Modules: Collateral Damage - Reflection on private/protected fields/methods

```clojure
(defn add-url [^URLClassLoader cl url]
  (let [method (.getDeclaredMethod URLClassLoader
                  "addURL" (into-array [URL]))]
    (.setAccessible method true)
    (.invoke method cl (into-array [(URL. url)]))))
```

```
java.lang.reflect.InaccessibleObjectException: Unable to
make protected void
java.net.URLClassLoader.addURL(java.net.URL) accessible:
module java.base does not "opens java.net" to unnamed
module @24bdb479
```

Modules: Collateral Damage - Reflection on private/protected fields/ methods

```
(defn add-url [^URLClassLoader cl url]
  (let [method (.getDeclaredMethod URLClassLoader
                  "addURL" (into-array [URL]))]
    (.setAccessible method true)
    (.invoke method cl (into-array [(URL. url)]))))
```

```
java.lang.reflect.InaccessibleObjectException: Unable to
make protected void
java.net.URLClassLoader.addURL(java.net.URL) accessible:
module java.base does not "opens java.net" to unnamed
module @24bdb479
```

Modules: Collateral Damage - Reflection on private/protected fields/methods

```
java \
  --add-opens=java.base/java.net=ALL-UNNAMED ...
```