Working with Diagrams

UML modeling support provided by IntelliJ IDEA involves two aspects:

- *Reverse engineering*, which involves drawing a UML model on the base of the existing code base. Such model helps you get an overview of the classes and packages that comprise your application, relationships between them, explore libraries, and view dependencies.
- *Forward engineering*, which enables you to design and create a visual model, and populate it with node elements, members and relationships. IntelliJ IDEA automatically generates source code and keeps it synchronized with the model.

UML model in IntelliJ IDEA is represented by a Class diagram in standard notation.

IntelliJ IDEA enables using UML class diagrams to analyze Java, ActionScript/Flex, PHP, and Maven applications, and the structure of the databases and tables. Besides that, you can explore changes committed to VCS.

In IntelliJ IDEA, Class diagram features:

- Ability to view UML model as a diagram in a separate editor tab, in a pop-up window, or as a preview.
- Ability to invoke UML class diagram from the **Project**, **Structure**, **Database**, **Maven**, **Changes** tool windows, the **History** tab of the **Version Control** tool window, and **Navigation** bar.

  In the editor, one can view class diagram for the whole class, or for a symbol at caret.
- Navigation from a diagram element to the underlying source code.
- Highlighting siblings and children classes and packages.
- Refactorings (Rename, Type Migration, Move, Safe Delete, Extract Class, Invert Boolean, Remove Middleman, Inline, Encapsulate, Migrate, Change Signature, Make Static, Convert to Instance Method, Introduce Parameter Object, Wrap Return Value)
- Navigation to class, file or symbol by name and to the last edit location.
- Viewing class or package information at the tooltip, and quick documentation lookup.
- Viewing changed classes as a UML Class diagram.
- Quick hierarchy view in a UML Class diagram pop-up window.
- Viewing subtypes, super classes and classes from signatures.
- Ability to find usages of a node element or member.
- ability to configure default settings for UML Class diagram.

See Also

Concepts:

- Dependencies Analysis

Reference:

- Class Diagram Toolbar and Context Menu

External Links:

- http://www.uml.org/#UML2.0

Web Resources:

- Developer Community