```java
1   /* Generated By:JavaCC: Do not edit this line. JavaCCParserTokenManager.java */
2   package jminusminus;
3   import java.util.ArrayList;
4
5   public class JavaCCParserTokenManager implements JavaCCParserConstants
6   {
7     public  java.io.PrintStream debugStream = System.out;
8     public  void setDebugStream(java.io.PrintStream ds) { debugStream = ds; }
9   private final int jjStopStringLiteralDfa_0(int pos, long active0)
10  {
11    switch (pos)
12    {
13       case 0:
14          if ((active0 & 0x7fffff80L) != 0L)
15          {
16             jjmatchedKind = 52;
17             return 7;
18          }
19          return -1;
20       case 1:
21          if ((active0 & 0x7fffbf80L) != 0L)
22          {
23             jjmatchedKind = 52;
24             jjmatchedPos = 1;
25             return 7;
26          }
27          if ((active0 & 0x44000L) != 0L)
28             return 7;
29          return -1;
30       case 2:
31          if ((active0 & 0x7ff9bf80L) != 0L)
32          {
33             jjmatchedKind = 52;
34             jjmatchedPos = 2;
35             return 7;
36          }
37          if ((active0 & 0x60000L) != 0L)
38             return 7;
39          return -1;
40       case 3:
41          if ((active0 & 0x38080a00L) != 0L)
42             return 7;
43          if ((active0 & 0x47f1b580L) != 0L)
44          {
45             jjmatchedKind = 52;
46             jjmatchedPos = 3;
47             return 7;
48          }
49          return -1;
50       case 4:
51          if ((active0 & 0x44002400L) != 0L)
52             return 7;
53          if ((active0 & 0x3f19180L) != 0L)
54          {
55             jjmatchedKind = 52;
56             jjmatchedPos = 4;
57             return 7;
58          }
59          return -1;
60       case 5:
61          if ((active0 & 0x3808000L) != 0L)
62             return 7;
63          if ((active0 & 0x711180L) != 0L)
64          {
65             jjmatchedKind = 52;
66             jjmatchedPos = 5;
```

```
67              return 7;
68          }
69          return -1;
70        case 6:
71          if ((active0 & 0x301100L) != 0L)
72              return 7;
73          if ((active0 & 0x410080L) != 0L)
74          {
75              jjmatchedKind = 52;
76              jjmatchedPos = 6;
77              return 7;
78          }
79          return -1;
80        case 7:
81          if ((active0 & 0x410000L) != 0L)
82          {
83              jjmatchedKind = 52;
84              jjmatchedPos = 7;
85              return 7;
86          }
87          if ((active0 & 0x80L) != 0L)
88              return 7;
89          return -1;
90        case 8:
91          if ((active0 & 0x10000L) != 0L)
92          {
93              jjmatchedKind = 52;
94              jjmatchedPos = 8;
95              return 7;
96          }
97          if ((active0 & 0x400000L) != 0L)
98              return 7;
99          return -1;
100       default :
101          return -1;
102    }
103 }
104 private final int jjStartNfa_0(int pos, long active0)
105 {
106    return jjMoveNfa_0(jjStopStringLiteralDfa_0(pos, active0), pos + 1);
107 }
108 private final int jjStopAtPos(int pos, int kind)
109 {
110    jjmatchedKind = kind;
111    jjmatchedPos = pos;
112    return pos + 1;
113 }
114 private final int jjStartNfaWithStates_0(int pos, int kind, int state)
115 {
116    jjmatchedKind = kind;
117    jjmatchedPos = pos;
118    try { curChar = input_stream.readChar(); }
119    catch(java.io.IOException e) { return pos + 1; }
120    return jjMoveNfa_0(state, pos + 1);
121 }
122 private final int jjMoveStringLiteralDfa0_0()
123 {
124    switch(curChar)
125    {
126       case 33:
127          return jjStopAtPos(0, 39);
128       case 38:
129          return jjMoveStringLiteralDfa1_0(0x2000000000L);
130       case 40:
131          return jjStopAtPos(0, 43);
132       case 41:
133          return jjStopAtPos(0, 44);
134       case 42:
135          return jjStopAtPos(0, 42);
```

```java
136          case 43:
137             jjmatchedKind = 31;
