Homework 6. Generics in Go, Java and OCaml

Background

You're working for a Metahilang, a software development company that writes a lot of high-level code in Java and OCaml, mostly for financial applications but also for other uses. Your codebase heavily uses generic types and you and your customers rely on these generic types heavily. Some of your app and libraries are available in both Java and OCaml; others are just in one language.

Some of your customers have been using Go for projects and have asked for Go versions of some of your libraries. You've said no because Go lacks generics and your software designs depend on generics so heavily. However, Go 1.18 (2022-03-15) added support for generic code using parameterized types, so you are now thinking of revisiting this decision.

Assignment

First, read the following:

- Cox R, Greisner R, Pike R, Taylor IL, Thompson K. The Go programming language and environment. *CACM*. 2022;54(9):1–40. doi:10.1145/3488716.
- The Go Team, Go 1.18 is released! (2022-03-15).
- Greisemer R, Taylor IL. An introduction to generics (2022-03-22).
- Taylor IL. When to use generics (2022-04-12).

Then, answer the following questions:

1. Greisemer and Taylor write:

   > The exact details of how type inference works are complicated, but using it is not: type inference either succeeds or fails. If it succeeds, type arguments can be omitted, and calling generic functions looks no different than calling ordinary functions. If type inference fails, the compiler will give an error message, and in those cases we can just provide the necessary type arguments.

   How well does this statement apply to Java and OCaml? Where there are differences, briefly explain the differences with examples; if there are no differences, briefly explain why not.
2. Taylor's "When to use generics" contains four examples of when type parameters are useful:
   - When using language-defined container types
   - General purpose data structures
   - For type parameters, prefer functions to methods
   - Implementing a common method

   and three examples of when type parameters are not useful:
   - Don't replace interface types with type parameters
   - Don't use type parameters if method implementations differ
   - Use reflection where appropriate

   For each of these seven examples, briefly explain whether it applies equally well to Java.

Submit

Submit a file `hw6.pdf` containing your answers to the questions.

$Id: hw6.html,v 1.84 2022/05/26 16:36:15 eggert Exp $