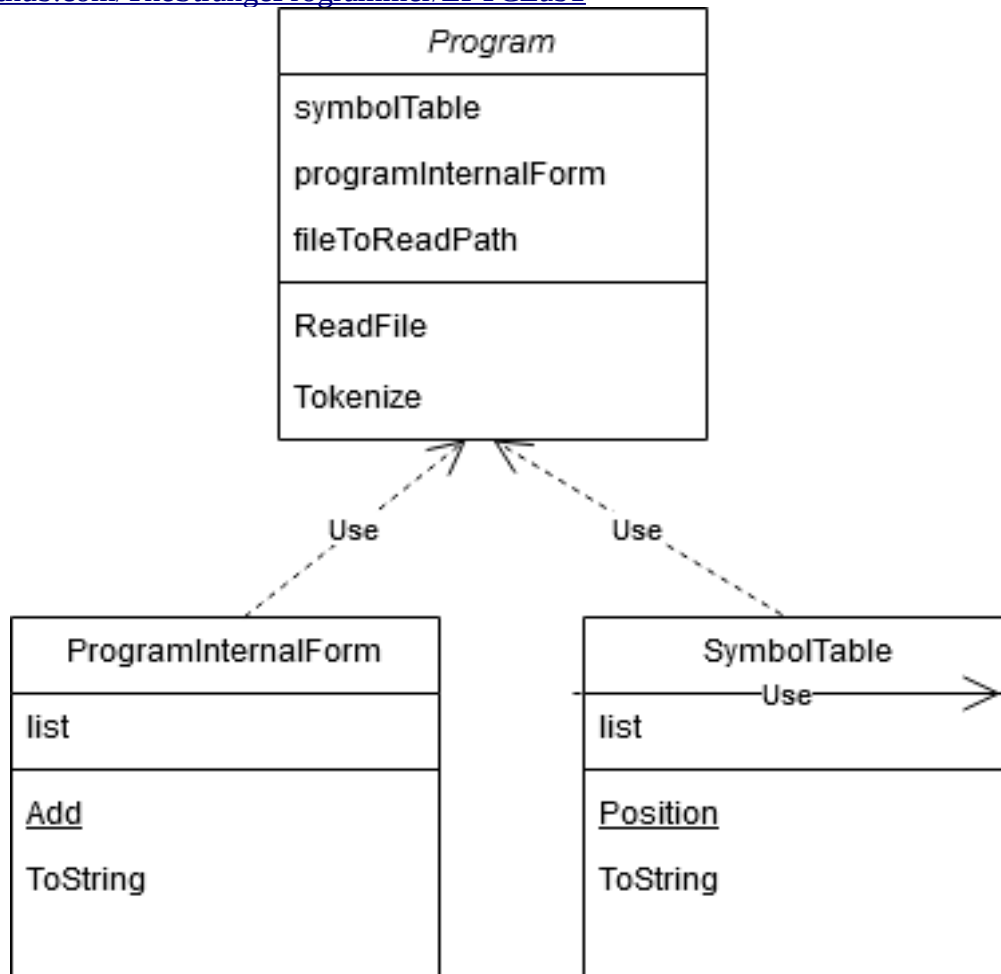Scherer Edward Serban

/// <summary>
/// Reads file and produces the PIF and ST
/// </summary>
static void ReadFile()

/// <summary>
/// Takes an array of tokens and splits them by the regex given
/// </summary>
/// <param name="tokens"></param>
/// <param name="regex"></param>
/// <returns>Returns an array of tokens</returns>
static string[] Tokenize(string[] tokens,Regex regex)

/// <summary>
/// Checks if the token is a newline
/// </summary>
/// <param name="token"></param>
/// <returns>true or false</returns>
private static bool isEndline(string token)

/// <summary>

```csharp
/// Checks if the token is a newline
/// </summary>
/// <param name="token"></param>
/// <returns>true or false</returns>
static bool isConstant(string token)

/// <summary>
/// Checks if the token is an identifier
/// </summary>
/// <param name="token"></param>
/// <returns>true or false</returns>
static bool isIdentifier(string token)

/// <summary>
/// Checks if the token is a reserved word
/// </summary>
/// <param name="token"></param>
/// <returns>true or false</returns>
static bool isReservedWord(string token)

/// <summary>
/// Checks if the token is a comment
/// </summary>
/// <param name="token"></param>
/// <returns>true or false</returns>
static bool isComment(string token)

/// <summary>
/// Checks if the token is a separator
/// </summary>
/// <param name="token"></param>
/// <returns>true or false</returns>
static bool isSeparator(string token)

/// <summary>
/// Checks if the token is an operator
/// </summary>
/// <param name="token"></param>
/// <returns>true or false</returns>
static bool isOperator(string token)
```