138             return jjMoveStringLiteralDfa1_0(0x21000000000L);
139          case 44:
140             return jjStopAtPos(0, 50);
141          case 45:
142             jjmatchedKind = 40;
143             return jjMoveStringLiteralDfa1_0(0x200000000L);
144          case 46:
145             return jjStopAtPos(0, 51);
146          case 59:
147             return jjStopAtPos(0, 49);
148          case 60:
149             return jjMoveStringLiteralDfa1_0(0x4000000000L);
150          case 61:
151             jjmatchedKind = 32;
152             return jjMoveStringLiteralDfa1_0(0x400000000L);
153          case 62:
154             return jjStopAtPos(0, 35);
155          case 91:
156             return jjStopAtPos(0, 47);
157          case 93:
158             return jjStopAtPos(0, 48);
159          case 97:
160             return jjMoveStringLiteralDfa1_0(0x80L);
161          case 98:
162             return jjMoveStringLiteralDfa1_0(0x100L);
163          case 99:
164             return jjMoveStringLiteralDfa1_0(0x600L);
165          case 101:
166             return jjMoveStringLiteralDfa1_0(0x1800L);
167          case 102:
168             return jjMoveStringLiteralDfa1_0(0x2000L);
169          case 105:
170             return jjMoveStringLiteralDfa1_0(0xc000L);
171          case 110:
172             return jjMoveStringLiteralDfa1_0(0xc0000L);
173          case 112:
174             return jjMoveStringLiteralDfa1_0(0xf00000L);
175          case 114:
176             return jjMoveStringLiteralDfa1_0(0x1000000L);
177          case 115:
178             return jjMoveStringLiteralDfa1_0(0x6000000L);
179          case 116:
180             return jjMoveStringLiteralDfa1_0(0x18000000L);
181          case 118:
182             return jjMoveStringLiteralDfa1_0(0x20000000L);
183          case 119:
184             return jjMoveStringLiteralDfa1_0(0x40000000L);
185          case 123:
186             return jjStopAtPos(0, 45);
187          case 125:
188             return jjStopAtPos(0, 46);
189          default :
190             return jjMoveNfa_0(5, 0);
191       }
192 }
193 private final int jjMoveStringLiteralDfa1_0(long active0)
194 {
195    try { curChar = input_stream.readChar(); }
196    catch(java.io.IOException e) {
197       jjStopStringLiteralDfa_0(0, active0);
198       return 1;
199    }
200    switch(curChar)
201    {
202       case 38:
203          if ((active0 & 0x2000000000L) != 0L)
204             return jjStopAtPos(1, 37);
```

```java
205            break;
206         case 43:
207            if ((active0 & 0x1000000000L) != 0L)
208               return jjStopAtPos(1, 36);
209            break;
210         case 45:
211            if ((active0 & 0x200000000L) != 0L)
212               return jjStopAtPos(1, 33);
213            break;
214         case 61:
215            if ((active0 & 0x400000000L) != 0L)
216               return jjStopAtPos(1, 34);
217            else if ((active0 & 0x4000000000L) != 0L)
218               return jjStopAtPos(1, 38);
219            else if ((active0 & 0x20000000000L) != 0L)
220               return jjStopAtPos(1, 41);
221            break;
222         case 97:
223            return jjMoveStringLiteralDfa2_0(active0, 0x102000L);
224         case 98:
225            return jjMoveStringLiteralDfa2_0(active0, 0x80L);
226         case 101:
227            return jjMoveStringLiteralDfa2_0(active0, 0x1040000L);
228         case 102:
229            if ((active0 & 0x4000L) != 0L)
230               return jjStartNfaWithStates_0(1, 14, 7);
231            break;
232         case 104:
233            return jjMoveStringLiteralDfa2_0(active0, 0x48000200L);
234         case 108:
235            return jjMoveStringLiteralDfa2_0(active0, 0xc00L);
236         case 109:
237            return jjMoveStringLiteralDfa2_0(active0, 0x8000L);
238         case 110:
239            return jjMoveStringLiteralDfa2_0(active0, 0x80000L);
240         case 111:
241            return jjMoveStringLiteralDfa2_0(active0, 0x20000100L);
242         case 114:
243            return jjMoveStringLiteralDfa2_0(active0, 0x10600000L);
