

## SimpleCharStream.java

```
1  /* Generated By:JavaCC: Do not edit this line. SimpleCharStream.java Version 4.0
2  */
3
4  package jminusminus;
5
6  /**
7   * An implementation of interface CharStream, where the stream is assumed to
8   * contain only ASCII characters (without unicode processing).
9   */
10
11 public class SimpleCharStream
12 {
13     public static final boolean staticFlag = false;
14     int bufsize;
15     int available;
16     int tokenBegin;
17     public int bufpos = -1;
18     protected int bufline[];
19     protected int bufcolumn[];
20
21     protected int column = 0;
22     protected int line = 1;
23
24     protected boolean prevCharIsCR = false;
25     protected boolean prevCharIsLF = false;
26
27     protected java.io.Reader inputStream;
28     protected char[] buffer;
29     protected int maxNextCharInd = 0;
30     protected int inBuf = 0;
31     protected int tabSize = 8;
32
33     protected void setTabSize(int i) { tabSize = i; }
34     protected int getTabSize(int i) { return tabSize; }
35
36     protected void ExpandBuff(boolean wrapAround)
37     {
38         char[] newbuffer = new char[bufsize + 2048];
39         int newbufline[] = new int[bufsize + 2048];
40         int newbufcolumn[] = new int[bufsize + 2048];
41
42         try
43         {
44             if (wrapAround)
45             {
46                 System.arraycopy(buffer, tokenBegin, newbuffer, 0, bufsize -
47 tokenBegin);
48                 System.arraycopy(buffer, 0, newbuffer,
49                               bufsize - tokenBegin, bufpos);
50                 buffer = newbuffer;
51                 System.arraycopy(bufline, tokenBegin, newbufline, 0, bufsize -
52 tokenBegin);
53                 System.arraycopy(bufline, 0, newbufline, bufsize - tokenBegin,
54 bufpos);
55                 bufline = newbufline;
56                 System.arraycopy(bufcolumn, tokenBegin, newbufcolumn, 0, bufsize -
57 tokenBegin);
58                 System.arraycopy(bufcolumn, 0, newbufcolumn, bufsize - tokenBegin,
59 bufpos);
60                 bufcolumn = newbufcolumn;
61                 maxNextCharInd = (bufpos += (bufsize - tokenBegin));
62             }
63         }
64     }
65 }
```

```

61         else
62         {
63             System.arraycopy(buffer, tokenBegin, newbuffer, 0, bufsize -
tokenBegin);
64             buffer = newbuffer;
65
66             System.arraycopy(bufline, tokenBegin, newbufline, 0, bufsize -
tokenBegin);
67             bufline = newbufline;
68
69             System.arraycopy(bufcolumn, tokenBegin, newbufcolumn, 0, bufsize -
tokenBegin);
70             bufcolumn = newbufcolumn;
71
72             maxNextCharInd = (bufpos -= tokenBegin);
73         }
74     }
75     catch (Throwable t)
76     {
77         throw new Error(t.getMessage());
78     }
79
80
81     bufsize += 2048;
82     available = bufsize;
83     tokenBegin = 0;
84 }
85
86 protected void FillBuff() throws java.io.IOException
87 {
88     if (maxNextCharInd == available)
89     {
90         if (available == bufsize)
91         {
92             if (tokenBegin > 2048)
93             {
94                 bufpos = maxNextCharInd = 0;
95                 available = tokenBegin;
96             }
97             else if (tokenBegin < 0)
98                 bufpos = maxNextCharInd = 0;
99             else
100                 ExpandBuff(false);
101         }
102         else if (available > tokenBegin)
103             available = bufsize;
104         else if ((tokenBegin - available) < 2048)
105             ExpandBuff(true);
106         else
107             available = tokenBegin;
108     }
109
110     int i;
111     try {
112         if ((i = inputStream.read(buffer, maxNextCharInd,
available - maxNextCharInd)) == -1)
113         {
114             inputStream.close();
115             throw new java.io.IOException();
116         }
117         else
118             maxNextCharInd += i;
119         return;
120     }
121     catch (java.io.IOException e) {
122         --bufpos;
123         backup(0);
124         if (tokenBegin == -1)
125             tokenBegin = bufpos;
126     }

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

127         throw e;
128     }
129 }
130
131 public char BeginToken() throws java.io.IOException
132 {
133     tokenBegin = -1;
134     char c = readChar();
