

Pluggable Type Checking in Java For The Masses

Christopher Addison, Conner Knight



Pluggable Types In Java

We plan to use pluggable types to implement checks for things like @NonNull, @IntRange, and @Intdef in a complete manner for Java

Java added pluggable types in version 8 and presents hooks for plugins to use them

Implementation of the checks are done entirely in the plugin

The Checker Framework uses these hooks and provides backwards compatibility back to Java 4

This is an example:

```
public static @Nullable String getName(@NonNull String username)
```

Android Studio

Android Studio extends the framework built into IntelliJ from JetBrains

It provides different set of pluggable types to the Checker Framework

@IntRange and @FloatRange are some examples

Enhances the experience for programming Android in the IDE

Also allows Kotlin, the new 1st class language for Android, to make nullness guarantees

Bring the Types To The Checker Framework

We will create our own 3rd-party plugin for the Checker Framework

Specifically bring the `@FloatRange` and `@IntRange` types to the framework

And, experiment and research the effectiveness and usefulness of pluggable types in Java

Implement more than the current Java `@Nullable` and `@NotNull`, by adding additional possible restrictions and making ours complete

Provides the user with guarantees that their function will only be given certain inputs and that it will only be given certain outputs.