244         case 116:
245            return jjMoveStringLiteralDfa2_0(active0, 0x2000000L);
246         case 117:
247            return jjMoveStringLiteralDfa2_0(active0, 0x4880000L);
248         case 120:
249            return jjMoveStringLiteralDfa2_0(active0, 0x1000L);
250         default :
251            break;
252      }
253      return jjStartNfa_0(0, active0);
254 }
255 private final int jjMoveStringLiteralDfa2_0(long old0, long active0)
256 {
257      if (((active0 &= old0)) == 0L)
258         return jjStartNfa_0(0, old0);
259      try { curChar = input_stream.readChar(); }
260      catch(java.io.IOException e) {
261         jjStopStringLiteralDfa_0(1, active0);
262         return 2;
263      }
264      switch(curChar)
265      {
266         case 97:
267            return jjMoveStringLiteralDfa3_0(active0, 0x2000600L);
268         case 98:
269            return jjMoveStringLiteralDfa3_0(active0, 0x800000L);
270         case 99:
271            return jjMoveStringLiteralDfa3_0(active0, 0x100000L);
272         case 105:
273            return jjMoveStringLiteralDfa3_0(active0, 0x68200000L);
```

```java
274          case 108:
275             return jjMoveStringLiteralDfa3_0(active0, 0x82000L);
276          case 111:
277             return jjMoveStringLiteralDfa3_0(active0, 0x400100L);
278          case 112:
279             return jjMoveStringLiteralDfa3_0(active0, 0x4008000L);
280          case 115:
281             return jjMoveStringLiteralDfa3_0(active0, 0x10880L);
282          case 116:
283             if ((active0 & 0x20000L) != 0L)
284                return jjStartNfaWithStates_0(2, 17, 7);
285             return jjMoveStringLiteralDfa3_0(active0, 0x1001000L);
286          case 117:
287             return jjMoveStringLiteralDfa3_0(active0, 0x10000000L);
288          case 119:
289             if ((active0 & 0x40000L) != 0L)
290                return jjStartNfaWithStates_0(2, 18, 7);
291             break;
292          default :
293             break;
294       }
295       return jjStartNfa_0(1, active0);
296 }
297 private final int jjMoveStringLiteralDfa3_0(long old0, long active0)
298 {
299    if (((active0 &= old0)) == 0L)
300       return jjStartNfa_0(1, old0);
301    try { curChar = input_stream.readChar(); }
302    catch(java.io.IOException e) {
303       jjStopStringLiteralDfa_0(2, active0);
304       return 3;
305    }
306    switch(curChar)
307    {
308       case 100:
309          if ((active0 & 0x20000000L) != 0L)
310             return jjStartNfaWithStates_0(3, 29, 7);
311          break;
312       case 101:
313          if ((active0 & 0x800L) != 0L)
314             return jjStartNfaWithStates_0(3, 11, 7);
315          else if ((active0 & 0x10000000L) != 0L)
316             return jjStartNfaWithStates_0(3, 28, 7);
317          return jjMoveStringLiteralDfa4_0(active0, 0x4001000L);
318       case 107:
319          return jjMoveStringLiteralDfa4_0(active0, 0x100000L);
320       case 108:
321          if ((active0 & 0x80000L) != 0L)
322             return jjStartNfaWithStates_0(3, 19, 7);
323          return jjMoveStringLiteralDfa4_0(active0, 0x40800100L);
324       case 111:
325          return jjMoveStringLiteralDfa4_0(active0, 0x8000L);
326       case 114:
327          if ((active0 & 0x200L) != 0L)
328             return jjStartNfaWithStates_0(3, 9, 7);
329          break;
330       case 115:
331          if ((active0 & 0x8000000L) != 0L)
332             return jjStartNfaWithStates_0(3, 27, 7);
333          return jjMoveStringLiteralDfa4_0(active0, 0x2400L);
334       case 116:
335          return jjMoveStringLiteralDfa4_0(active0, 0x2410080L);
336       case 117:
337          return jjMoveStringLiteralDfa4_0(active0, 0x1000000L);
338       case 118:
339          return jjMoveStringLiteralDfa4_0(active0, 0x200000L);
340       default :
341          break;
342    }
```

```java
343          return jjStartNfa_0(2, active0);
344 }
345 private final int jjMoveStringLiteralDfa4_0(long old0, long active0)
346 {
347     if (((active0 &= old0)) == 0L)