135     tokenBegin = bufpos;
136
137     return c;
138 }
139
140 protected void UpdateLineColumn(char c)
141 {
142     column++;
143
144     if (prevCharIsLF)
145     {
146         prevCharIsLF = false;
147         line += (column = 1);
148     }
149     else if (prevCharIsCR)
150     {
151         prevCharIsCR = false;
152         if (c == '\n')
153         {
154             prevCharIsLF = true;
155         }
156         else
157             line += (column = 1);
158     }
159
160     switch (c)
161     {
162         case '\r' :
163             prevCharIsCR = true;
164             break;
165         case '\n' :
166             prevCharIsLF = true;
167             break;
168         case '\t' :
169             column--;
170             column += (tabSize - (column % tabSize));
171             break;
172         default :
173             break;
174     }
175
176     bufline[bufpos] = line;
177     bufcolumn[bufpos] = column;
178 }
179
180 public char readChar() throws java.io.IOException
181 {
182     if (inBuf > 0)
183     {
184         --inBuf;
185
186         if (++bufpos == bufsize)
187             bufpos = 0;
188
189         return buffer[bufpos];
190     }
191
192     if (++bufpos >= maxNextCharInd)
193         FillBuff();
194
195     char c = buffer[bufpos];

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

196
197     UpdateLineColumn(c);
198     return (c);
199 }
200
201 /**
202  * @deprecated
203  * @see #getEndColumn
204  */
205
206 public int getColumn() {
207     return bufcolumn[bufpos];
208 }
209
210 /**
211  * @deprecated
212  * @see #getEndLine
213  */
214
215 public int getLine() {
216     return buflin[e[bufpos]];
217 }
218
219 public int getEndColumn() {
220     return bufcolumn[bufpos];
221 }
222
223 public int getEndLine() {
224     return buflin[e[bufpos]];
225 }
226
227 public int getBeginColumn() {
228     return bufcolumn[tokenBegin];
229 }
230
231 public int getBeginLine() {
232     return buflin[tokenBegin];
233 }
234
235 public void backup(int amount) {
236
237     inBuf += amount;
238     if ((bufpos -= amount) < 0)
239         bufpos += bufsize;
240 }
241
242 public SimpleCharStream(java.io.Reader dstream, int startline,
243 int startcolumn, int buffersize)
244 {
245     inputStream = dstream;
246     line = startline;
247     column = startcolumn - 1;
248
249     available = bufsize = buffersize;
250     buffer = new char[buffersize];
251     buflin = new int[buffersize];
252     bufcolumn = new int[buffersize];
253 }
254
255 public SimpleCharStream(java.io.Reader dstream, int startline,
256 int startcolumn)
257 {
258     this(dstream, startline, startcolumn, 4096);
259 }
260
261 public SimpleCharStream(java.io.Reader dstream)
262 {
263     this(dstream, 1, 1, 4096);
264 }

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

265 public void ReInit(java.io.Reader dstream, int startline,
266 int startcolumn, int buffersize)
267 {
268     inputStream = dstream;
269     line = startline;
270     column = startcolumn - 1;
271
272     if (buffer == null || buffersize != buffer.length)
273     {
274         available = bufsize = buffersize;
275         buffer = new char[buffersize];
276         bufline = new int[buffersize];
277         bufcolumn = new int[buffersize];
278     }
279     prevCharIsLF = prevCharIsCR = false;
280     tokenBegin = inBuf = maxNextCharInd = 0;
281     bufpos = -1;
282 }
283
284 public void ReInit(java.io.Reader dstream, int startline,
285 int startcolumn)
286 {
287     ReInit(dstream, startline, startcolumn, 4096);
288 }
289
290 public void ReInit(java.io.Reader dstream)
291 {
292     ReInit(dstream, 1, 1, 4096);
293 }
294 public SimpleCharStream(java.io.InputStream dstream, String encoding, int
startline,
295 int startcolumn, int buffersize) throws java.io.UnsupportedEncodingException
296 {
297     this(encoding == null ? new java.io.InputStreamReader(dstream) : new
java.io.InputStreamReader(dstream, encoding), startline, startcolumn, buffersize);
298 }
299
300 public SimpleCharStream(java.io.InputStream dstream, int startline,
301 int startcolumn, int buffersize)
302 {
303     this(new java.io.InputStreamReader(dstream), startline, startcolumn,
buffersize);
304 }
305
306 public SimpleCharStream(java.io.InputStream dstream, String encoding, int
startline,
307 int startcolumn) throws
