In-line JavaDoc Comment Editing

Comment Formatting and Highlighting

Users may have varying preferences for formatting their JavaDoc comments. The IDE should support a general set of automatic JavaDoc formatting options to accommodate these preferences.

1. The user should have the option to have comments added on or below the line containing the opening comment
2. First sentence syntax highlighting. Because this comment is special in the JavaDoc (i.e. it is the comment that shows up in the package or class member summary), the user should have the option of having this line highlighted to distinguish it from the rest of the comment.
3. JavaDoc tags should be separately highlighted.
4. Parameters to JavaDoc tags should be separately highlighted.
5. The user should have options for aligning and spacing JavaDoc tag parameters. (See example)
6. The user should have options for wrapped line indentation. The user should be able to decide the column position at which wrapped lines are indented or whether to continue the wrapped line at the column position at which the comment began. (See example)
7. The formatting should allow for inserting spaces between paragraphs and tag groups for easier reading.

Example:

```
/**
 * <p>Searches for records having the given keywords and returns a result
 * containing no more than the number of matches specified by
 * <code>fetchLimit</code> beginning at the given <code>offset</code>
 * position. If no items are found the returned <code>SearchResult</code>
 * will contain a collection of zero items.</p>
 *
 * <p>The implementation may choose to apply an upper constraint to the
 * <code>fetchLimit</code> parameter.</p>
 *
 * @param  keywords    The keywords to look for in each record.
 * @param  offset      The first position in the result set to include in
 *                     the returned <code>SearchResult</code>. Must be
 *                     zero or greater.
 * @param  fetchLimit  The maximum number of items to include in the
 *                     returned <code>SearchResult</code>. Must be greater
 *                     than zero.
 *
 * @return A search result.
 *
 * @throws IllegalArgumentException if <code>keywords</code> is blank,
 *             <code>offset</code is negative, or <code>fetchLimit</code>
 *             is negative or zero (optional).
 * @throws SearchException if an error occurs that prevents the search
 *             from completing normally (optional).
 *
 * @See    SearchResult
 */
public SearchResult search(String keywords, int offset, int fetchLimit)
```

```
        throws IllegalArgumentException, SearchException;
```

Word-wrapping

The IDE should provide an option for word-wrapping JavaDoc comments.

1. The editor options should provide automatic word wrapping at a position defined in the user's preferences.
2. The word wrapping feature should take into account the indentation of the comment so as not to overflow the editor's defined text limit line.
3. Word wrapping should not affect any data inside <pre></pre> tags.
4. Wrapped comments should be continued on the next line at the appropriate column position according to the user's comment formatting preferences. (See Comment formatting and Highlighting)