return result;
1037 }
1038
1039 /**
1040  * visit()
1041  * @return String
1042  */
1043 private String visit(String token)
1044 {
1045     result += token + " ";
1046     return result;
1047 }
1048
1049 /**
1050  * inFixGeneration()
1051  * @return String
1052  */
1053 private String inFixGeneration()
1054 {
1055     traverse();
1056     return result;
1057 }
1058
1059 /**
1060  * traverse()
1061  * @return String
1062  */
1063 private String traverse()
1064 {
1065     traverse(2);
1066     visit(2);
1067     return result;
1068 }
1069
1070 /**
1071  * visit()
1072  * @return String
1073  */
1074 private String visit(String token)
1075 {
1076     result += token + " ";
1077     return result;
1078 }
1079
1080 /**
1081  * inOrderTraversal()
1082  * @return String
1083  */
1084 private String inOrderTraversal()
1085 {
1086     traverse();
1087     return result;
1088 }
1089
1090 /**
1091  * traverse()
1092  * @return String
1093  */
1094 private String traverse()
1095 {
1096     traverse(sqrt);
1097     traverse(4);
1098     visit(4);
1099     return result;
1100 }
1101
1102 /**
1103  * visit()
1104  * @return String
1105  */
1106 private String visit(String token)
1107 {
1108     result += token + " ";
1109     return result;
1110 }
1111
1112 /**
1113  * inFixGeneration()
1114  * @return String
1115  */
1116 private String inFixGeneration()
1117 {
1118     traverse();
1119     return result;
1120 }
1121
1122 /**
1123  * traverse()
1124  * @return String
1125  */
1126 private String traverse()
1127 {
1128     traverse(2);
1129     visit(2);
1130     return result;
1131 }
1132
1133 /**
1134  * visit()
1135  * @return String
1136  */
1137 private String visit(String token)
1138 {
1139     result += token + " ";
1140     return result;
1141 }
1142
1143 /**
1144  * inOrderTraversal()
1145  * @return String
1146  */
1147 private String inOrderTraversal()
1148 {
1149     traverse();
1150     return result;
1151 }
1152
1153 /**
1154  * traverse()
1155  * @return String
1156  */
1157 private String traverse()
1158 {
1159     traverse(sqrt);
1160     traverse(4);
1161     visit(4);
1162     return result;
1163 }
1164
1165 /**
1166  * visit()
1167  * @return String
1168  */
1169 private String visit(String token)
1170 {
1171     result += token + " ";
1172     return result;
1173 }
1174
1175 /**
1176  * inFixGeneration()
1177  * @return String
1178  */
1179 private String inFixGeneration()
1180 {
1181     traverse();
1182     return result;
1183 }
1184
1185 /**
1186  * traverse()
1187  * @return String
1188  */
1189 private String traverse()
1190 {
1191     traverse(2);
1192     visit(2);
1193     return result;
1194 }
1195
1196 /**
1197  * visit()
1198  * @return String
1199  */
1200 private String visit(String token)
1201 {
1202     result += token + " ";
1203     return result;
1204 }
1205
1206 /**
1207  * inOrderTraversal()
1208  * @return String
1209  */
1210 private String inOrderTraversal()
1211 {
1212     traverse();
1213     return result;
1214 }
1215
1216 /**
1217  * traverse()
1218  * @return String
1219  */
1220 private String traverse()
1221 {
1222     traverse(sqrt);
1223     traverse(4);
1224     visit(4);
1225     return result;
1226 }
1227
1228 /**
1229  * visit()
1230  * @return String
1231  */
1232 private String visit(String token)
1233 {
1234     result += token + " ";
1235     return result;
1236 }
1237
1238 /**
1239  * inFixGeneration()
1240  * @return String
1241  */
1242 private String inFixGeneration()
1243 {
1244     traverse();
1245     return result;
1246 }
1247
1248 /**
1249  * traverse()
1250  * @return String
1251  */
1252 private String traverse()
1253 {
1254     traverse(2);
1255     visit(2);
1256     return result;
1257 }
1258
1259 /**
1260  * visit()
1261  * @return String
1262  */
1263 private String visit(String token)
1264 {
1265     result += token + " ";
1266     return result;
1267 }
1268
1269 /**
1270  * inOrderTraversal()
1271  * @return String
1272  */
1273 private String inOrderTraversal()
1274 {
1275     traverse();
1276     return result;
1277 }
1278
1279 /**
1280  * traverse()
1281  * @return String
1282  */
1283 private String traverse()
1284 {
1285     traverse(sqrt);
1286     traverse(4);
1287     visit(4);
1288     return result;
1289 }
1290
1291 /**
1292  * visit()
1293  * @return String
1294  */
1295 private String visit(String token)
1296 {
1297     result += token + " ";
1298     return result;
1299 }
1300
1301 /**
1302  * inFixGeneration()
1303  * @return String
1304  */
1305 private String inFixGeneration()
1306 {
1307     traverse();
1308     return result;
1309 }
1310
1311 /**
1312  * traverse()
1313  * @return String
1314  */
1315 private String traverse()
1316 {
1317     traverse(2);
1318     visit(2);
1319     return result;
1320 }
1321
1322 /**
1323  * visit()
1324  * @return String
1325  */
1326 private String visit(String token)
1327 {
1328     result += token + " ";
1329     return result;
1330 }
1331
1332 /**
1333  * inOrderTraversal()
1334  * @return String
1335  */
1336 private String inOrderTraversal()
