

Tail Recursion

Implementation Consideration

- If a function calls itself as its last action, the function's stack frame can be reused; this is called tail recursion
- Tail recursive functions are iterative processes
- In general, if the last action of a function consists of calling a function, one stack frame would be sufficient for both functions; such calls are called tail-calls

Tail Recursion in Scala

- Only directly recursive calls to the current function are optimized
- Such an optimization can help the runtime speed of the function and it can avoid stack overflows
- One can require that a function is tail-recursive using a **@tailrec** annotation
- If the annotation is given, but the function is not tail-recursive, then the compiler would give you a warning: *This function is declared as @tailrec but the function is not tail-recursive*