348         return jjStartNfa_0(2, old0);
349     try { curChar = input_stream.readChar(); }
350     catch(java.io.IOException e) {
351         jjStopStringLiteralDfa_0(3, active0);
352         return 4;
353     }
354     switch(curChar)
355     {
356         case 97:
357             return jjMoveStringLiteralDfa5_0(active0, 0x310000L);
358         case 101:
359             if ((active0 & 0x2000L) != 0L)
360                 return jjStartNfaWithStates_0(4, 13, 7);
361             else if ((active0 & 0x40000000L) != 0L)
362                 return jjStartNfaWithStates_0(4, 30, 7);
363             return jjMoveStringLiteralDfa5_0(active0, 0x400100L);
364         case 105:
365             return jjMoveStringLiteralDfa5_0(active0, 0x2800000L);
366         case 110:
367             return jjMoveStringLiteralDfa5_0(active0, 0x1000L);
368         case 114:
369             if ((active0 & 0x4000000L) != 0L)
370                 return jjStartNfaWithStates_0(4, 26, 7);
371             return jjMoveStringLiteralDfa5_0(active0, 0x1008080L);
372         case 115:
373             if ((active0 & 0x400L) != 0L)
374                 return jjStartNfaWithStates_0(4, 10, 7);
375             break;
376         default :
377             break;
378     }
379     return jjStartNfa_0(3, active0);
380 }
381 private final int jjMoveStringLiteralDfa5_0(long old0, long active0)
382 {
383     if (((active0 &= old0)) == 0L)
384         return jjStartNfa_0(3, old0);
385     try { curChar = input_stream.readChar(); }
386     catch(java.io.IOException e) {
387         jjStopStringLiteralDfa_0(4, active0);
388         return 5;
389     }
390     switch(curChar)
391     {
392         case 97:
393             return jjMoveStringLiteralDfa6_0(active0, 0x180L);
394         case 99:
395             if ((active0 & 0x800000L) != 0L)
396                 return jjStartNfaWithStates_0(5, 23, 7);
397             else if ((active0 & 0x2000000L) != 0L)
398                 return jjStartNfaWithStates_0(5, 25, 7);
399             return jjMoveStringLiteralDfa6_0(active0, 0x400000L);
400         case 100:
401             return jjMoveStringLiteralDfa6_0(active0, 0x1000L);
402         case 103:
403             return jjMoveStringLiteralDfa6_0(active0, 0x100000L);
404         case 110:
405             if ((active0 & 0x1000000L) != 0L)
406                 return jjStartNfaWithStates_0(5, 24, 7);
407             return jjMoveStringLiteralDfa6_0(active0, 0x10000L);
408         case 116:
409             if ((active0 & 0x8000L) != 0L)
410                 return jjStartNfaWithStates_0(5, 15, 7);
411             return jjMoveStringLiteralDfa6_0(active0, 0x200000L);
```

```java
412            default :
413               break;
414         }
415         return jjStartNfa_0(4, active0);
416 }
417 private final int jjMoveStringLiteralDfa6_0(long old0, long active0)
418 {
419         if (((active0 &= old0)) == 0L)
420            return jjStartNfa_0(4, old0);
421         try { curChar = input_stream.readChar(); }
422         catch(java.io.IOException e) {
423            jjStopStringLiteralDfa_0(5, active0);
424            return 6;
425         }
426         switch(curChar)
427         {
428            case 99:
429               return jjMoveStringLiteralDfa7_0(active0, 0x10080L);
430            case 101:
431               if ((active0 & 0x100000L) != 0L)
432                  return jjStartNfaWithStates_0(6, 20, 7);
433               else if ((active0 & 0x200000L) != 0L)
434                  return jjStartNfaWithStates_0(6, 21, 7);
435               break;
436            case 110:
437               if ((active0 & 0x100L) != 0L)
438                  return jjStartNfaWithStates_0(6, 8, 7);
439               break;
440            case 115:
441               if ((active0 & 0x1000L) != 0L)
442                  return jjStartNfaWithStates_0(6, 12, 7);
443               break;
444            case 116:
445               return jjMoveStringLiteralDfa7_0(active0, 0x400000L);
446            default :
447               break;