java.io.UnsupportedEncodingException
308 {
309     this(dstream, encoding, startline, startcolumn, 4096);
310 }
311
312 public SimpleCharStream(java.io.InputStream dstream, int startline,
313 int startcolumn)
314 {
315     this(dstream, startline, startcolumn, 4096);
316 }
317
318 public SimpleCharStream(java.io.InputStream dstream, String encoding) throws
java.io.UnsupportedEncodingException
319 {
320     this(dstream, encoding, 1, 1, 4096);
321 }
322
323 public SimpleCharStream(java.io.InputStream dstream)
324 {
325     this(dstream, 1, 1, 4096);
326 }
327

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

328     public void ReInit(java.io.InputStream dstream, String encoding, int startline,
329                         int startcolumn, int buffersize) throws
java.io.UnsupportedEncodingException
330     {
331         ReInit(encoding == null ? new java.io.InputStreamReader(dstream) : new
java.io.InputStreamReader(dstream, encoding), startline, startcolumn, buffersize);
332     }
333
334     public void ReInit(java.io.InputStream dstream, int startline,
335                         int startcolumn, int buffersize)
336     {
337         ReInit(new java.io.InputStreamReader(dstream), startline, startcolumn,
buffersize);
338     }
339
340     public void ReInit(java.io.InputStream dstream, String encoding) throws
java.io.UnsupportedEncodingException
341     {
342         ReInit(dstream, encoding, 1, 1, 4096);
343     }
344
345     public void ReInit(java.io.InputStream dstream)
346     {
347         ReInit(dstream, 1, 1, 4096);
348     }
349     public void ReInit(java.io.InputStream dstream, String encoding, int startline,
350                         int startcolumn) throws java.io.UnsupportedEncodingException
351     {
352         ReInit(dstream, encoding, startline, startcolumn, 4096);
353     }
354     public void ReInit(java.io.InputStream dstream, int startline,
355                         int startcolumn)
356     {
357         ReInit(dstream, startline, startcolumn, 4096);
358     }
359     public String GetImage()
360     {
361         if (bufpos >= tokenBegin)
362             return new String(buffer, tokenBegin, bufpos - tokenBegin + 1);
363         else
364             return new String(buffer, tokenBegin, bufsize - tokenBegin) +
new String(buffer, 0, bufpos + 1);
365     }
366 }
367
368     public char[] GetSuffix(int len)
369     {
370         char[] ret = new char[len];
371
372         if ((bufpos + 1) >= len)
373             System.arraycopy(buffer, bufpos - len + 1, ret, 0, len);
374         else
375         {
376             System.arraycopy(buffer, bufsize - (len - bufpos - 1), ret, 0,
len - bufpos - 1);
377             System.arraycopy(buffer, 0, ret, len - bufpos - 1, bufpos + 1);
378         }
379     }
380
381     return ret;
382 }
383
384     public void Done()
385     {
386         buffer = null;
387         bufline = null;
388         bufcolumn = null;
389     }
390
391 /**
392  * Method to adjust line and column numbers for the start of a token.

```

```

393  */
394  public void adjustBeginLineColumn(int newLine, int newCol)
395  {
396      int start = tokenBegin;
397      int len;
398
399      if (bufpos >= tokenBegin)
400      {
401          len = bufpos - tokenBegin + inBuf + 1;
402      }
403      else
404      {
405          len = bufsize - tokenBegin + bufpos + 1 + inBuf;
406      }
407
408      int i = 0, j = 0, k = 0;
409      int nextColDiff = 0, columnDiff = 0;
410
411      while (i < len &&
412             bufline[j = start % bufsize] == bufline[k = ++start % bufsize])
413      {
414          bufline[j] = newLine;
415          nextColDiff = columnDiff + bufcolumn[k] - bufcolumn[j];
416          bufcolumn[j] = newCol + columnDiff;
417          columnDiff = nextColDiff;
418          i++;
419      }
420
421      if (i < len)
422      {
423          bufline[j] = newLine++;
424          bufcolumn[j] = newCol + columnDiff;
425
426          while (i++ < len)
427          {
428              if (bufline[j = start % bufsize] != bufline[++start % bufsize])
429                  bufline[j] = newLine++;
430              else
431                  bufline[j] = newLine;
432          }
433      }
434
435      line = bufline[j];
436      column = bufcolumn[j];
437  }
438
439 }
440

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder