Patrick Doyle

9/17/20

CIS 211 001

Lab Assignment – Kotlin Features

In the world of software development, there are a wide variety of programming languages. These languages all have different characteristics that can make one language more useful for a certain purpose than another. Two common features of programming languages include whether the language is statically or dynamically typed and whether it is strongly or weakly typed. If a language is statically-typed that means variable types are known at compile time, the data type of variables must be declared before using them, and variables can't be assigned to other variables of different types. The opposite of this is a dynamically-typed language, where variable types are checked at run time so the type does not need to be declared and they can be used with variables of different types. If a language is strongly-typed this means that variables are given a permanent data type and errors will occur if types don't match the expected type of an expression. The opposite of this is a weakly-typed language where variables do not have a permanent data type and expressions are more lenient with type constraints.

In this assignment, the problem presented was to write software in two different languages that asks the user to enter three numbers, calculates and outputs the mean, asks for a fourth number, adds it to the mean, and outputs the final value. The first language I coded in was Python, a dynamic and strongly-typed language. Because of this many errors are found early on and compiled code executes more quickly, as the compiler can optimize machine code based on the data types present. Disadvantages of this are that you cannot test your code until the compiler is finished, making debugging a longer process. Since Python is also strongly-typed, developers must be very careful with the data types they use because many expressions will return an error if data types don't match.

Is there a preference most developers have regarding static/dynamic and strongly/weakly typing? In researching this question, I could not find a consensus answer. So, what I decided to do was to see what the most popular languages are and determine if there is a preference from there. According to Northeastern University, the top 5 most popular languages in 2020 are Python, JavaScript, Java, C#, and C. Python is dynamic and strongly-typed, JavaScript is dynamic and weakly-typed, Java and C# are static and strongly-typed, and C is static and weakly-typed. Based on these results there does not appear to be a whole lot of preference for a certain characteristic. It can be assumed that each of these languages has its pros and cons in different scenarios.

The major differences in the two programs were how many data conversions were required and how many lines it took to perform certain operations. In Python, for example, you can only output variables of type string, so every integer and double variable had to be converted to a string. This is not the case in Kotlin, you can simply concatenate the integer or double value and it will print. However, Kotlin required extra lines when calculating the mean. If you simply added the integer variables and divided them, it would return an integer for the mean. This is not accurate, as decimals are common when averaging numbers. So, the original integers had to be converted to type double before calculating the mean. This is not necessary in Python. Moving onto the other major difference, assigning user input to variables in Kotlin requires more code than in Python. For my Kotlin program, I had to output instructions to the user on one line, assign a variable to the readLine() function followed by ?:”” (null coalescing operator must be used so if there is no input it will input an empty string) on another line. In python, this can all be done on one line.

There were some challenges in writing the code for these programs. For starters, I am not all that familiar with Kotlin or Python, as most of my experience in programming is with Java. However, this assignment was extremely basic so it really was not a big issue. The biggest inconvenience occurred due to not knowing Python can only output strings.

Strong-typing may be annoying to a developer at times, but it can prevent many type errors. When a program is constantly mixing data types with one another, it can lead to unexpected results. For example, if variable a is the string “2” and variable b is the integer 2 and you wanted to add them mathematically (a + b), a strongly typed language would probably give you an error. You could then change the string to an integer and move on. However, a weakly-typed language would probably just concatenate the values, resulting in something different than anticipated. It would not give you an error, so you might not notice the issue.

In conclusion, there are many languages with many differences. Languages are either statically or dynamically typed. They are also either strongly or weakly typed. These characteristics make different languages unique in usefulness and feel.

Sources

https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/

https://android.jlelse.eu/magic-lies-here-statically-typed-vs-dynamically-typed-languages-d151c7f95e2b