448         }
449         return jjStartNfa_0(5, active0);
450 }
451 private final int jjMoveStringLiteralDfa7_0(long old0, long active0)
452 {
453         if (((active0 &= old0)) == 0L)
454            return jjStartNfa_0(5, old0);
455         try { curChar = input_stream.readChar(); }
456         catch(java.io.IOException e) {
457            jjStopStringLiteralDfa_0(6, active0);
458            return 7;
459         }
460         switch(curChar)
461         {
462            case 101:
463               return jjMoveStringLiteralDfa8_0(active0, 0x410000L);
464            case 116:
465               if ((active0 & 0x80L) != 0L)
466                  return jjStartNfaWithStates_0(7, 7, 7);
467               break;
468            default :
469               break;
470         }
471         return jjStartNfa_0(6, active0);
472 }
473 private final int jjMoveStringLiteralDfa8_0(long old0, long active0)
474 {
475         if (((active0 &= old0)) == 0L)
476            return jjStartNfa_0(6, old0);
477         try { curChar = input_stream.readChar(); }
478         catch(java.io.IOException e) {
479            jjStopStringLiteralDfa_0(7, active0);
480            return 8;
```

```java
481        }
482        switch(curChar)
483        {
484           case 100:
485              if ((active0 & 0x400000L) != 0L)
486                 return jjStartNfaWithStates_0(8, 22, 7);
487              break;
488           case 111:
489              return jjMoveStringLiteralDfa9_0(active0, 0x10000L);
490           default :
491              break;
492        }
493        return jjStartNfa_0(7, active0);
494  }
495  private final int jjMoveStringLiteralDfa9_0(long old0, long active0)
496  {
497        if (((active0 &= old0)) == 0L)
498           return jjStartNfa_0(7, old0);
499        try { curChar = input_stream.readChar(); }
500        catch(java.io.IOException e) {
501           jjStopStringLiteralDfa_0(8, active0);
502           return 9;
503        }
504        switch(curChar)
505        {
506           case 102:
507              if ((active0 & 0x10000L) != 0L)
508                 return jjStartNfaWithStates_0(9, 16, 7);
509              break;
510           default :
511              break;
512        }
513        return jjStartNfa_0(8, active0);
514  }
515  private final void jjCheckNAdd(int state)
516  {
517        if (jjrounds[state] != jjround)
518        {
519           jjstateSet[jjnewStateCnt++] = state;
520           jjrounds[state] = jjround;
521        }
522  }
523  private final void jjAddStates(int start, int end)
524  {
525        do {
526           jjstateSet[jjnewStateCnt++] = jjnextStates[start];
527        } while (start++ != end);
528  }
529  private final void jjCheckNAddTwoStates(int state1, int state2)
530  {
531        jjCheckNAdd(state1);
532        jjCheckNAdd(state2);
533  }
534  private final void jjCheckNAddStates(int start, int end)
535  {
536        do {
537           jjCheckNAdd(jjnextStates[start]);
538        } while (start++ != end);
539  }
540  private final void jjCheckNAddStates(int start)
541  {
542        jjCheckNAdd(jjnextStates[start]);
543        jjCheckNAdd(jjnextStates[start + 1]);
544  }
545  static final long[] jjbitVec0 = {
546        0x0L, 0x0L, 0xffffffffffffffffL, 0xffffffffffffffffL
547  };
548  private final int jjMoveNfa_0(int startState, int curPos)
549  {
```

```
550    int[] nextStates;
551    int startsAt = 0;
552    jjnewStateCnt = 21;
553    int i = 1;
554    jjstateSet[0] = startState;
555    int j, kind = 0x7fffffff;
556    for (;;)
557    {
558       if (++jjround == 0x7fffffff)
559          ReInitRounds();
560       if (curChar < 64)
561       {
562          long l = 1L << curChar;
563          MatchLoop: do
564          {
565             switch(jjstateSet[--i])
566             {
567                case 5:
568                   if ((0x3fe000000000000L & l) != 0L)
569                   {
570                      if (kind > 56)
571                         kind = 56;
572                      jjCheckNAdd(10);
573                   }
