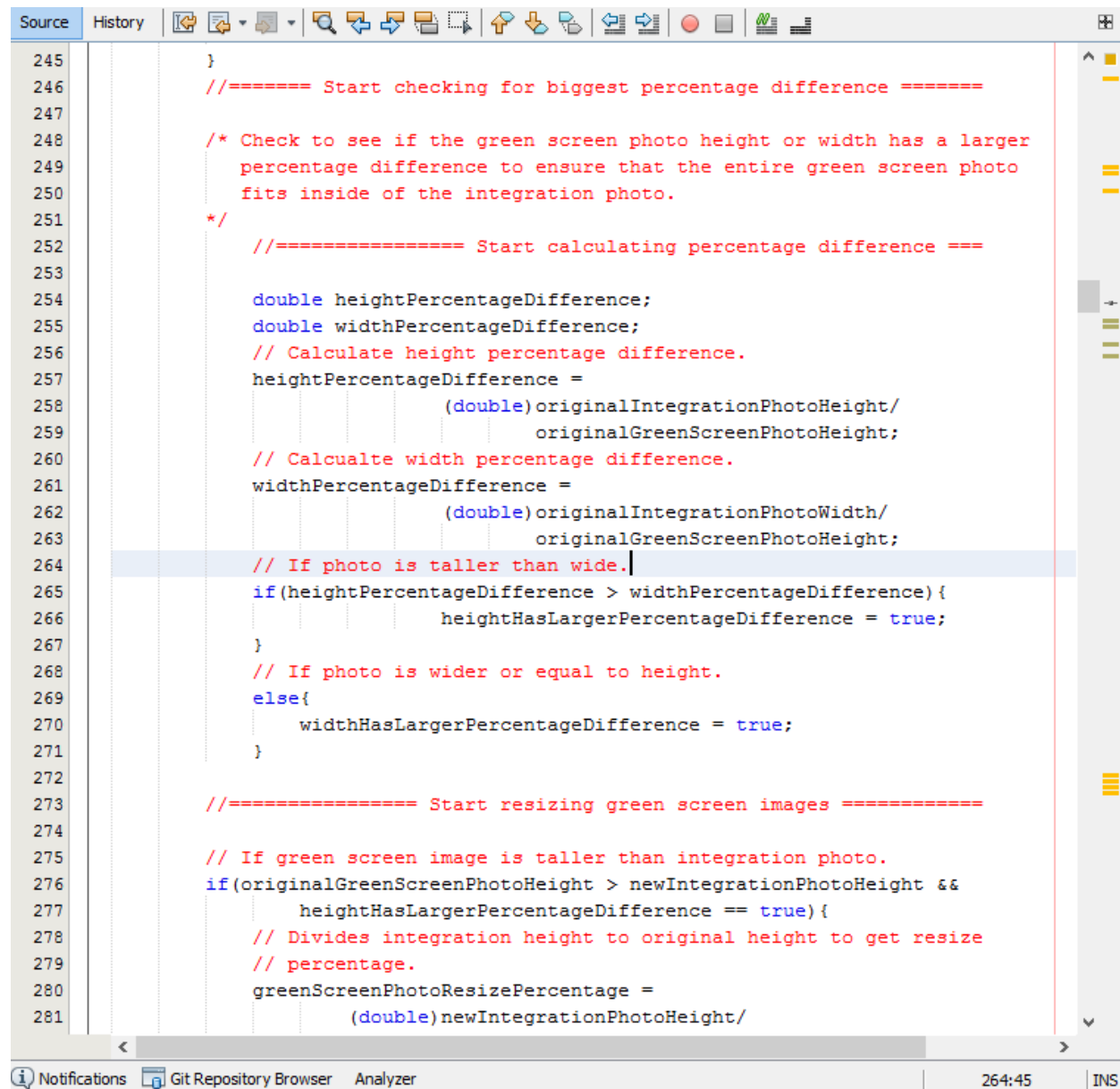


*All images are photo shopped and used to display a suggested idea of highlighted comment markers in place of in-text commenting. Not an actual working IDE.

