Subject: Type aliases have too many unreasonable restrictions
From: Van Snyder

There is no good reason to prohibit type aliases to have assumed or deferred type parameters. They offer no challenges that are different from type parameter values that are specification expressions. It would be more consistent to require initialization expressions for all type parameters in type alias definitions, but that would be too strong a requirement.

1 Edits

Edits refer to 01-007. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by immediately following text, while a page and line number followed by + (-) indicates that immediately following text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

The type parameter order of a type alias name is the order of the deferred type parameters of the type for which it is an alias. 54:9+ New ¶

---

or *type-alias-name* [(*type-param-spec-list*)] 54:17

Constraint: *type-param-list* shall appear if and only if *type-name* specifies a type that is parameterized, or *type-alias-name* specifies a type alias that has deferred type parameters. 54:22

---

type-name or each deferred parameter of the *type-alias-name*. 54:24

[Editor: "type" ⇒ "*type-name* or a deferred parameter of the *type-alias-name*"] 54:27

The type parameters of a type alias name are the type parameters of the type specified by the *type-spec*. 58:6-7

NOTE 4.59$\frac{1}{2}$

The only type parameters that can be specified when using a type alias in a declaration are the deferred parameters of the *type-spec*.

---

or TYPE (*type-alias-name* [(*type-param-spec-list*)]) 63:34