574                   else if (curChar == 34)
575                      jjCheckNAddStates(0, 2);
576                   else if (curChar == 39)
577                      jjAddStates(3, 4);
578                   else if (curChar == 48)
579                   {
580                      if (kind > 56)
581                         kind = 56;
582                   }
583                   else if (curChar == 36)
584                   {
585                      if (kind > 52)
586                         kind = 52;
587                      jjCheckNAdd(7);
588                   }
589                   else if (curChar == 47)
590                      jjstateSet[jjnewStateCnt++] = 0;
591                   break;
592                case 0:
593                   if (curChar == 47)
594                      jjCheckNAddStates(5, 7);
595                   break;
596                case 1:
597                   if ((0xffffffffffffdbffL & l) != 0L)
598                      jjCheckNAddStates(5, 7);
599                   break;
600                case 2:
601                   if ((0x2400L & l) != 0L && kind > 6)
602                      kind = 6;
603                   break;
604                case 3:
605                   if (curChar == 10 && kind > 6)
606                      kind = 6;
607                   break;
608                case 4:
609                   if (curChar == 13)
610                      jjstateSet[jjnewStateCnt++] = 3;
611                   break;
612                case 6:
613                   if (curChar != 36)
614                      break;
615                   if (kind > 52)
616                      kind = 52;
617                   jjCheckNAdd(7);
618                   break;
```

```java
                   case 7:
                      if ((0x3ff001000000000L & l) == 0L)
                         break;
                      if (kind > 52)
                         kind = 52;
                      jjCheckNAdd(7);
                      break;
                   case 8:
                      if (curChar == 48 && kind > 56)
                         kind = 56;
                      break;
                   case 9:
                      if ((0x3fe000000000000L & l) == 0L)
                         break;
                      if (kind > 56)
                         kind = 56;
                      jjCheckNAdd(10);
                      break;
                   case 10:
                      if ((0x3ff000000000000L & l) == 0L)
                         break;
                      if (kind > 56)
                         kind = 56;
                      jjCheckNAdd(10);
                      break;
                   case 11:
                      if (curChar == 39)
                         jjAddStates(3, 4);
                      break;
                   case 13:
                      if ((0x8400000000L & l) != 0L)
                         jjCheckNAdd(14);
                      break;
                   case 14:
                      if (curChar == 80 && kind > 57)
                         kind = 57;
                      break;
                   case 15:
                      if ((0xffffffff7fffffdbffL & l) != 0L)
                         jjCheckNAdd(14);
                      break;
                   case 16:
                      if (curChar == 34)
                         jjCheckNAddStates(0, 2);
                      break;
                   case 18:
                      if ((0x8400000000L & l) != 0L)
                         jjCheckNAddStates(0, 2);
                      break;
                   case 19:
                      if ((0xffffffffbffffdbffL & l) != 0L)
                         jjCheckNAddStates(0, 2);
                      break;
                   case 20:
                      if (curChar == 34 && kind > 58)
                         kind = 58;
                      break;
                   default : break;
                }
             } while(i != startsAt);
          }
          else if (curChar < 128)
          {
             long l = 1L << (curChar & 077);