1337 {
1338     traverse();
1339     return result;
1340 }
1341
1342 /**
1343  * traverse()
1344  * @return String
1345  */
1346 private String traverse()
1347 {
1348     traverse(sqrt);
1349     traverse(4);
1350     visit(4);
1351     return result;
1352 }
1353
1354 /**
1355  * visit()
1356  * @return String
1357  */
1358 private String visit(String token)
1359 {
1360     result += token + " ";
1361     return result;
1362 }
1363
1364 /**
1365  * inFixGeneration()
1366  * @return String
1367  */
1368 private String inFixGeneration()
1369 {
1370     traverse();
1371     return result;
1372 }
1373
1374 /**
1375  * traverse()
1376  * @return String
1377  */
1378 private String traverse()
1379 {
1380     traverse(2);
1381     visit(2);
1382     return result;
1383 }
1384
1385 /**
1386  * visit()
1387  * @return String
1388  */
1389 private String visit(String token)
1390 {
1391     result += token + " ";
1392     return result;
1393 }
1394
1395 /**
1396  * inOrderTraversal()
1397  * @return String
1398  */
1399 private String inOrderTraversal()
1400 {
1401     traverse();
1402     return result;
1403 }
1404
1405 /**
1406  * traverse()
1407  * @return String
1408  */
1409 private String traverse()
1410 {
1411     traverse(sqrt);
1412     traverse(4);
1413     visit(4);
1414     return result;
1415 }
1416
1417 /**
1418  * visit()
1419  * @return String
1420  */
1421 private String visit(String token)
1422 {
1423     result += token + " ";
1424     return result;
1425 }
1426
1427 /**
1428  * inFixGeneration()
1429  * @return String
1430  */
1431 private String inFixGeneration()
1432 {
1433     traverse();
1434     return result;
1435 }
1436
1437 /**
1438  * traverse()
1439  * @return String
1440  */
1441 private String traverse()
1442 {
1443     traverse(2);
1444     visit(2);
1445     return result;
1446 }
1447
1448 /**
1449  * visit()
1450  * @return String
1451  */
1452 private String visit(String token)
1453 {
1454     result += token + " ";
1455     return result;
1456 }
1457
1458 /**
1459  * inOrderTraversal()
1460  * @return String
1461  */
1462 private String inOrderTraversal()
1463 {
1464     traverse();
1465     return result;
1466 }
1467
1468 /**
1469  * traverse()
1470  * @return String
1471  */
1472 private String traverse()
1473 {
1474     traverse(sqrt);
1475     traverse(4);
1476     visit(4);
1477     return result;
1478 }
1479
1480 /**
1481  * visit()
1482  * @return String
1483  */
1484 private String visit(String token)
1485 {
1486     result += token + " ";
1487     return result;
1488 }
1489
1490 /**
1491  * inFixGeneration()
1492  * @return String
1493  */
1494 private String inFixGeneration()
1495 {
1496     traverse();
1497     return result;
1498 }
1499
1500 /**
1501  * traverse()
1502  * @return String
1503  */
1504 private String traverse()
1505 {
1506     traverse(2);
1507     visit(2);
1508     return result;
1509 }
1510
1511 /**
1512  * visit()
1513  * @return String
1514  */
1515 private String visit(String token)
1516 {
1517     result += token + " ";
1518     return result;
1519 }
1520
1521 /**
1522  * inOrderTraversal()
1523  * @return String
1524  */
1525 private String inOrderTraversal()
1526 {
1527     traverse();
1528     return result;
1529 }
1530
1531 /**
1532  * traverse()
1533  * @return String
1534  */
1535 private String traverse()
1536 {
1537     traverse(sqrt);
1538     traverse(4);
1539     visit(4);
1540     return result;
1541 }
1542
1543 /**
1544  * visit()
1545  * @return String
1546  */
1547 private String visit(String token)
1548 {
1549     result += token + " ";
1550     return result;
1551 }
1552
1553 /**
1554  * inFixGeneration()
1555  * @return String
1556  */
1557 private String inFixGeneration()
1558 {
1559     traverse();
1560     return result;
1561 }
1562
1563 /**
1564  * traverse()
1565  * @return String
1566  */
1567 private String traverse()
1568 {
1569     traverse(2);
1570     visit(2);
1571     return result;
1572 }
1573
1574 /**
1575  * visit()
1576  * @return String
1577  */
1578 private String
```

The check for correctness can be done by plugging in this expression that is in the result in the exact same way it is on the pocket calculator and checking the result. If you calculate it without a calculator, you will find out that the result is 8, which is the same result as plugging it on to the calculator.

LAB

1. Constructor from my partner's code was added. Please check the source code for more information.
2. These are the following test cases that were used to test the correctness of the code and all works fine.
 - a. $((\sin(4 * (\pi / 3))) * (\exp(-1 * ((\sqrt{2}) - 1) / 8))) / (\sqrt{6 * \pi}))$
 - b. $((\sqrt{4}) * (2^3)) / 2$
 - c. $(((((5^4)^2)^4) + 9) - 8) / 2$
 - d. $\exp((\sin(30) * 90) + (100 / 200))$
 - e. $2^{\exp((\sin(30) * 90) + (100 / 200))}$
 - f. $9 * (3^{(2^{\exp((\sin(30) * 90) + (100 / 200))})})$
3. The only case that the tree is incorrect is that the expression is incorrect and, in that case, the tree will throw an EmptyStackException mid-construction.