

```
169     /// <summary>
170     /// Implements CFG Rule: <block> -> [<label-declaration>] {<variable-dec-sec>}* <block-body>
171     /// Also contains main error handling logic.
172     /// </summary>
173     /// <returns></returns>
174     private int Block()
175     {
176         if (IsError)
177             return -1;
178
179         Debug(true, "Block()");
180
181         if (Scanner.TokenCode == LABEL)
182         {
183             Scanner.PastDeclarationSection = false;
184             LabelDeclaration();
185         }
186
187         while (Scanner.TokenCode == VAR && !IsError)
188         {
189             Scanner.PastDeclarationSection = false;
190             VariableDecSec();
191         }
192
193         Scanner.PastDeclarationSection = true;
194
195         BlockBody();
196
197         // Error handling and resyncing
198         while (IsError == true && !Scanner.EndOfFile)
199         {
200             Resync();
201             IsError = false;
202             PrintError = true;
203             while (IsError == false && !Scanner.EndOfFile)
204             {
205                 Statement();
206
207                 if (Scanner.TokenCode == END)
208                 {
```

```
209         GetNextToken();
210         if (Scanner.TokenCode == PERIOD)
211             GetNextToken();
212         else
213             UnexpectedTokenError("PERIOD");
214     }
215     else if (Scanner.TokenCode == SEMICOLON)
216         GetNextToken();
217     else
218         UnexpectedTokenError("END or SEMICOLON");
219     }
220 }
221
222 Debug(false, "Block()");
223 return -1;
224 }
```