             MatchLoop: do
             {
                switch(jjstateSet[--i])
                {
                   case 5:
```

```
688              case 7:
689                 if ((0x7fffffe87fffffeL & l) == 0L)
690                    break;
691                 if (kind > 52)
692                    kind = 52;
693                 jjCheckNAdd(7);
694                 break;
695              case 1:
696                 jjAddStates(5, 7);
697                 break;
698              case 12:
699                 if (curChar == 92)
700                    jjstateSet[jjnewStateCnt++] = 13;
701                 break;
702              case 13:
703                 if ((0x14404410000000L & l) != 0L)
704                    jjCheckNAdd(14);
705                 break;
706              case 15:
707                 if ((0xfffffffffeffffffffL & l) != 0L)
708                    jjCheckNAdd(14);
709                 break;
710              case 17:
711                 if (curChar == 92)
712                    jjstateSet[jjnewStateCnt++] = 18;
713                 break;
714              case 18:
715                 if ((0x14404410000000L & l) != 0L)
716                    jjCheckNAddStates(0, 2);
717                 break;
718              case 19:
719                 if ((0xfffffffffeffffffffL & l) != 0L)
720                    jjCheckNAddStates(0, 2);
721                 break;
722              default : break;
723           }
724        } while(i != startsAt);
725     }
726     else
727     {
728        int i2 = (curChar & 0xff) >> 6;
729        long l2 = 1L << (curChar & 077);
730        MatchLoop: do
731        {
732           switch(jjstateSet[--i])
733           {
734              case 1:
735                 if ((jjbitVec0[i2] & l2) != 0L)
736                    jjAddStates(5, 7);
737                 break;
738              case 15:
739                 if ((jjbitVec0[i2] & l2) != 0L)
740                    jjstateSet[jjnewStateCnt++] = 14;
741                 break;
742              case 19:
743                 if ((jjbitVec0[i2] & l2) != 0L)
744                    jjAddStates(0, 2);
745                 break;
746              default : break;
747           }
748        } while(i != startsAt);
749     }
750     if (kind != 0x7fffffff)
751     {
752        jjmatchedKind = kind;
753        jjmatchedPos = curPos;
754        kind = 0x7fffffff;
755     }
756     ++curPos;
```

```java
757        if ((i = jjnewStateCnt) == (startsAt = 21 - (jjnewStateCnt = startsAt)))
758           return curPos;
759        try { curChar = input_stream.readChar(); }
760        catch(java.io.IOException e) { return curPos; }
761     }
762 }
763 static final int[] jjnextStates = {
764    17, 19, 20, 12, 15, 1, 2, 4,
765 };
766 public static final String[] jjstrLiteralImages = {
767 "", null, null, null, null, null, null, "\141\142\163\164\162\141\143\164",
768 "\142\157\157\154\145\141\156", "\143\150\141\162", "\143\154\141\163\163", "\
145\154\163\145",
769 "\145\170\164\145\156\144\163", "\146\141\154\163\145", "\151\146", "\
151\155\160\157\162\164",
770 "\151\156\163\164\141\156\143\145\157\146", "\151\156\164", "\156\145\167", "\
156\165\154\154",
771 "\160\141\143\153\141\147\145", "\160\162\151\166\141\164\145", "\
160\162\157\164\145\143\164\145\144",
772 "\160\165\142\154\151\143", "\162\145\164\165\162\156", "\
163\164\141\164\151\143",
773 "\163\165\160\145\162", "\164\150\151\163", "\164\162\165\145", "\
166\157\151\144",
774 "\167\150\151\154\145", "\53", "\75", "\55\55", "\75\75", "\76", "\53\53", "\
46\46", "\74\75", "\41",
775 "\55", "\53\75", "\52", "\50", "\51", "\173", "\175", "\133", "\135", "\73", "\
54",
776 "\56", null, null, null, null, null, null, null, null, null, };
777 public static final String[] lexStateNames = {
778    "DEFAULT",
779 };
780 static final long[] jjtoToken = {
781    0x171ffffffffff81L,
782 };
783 static final long[] jjtoSkip = {
784    0x7eL,
785 };
786 protected SimpleCharStream input_stream;
787 private final int[] jjrounds = new int[21];
