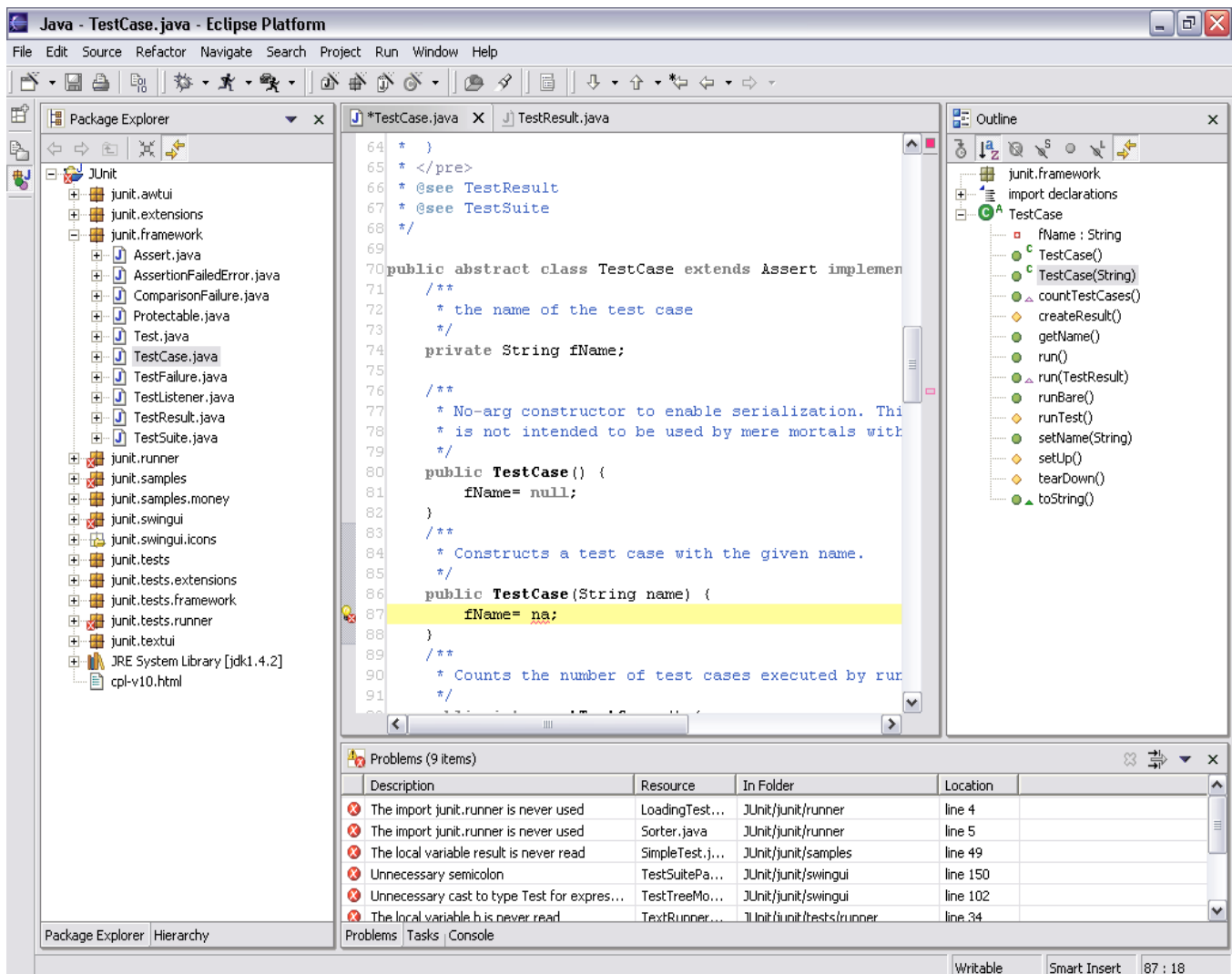


Editor Focused Workbench

Tom Eicher, Christof Marti, Daniel Megert
IBM OTI Labs Zurich

- When writing code, you want to:
 - see the context of the edited source code,
 - have proposals of possible completions,
 - avoid typing mistakes and syntax errors,
 - avoid entering the same text repeatedly,
 - minimize scrolling, dragging, clicking.
- When browsing code, you want to:
 - directly navigate type and call hierarchies,
 - quickly jump to a specific type or method,
 - get an overview of a source file,
 - view relevant code section at a glance,
 - navigate back and forth in the browsing history.
- When drinking coffee, you want to:
 - have a big mug,
 - increase the caffeine level.



