Static and Stack functions do not take very long to execute. This because for those functions, the array was already bound to some memory cells throughout the program running. Basically, this means that the memory for this had already been allocated and no real calculations or allocations were needed to find the storage for this. Now the heap functions takes significantly longer because there is a need to look for storage in a disorganized storage area. Also, in order to return the variable it needs to find the right reference. Finding the correct reference in something as disorganized in the heap takes a good amount of time.

Why or why not in Java?

In Java, the memory allocation can only be done to the heap. The heap space will contain the object and stack will contain the reference to the heap memory. In the code that I provided, it does not function exactly like this since I am explicitly creating the reference when saving to the heap memory. This means it has to go and find the reference again which is why it takes a long time to run the last method.