Dropped: private[this] and protected[this]

The `private[this]` and `protected[this]` access modifiers are deprecated and will be phased out.

Previously, these modifiers were needed for

- avoiding the generation of getters and setters
- excluding code under a `private[this]` from variance checks. (Scala 2 also excludes `protected[this]` but this was found to be unsound and was therefore removed).
- avoiding the generation of fields, if a `private[this] val` is not accessed by a class method.

The compiler now infers for `private` members the fact that they are only accessed via `this`. Such members are treated as if they had been declared `private[this]`. `protected[this]` is dropped without a replacement.

This change can in some cases change the semantics of a Scala program, since a `private` val is no longer guaranteed to generate a field. The field is omitted if

- the `val` is only accessed via `this`, and
- the `val` is not accessed from a method in the current class.

This can cause problems if a program tries to access the missing private field via reflection. The recommended fix is to declare the field instead to be qualified private with the enclosing class as qualifier. Example:

```
class C(x: Int):
    private[C] val field = x + 1
      // [C] needed if `field` is to be accessed through reflection
```

```
val retained = field * field
```

Deprec... Dropp...