788 private final int[] jjstateSet = new int[42];
789 protected char curChar;
790 public JavaCCParserTokenManager(SimpleCharStream stream){
791    if (SimpleCharStream.staticFlag)
792       throw new Error("ERROR: Cannot use a static CharStream class with a non-
static lexical analyzer.");
793    input_stream = stream;
794 }
795 public JavaCCParserTokenManager(SimpleCharStream stream, int lexState){
796    this(stream);
797    SwitchTo(lexState);
798 }
799 public void ReInit(SimpleCharStream stream)
800 {
801    jjmatchedPos = jjnewStateCnt = 0;
802    curLexState = defaultLexState;
803    input_stream = stream;
804    ReInitRounds();
805 }
806 private final void ReInitRounds()
807 {
808    int i;
809    jjround = 0x80000001;
810    for (i = 21; i-- > 0;)
811       jjrounds[i] = 0x80000000;
812 }
813 public void ReInit(SimpleCharStream stream, int lexState)
814 {
815    ReInit(stream);
816    SwitchTo(lexState);
```

```java
817  }
818  public void SwitchTo(int lexState)
819  {
820     if (lexState >= 1 || lexState < 0)
821        throw new TokenMgrError("Error: Ignoring invalid lexical state : " +
lexState + ". State unchanged.", TokenMgrError.INVALID_LEXICAL_STATE);
822     else
823        curLexState = lexState;
824  }

825
826  protected Token jjFillToken()
827  {
828     Token t = Token.newToken(jjmatchedKind);
829     t.kind = jjmatchedKind;
830     String im = jjstrLiteralImages[jjmatchedKind];
831     t.image = (im == null) ? input_stream.GetImage() : im;
832     t.beginLine = input_stream.getBeginLine();
833     t.beginColumn = input_stream.getBeginColumn();
834     t.endLine = input_stream.getEndLine();
835     t.endColumn = input_stream.getEndColumn();
836     return t;
837  }

838
839  int curLexState = 0;
840  int defaultLexState = 0;
841  int jjnewStateCnt;
842  int jjround;
843  int jjmatchedPos;
844  int jjmatchedKind;

845
846  public Token getNextToken()
847  {
848    int kind;
849    Token specialToken = null;
850    Token matchedToken;
851    int curPos = 0;

852
853    EOFLoop :
854    for (;;)
855    {
856     try
857     {
858        curChar = input_stream.BeginToken();
859     }
860     catch(java.io.IOException e)
861     {
862        jjmatchedKind = 0;
863        matchedToken = jjFillToken();
864        return matchedToken;
865     }

866
867     try { input_stream.backup(0);
868        while (curChar <= 32 && (0x100003600L & (1L << curChar)) != 0L)
869           curChar = input_stream.BeginToken();
870     }
871     catch (java.io.IOException e1) { continue EOFLoop; }
872     jjmatchedKind = 0x7fffffff;
873     jjmatchedPos = 0;
874     curPos = jjMoveStringLiteralDfa0_0();
875     if (jjmatchedPos == 0 && jjmatchedKind > 60)
876     {
877        jjmatchedKind = 60;
878     }
879     if (jjmatchedKind != 0x7fffffff)
880     {
881        if (jjmatchedPos + 1 < curPos)
882           input_stream.backup(curPos - jjmatchedPos - 1);
883        if ((jjtoToken[jjmatchedKind >> 6] & (1L << (jjmatchedKind & 077))) != 0L)
884        {
```

```java
885         matchedToken = jjFillToken();
886         return matchedToken;
887       }
888       else
889       {
890         continue EOFLoop;
891       }
892     }
893     int error_line = input_stream.getEndLine();
894     int error_column = input_stream.getEndColumn();
895     String error_after = null;
896     boolean EOFSeen = false;
897     try { input_stream.readChar(); input_stream.backup(1); }
898     catch (java.io.IOException e1) {
899       EOFSeen = true;
900       error_after = curPos <= 1 ? "" : input_stream.GetImage();
901       if (curChar == '\n' || curChar == '\r') {
902         error_line++;
903         error_column = 0;
904       }
905       else
906         error_column++;
907     }
908     if (!EOFSeen) {
909       input_stream.backup(1);
910       error_after = curPos <= 1 ? "" : input_stream.GetImage();
911     }
912     throw new TokenMgrError(EOFSeen, curLexState, error_line, error_column,
error_after, curChar, TokenMgrError.LEXICAL_ERROR);
913   }
914 }
915
916 }
917
```