Compiler Design

Final Report

Joseph Fitzpatrick | Compiler Design | 22-Dec-16

# Introduction

In this report I will discuss the alterations I made to the original Tastier Code to allow it to gain a number of different attributes as specified through exercises through the semester. The changes were incremental week on week and allowed me to gain a more in-depth understanding of the language.

In this report I will try to explain the changes I made to the language.

# Comments

As part of the first lab we had to add the generation of comments to the language. This involved using the member data of each object in scope and printing them to the generated file. This alteration was made in the SysTab.cs file. The changes were made using the C# language.

# Constants

Constant Boolean and integers were added to the language in lab 2. Constants can be declared in either global or local scope. Once a constant has been initialised to a value it cannot be altered or changed. If an attempt to change them is made an error will be thrown and the program will not compile. A constant is declared using the “const” identifier at the start of the line.

# Arrays

The addition of arrays to the language was one of the most challenging tasks assigned. The code relating to the addition of arrays mainly took place in the Tastier.ATG file. Here I edited the “Stat” section. I added the ability for arrays to be created similar to that in another more well-known language, Java! The compiler takes in the line and can check to see if it has square brackets either at the start or encasing a number (int[3]temp; accessed by int k = temp[0] and updated by temp[1] = 3). This allows for an array to be declared and accessed easily in a familiar format.

To allow for arrays to be declared I had to allocate space for the elements of the array in a manner in which I could access the memory at a later stage. To do this I edited the SymTab.cs file to allow for arrays when a new object is being created. This involved creating an extra parameter to the NewObj function which helped with the indexing. If the new object that was being created wasn’t part of an array it was simply indexed at zero as the parameter passed to the function was -1.

# In-line Conditionals

Inline conditional statements are essentially an “if-else statement” compressed down onto one line. If the condition is true the first value is taken, otherwise, if it false the second value is taken. To implement this, I had to add it to the Stat section of the tastier.atg file. I used the same general format of the pre-existing if statement already in the stat file. The condition is evaluated and then the correct branch is taken depending on whether it is true or false. I found this relatively straight forward, however I ran into a problem trying to decide how to identify that it was an inline conditional statement. To solve this, I added “<>” brackets around the statement, similar to the example.

# For-loops

For loops are similar to while loops in that it loops until the exit condition is met. However, in for loops an index is sometimes declared and increment at each iteration. To implement the for loop I followed the same format as the while loop in the code already, however I added extra Stat functions to deal with the value declaration and increment. This allows for the complete functionality of the for-loop to be declared on while line. The “do” section allows for the body of the for loop to work.

# Switch Statement

As part of the final exercise we were asked to add some kind of extra feature. I decided to add the use of a Switch statement to the language. To achieve this, I added the code to the Stat function to allow for a switch statement to be created easily by the user. The Switch statement works on integers. A value or variable is placed in the brackets after the switch declaration ie. “switch(var)”. The var value is then compared to each of the cases defined below it. When a match is achieved, a Stat may be invoked, a break then follows the stat to end the switch. If no matched to any of the cases is found, a default case may be created where another Stat may be declared if needed.