@Nullable and @NotNull Annotations

Overview

This section describes @Nullable and @NotNull annotations introduced in IntelliJ IDEA for catching NullPointerException's (NPE's) through the **Constant Conditions & Exceptions** and @Nullable problem inspections.

These annotations are designed to help you watch contracts throughout method hierarchies to avoid emergence of NPE's. Moreover, IntelliJ IDEA provides another benefit for them: the Code Inspection mechanism informs you on such contracts' discrepancies in places where annotated methods are called and provides automated solutions in some cases.

Two annotations - @Nullable and @NotNull - handle method invocations and field dereferences outside methods.

@Nullable

The @Nullable Annotation reminds you about the necessity to introduce an NPE check when:

- Calling methods that can return null.
- Dereferencing variables (fields, local variables, parameters) that can be null.

@NotNull

The @NotNull Annotation is, actually, an explicit contract declaring the following:

- A method should not return null.
- A variable (like fields, local variables, and parameters) cannot hold null value.

IntelliJ IDEA warns you if these contracts are violated.

> For more information and code examples, refer to online how-to.

Formal semantics

An element annotated with @Nullable claims null value is perfectly valid to return (for methods), pass to (for parameters) and hold (for local variables and fields).

An element annotated with @NotNull claims null value is forbidden to return (for methods), pass to (for parameters) and hold (for local variables and fields).

There is a covariance-contravariance relationship between @Nullable and @NotNull when overriding/implementing methods with annotated declaration or parameters.

- Overriding/implementing methods with an annotated declaration:
    - The @NotNull annotation of the parent method requires the @NotNull annotation for the child class method.
    - Methods with the @Nullable annotation in the parent method can have either @Nullable or @NotNull annotations in the child class method.
- Overriding/implementing methods with annotated parameters:
    - The @Nullable annotation of the parameter in the parent method requires the @Nullable annotation for the child class method parameter.
    - Methods with the @NotNull annotation of the parameter in the parent method can have either @Nullable or @NotNull annotations (or none of them) for the child class method parameter.

See Also

Concepts:

- External Annotations
- @NonNls Annotation

Procedures:

- Annotating Source Code Directly

Web Resources:

- Developer Community