@dynamicCallable Implementation Requirements

If a type is marked with the `@dynamicCallable` attribute, it must provide a valid implementation of `dynamicallyCall(withArguments:)`, `dynamicallyCall(withKeywordArguments:)`, or both. If it fails to do so, an error will be reported at compile-time. Note that an implementation of `dynamicallyCall(withKeywordArguments:)` is required to support calls with keyword arguments.

To be considered valid, an implementation of `dynamicallyCall(withArguments:)` must: - Be an instance method. `static` or `class` implementations are not allowed. - Have an argument type which conforms to the `ExpressibleByArrayLiteral` protocol. Often, this will be the built in `Array` type. - The return type of `dynamicallyCall(withArguments:)` may be any valid type.

To be considered valid, an implementation of `dynamicallyCall(withKeywordArguments:)` must: - Be an instance method. `static` or `class` implementations are not allowed. - Have an argument type which conforms to the `ExpressibleByDictionaryLiteral` protocol. This can be `Dictionary`, `KeyValuePairs` (which may be used to support duplicated keyword arguments), or some other conforming type. - The `Key` associated type of the argument type must conform to the `ExpressibleByStringLiteral` protocol. This type is used to represent the dynamic argument keywords. - The `Value` associated type of the argument type and the return type of `dynamicallyCall(withKeywordArguments:)` may be any valid types.