```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Dynamic;
 4  using System.Text;
 5
 6  namespace KyleBushCompiler
 7  {
 8      /// <summary>
 9      /// Used to specify data type of a symbol
10      /// </summary>
11      public enum DataType
12      {
13          Integer,
14          Double,
15          String
16      }
17
18      /// <summary>
19      /// Used to specify the kind of a symbol
20      /// </summary>
21      public enum SymbolKind
22      {
23          Label,
24          Variable,
25          Constant
26      }
27
28      public class Symbol
29      {
30          public string Name { get; set; }
31          public SymbolKind Kind { get; set; }
32          public DataType DataType { get; set; }
33
34          private int _intValue;
35          private string _stringValue;
36          private double _doubleValue;
37
38          /// <summary>
39          /// Contructor to initialize a Symbol containing an integer value.
40          /// </summary>
41          /// <param name="name">String name of symbol</param>
42          /// <param name="kind">Defines the kind of the symbol</param>
43          /// <param name="dataType">Defines the data type of the symbol</param>
44          /// <param name="value">The integer value of the symbol</param>
45          public Symbol(string name, SymbolKind kind, DataType dataType, int value)
46          {
47              Name = name;
48              Kind = kind;
49              DataType = dataType;
50              _intValue = value;
51          }
52
53          /// <summary>
54          /// Contructor to initialize a Symbol containing a double value.
```

```
55          /// </summary>
56          /// <param name="name">String name of symbol</param>
57          /// <param name="kind">Defines the kind of the symbol</param>
58          /// <param name="dataType">Defines the data type of the symbol</param>
59          /// <param name="value">The double value of the symbol</param>
60          public Symbol(string name, SymbolKind kind, DataType dataType, double
              value)
61          {
62              Name = name;
63              Kind = kind;
64              DataType = dataType;
65              _doubleValue = value;
66          }
67
68          /// <summary>
69          /// Contructor to initialize a Symbol containing a string value.
70          /// </summary>
71          /// <param name="name">String name of symbol</param>
72          /// <param name="kind">Defines the kind of the symbol</param>
73          /// <param name="dataType">Defines the data type of the symbol</param>
74          /// <param name="value">The string value of the symbol</param>
75          public Symbol(string name, SymbolKind kind, DataType dataType, string
              value)
76          {
77              Name = name;
78              Kind = kind;
79              DataType = dataType;
80              _stringValue = value;
81          }
82
83          /// <summary>
84          /// Sets a Symbol with an integer value.
85          /// </summary>
86          /// <param name="value">The integer value of the symbol</param>
87          public void SetValue(int value)
88          {
89              _intValue = value;
90          }
91
92          /// <summary>
93          /// Sets a Symbol with a string value.
94          /// </summary>
95          /// <param name="value">The string value of the symbol</param>
96          public void SetValue(string value)
97          {
98              _stringValue = value;
99          }
100
101         /// <summary>
102         /// Sets a Symbol with a double value.
103         /// </summary>
104         /// <param name="value">The double value of the symbol</param>
105         public void SetValue(double value)
106         {
```

```
107             _doubleValue = value;
108         }
109
110         /// <summary>
111         /// Checks the DataType of the Symbol and returns the appropriate value.
112         /// </summary>
113         /// <returns>int, string, or double depending on the DataType property.</ ↵
              returns>
114         public dynamic GetValue()
115         {
116             if (DataType == DataType.Integer)
117                 return _intValue;
118             else if (DataType == DataType.Double)
119                 return _doubleValue;
120             else
121                 return _stringValue;
122         }
123     }
124 }
125
```