Dropped: Existential Types

Existential types using `forSome` (as in SLS §3.2.12) have been dropped. The reasons for dropping them are:

- Existential types violate a type soundness principle on which DOT and Scala 3 are constructed. That principle says that every prefix (`p` , respectvely `S`) of a type selection `p.T` or `S#T` must either come from a value constructed at runtime or refer to a type that is known to have only good bounds.
- Existential types create many difficult feature interactions with other Scala constructs.
- Existential types largely overlap with path-dependent types, so the gain of having them is relatively minor.

Existential types that can be expressed using only wildcards (but not `forSome`) are still supported, but are treated as refined types. For instance, the type

```
Map[_ <: AnyRef, Int]
```

is treated as the type `Map` , where the first type parameter is upper-bounded by `AnyRef` and the second type parameter is an alias of `Int` .

When reading class files compiled with Scala 2, Scala 3 will do a best effort to approximate existential types with its own types. It will issue a warning that a precise emulation is not possible.

Dropp... Dropp...