Jake Taggart

Week 4 Day 4 Research

1. What is static, inferred and dynamic typing?

   Static typing means that the developer needs to specify which type of variable is being declared and also means that boilerplate code like default getters and setters must be defined explicitly while inferred typing means that the type of variable doesn't need to be explicitly stated, but is decided by the format in the assignment. Inferred also allows those default getters and setters to be supplemented automatically when creating an object.

2. How do string pools work in Java?

   String pooling in Java works on a dependent basis rooted in whether the developer creates a String by assigning and declaring with or without the "new" keyword. If a String is created using the "new" keyword, then a new memory space and value is allocated in memory by the JVM. However if a String initialization/declaration does not include the "new" keyword (such as String str = "Hello World"), then the JVM will search the memory heap for Strings that contain the exact value specified in the declaration. If the value of the String literal is found in the heap, then the newly declared variable is not given a new memory spot refers to the same memory allocation found in the heap that already existed. Until the String literal is changed, it will still refer to that same memory reference.

3. What is functional Programming and how does it differ from pure object oriented?

   Functional programming is a form of development in which the code is does not rely on objects. This uses less resources. Functional programming objects exist, but they are entirely immutable once created. In functional program, there is complete separation of data and behaviors of a program. A function is always going to return the same value so long as the input value is the same. In functional programming, any object created does not share its scope with other objects. The data input is not altered upon entering a function, it is just used for its value so the output can retrieve how the input would be altered if it were. Network and Database calls do not affect the return value in a function in functional programming. These are all oppositional to how OOP operates.

4. What is lambda function?

Lamda expressions are a way to ensure that programming languages are able to make full use of the parallel processing capabilities of multi-core environments. Lambda functions/methods are anonymous used to carry out methods defined by a functional interface. A functional interface is an interface which contains one and only one abstract method. A good example of this would be Runnable.

5. What are some ways Kotlin achieves null safety?

   First of all, Kotlin infers that any value cannot be null. “Lateinit” can be used to tell the system that the value will be assigned a value before running the code. A nullable reference can be specified in a variable assignment by putting a “?” after specifying the type of variable. Safe calls are another way that Kotlin achieves null safety. This allows programmers to only call a method if the passed parameter is non-null. The Elvis operator is a way to return a default value in a function if the parameter accepted is null. Kotlin also allows filtering out nulls in a list of values. The let() operator is a way to perform an action on only the non-nullable values in a list.