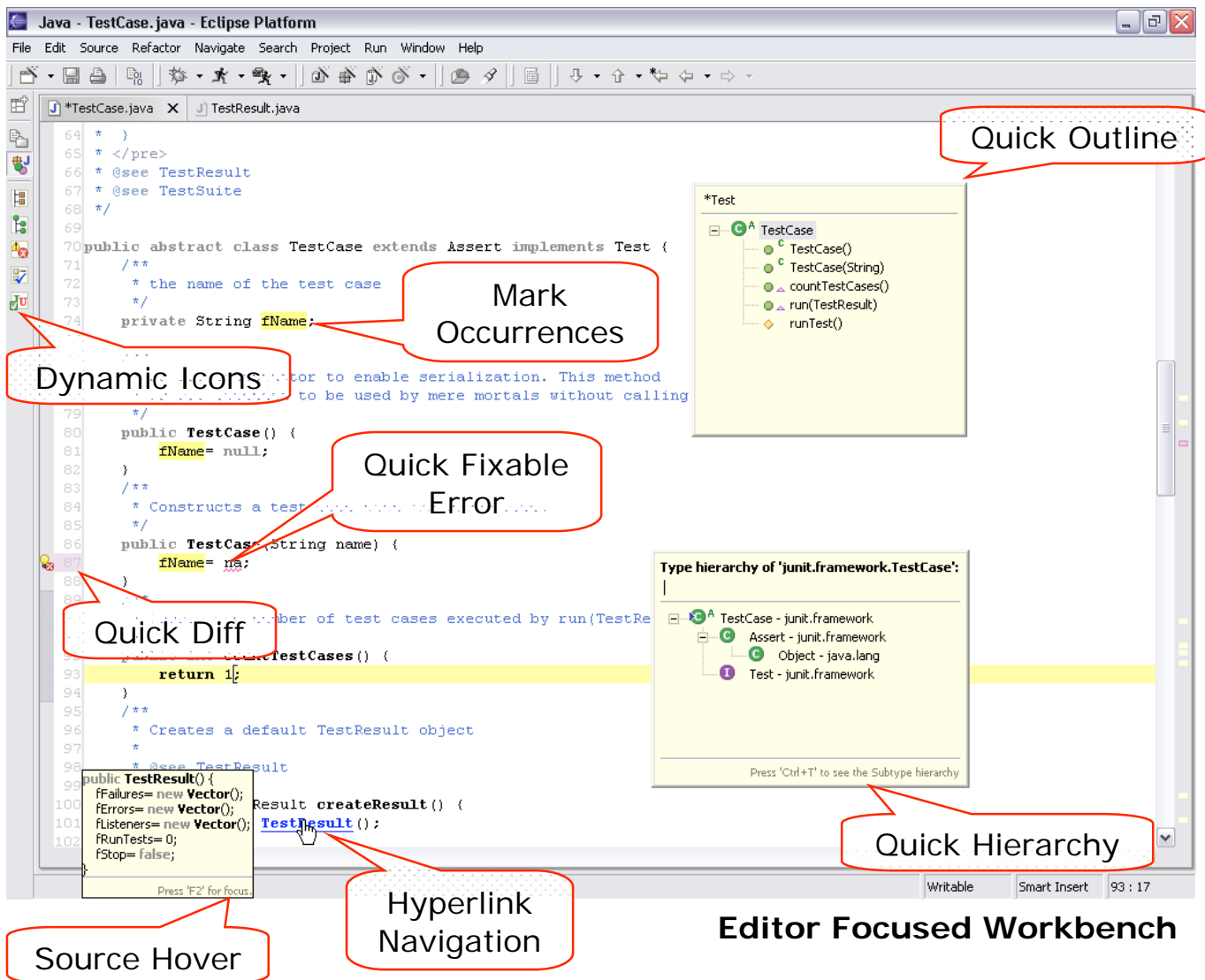
Classic Workbench Setup

The classic workbench setup allocates a **small area** of the application window to the editor. Parts of the relevant code **remain hidden**. Some of the displayed views are **not always needed**. Unproductive activities like scrolling, managing views, and manipulating the UI are **too frequent**.

During development you can maximize the editor area by removing persistent views and reducing dialogs. To compensate for the resulting loss of context information, Eclipse offers a growing set of specialized UI elements in the following areas:

- In-place context information
 - Quick navigation views
 - Keyboard accessible Fast Views
 - Quick views: Outline, Type / Method Hierarchy 3.0
 - Condense information
 - Dynamic icons
 - Quick Diff 3.0
 - Dynamic error annotation
 - Lightweight access to reference information
 - Dynamically mark occurrences 3.0
 - Javadoc and source hovers

- In-place interactions
 - Code input and transformation
 - Content Assist
 - Local rename
 - Code templates
 - Quick Assist
 - Navigation
 - Filterable pop-up dialogs
 - Hyperlink navigation
 - Go to next / previous annotation 3.0



An editor focused workbench configured to use **Fast Views** and **in-place controls** offers a lean user interface that lets you concentrate on writing code with **efficient browsing tools** and **precise context information** just one keystroke away.

Outlook:

- Folding of code blocks (e.g., Javadoc comments, method bodies)
- Flat Outline showing all inherited members