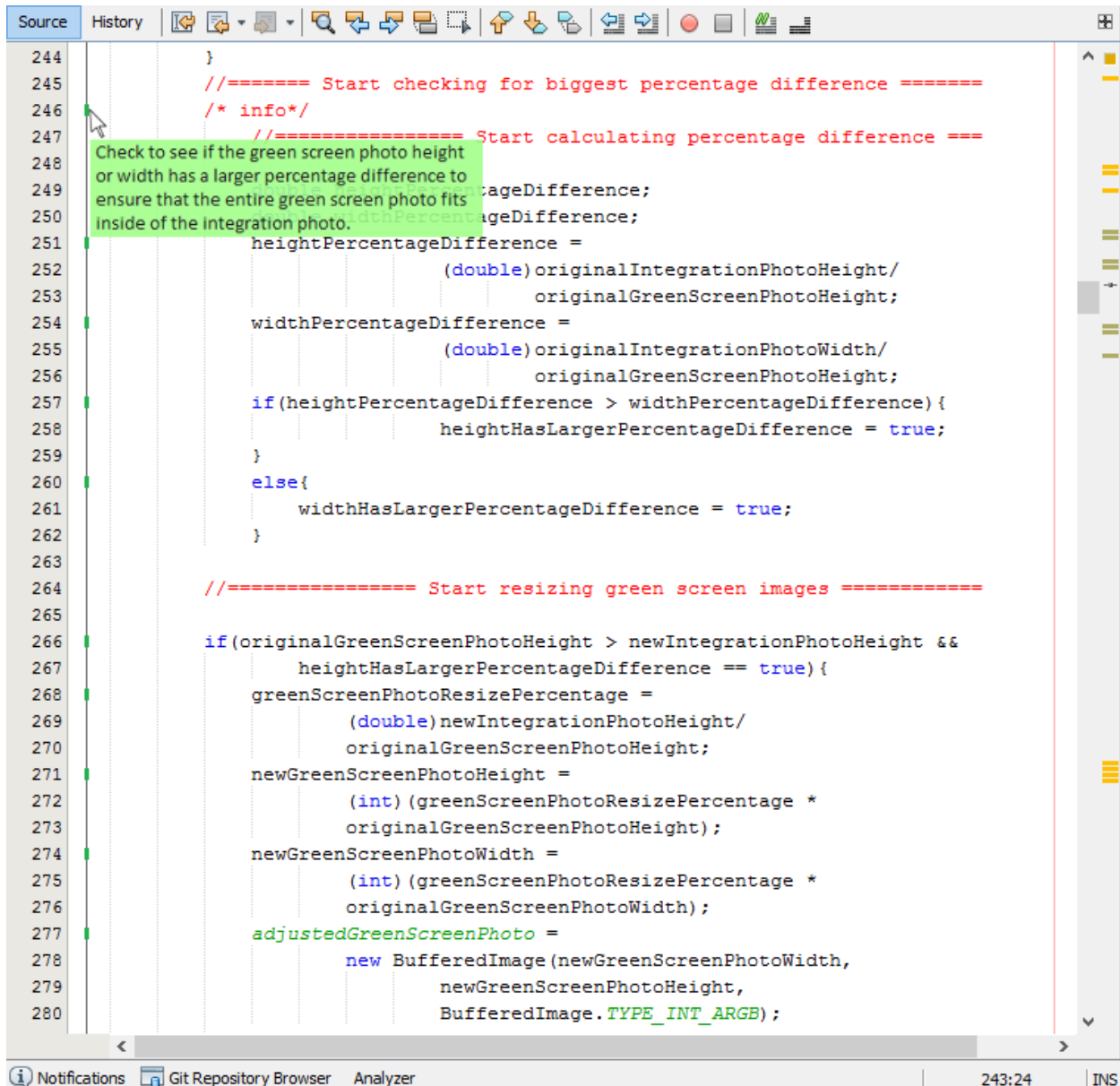
The idea is to be able to place something like “/* insert comment” above the line that you would like the hidden comment to be added too. The hidden comments can be toggled from viewable to hidden by a “Show/Hide Hidden Comments” option, to allow for editing of the comments.

Original code with comments.



```
245     }
246     //===== Start checking for biggest percentage difference =====
247
248     /* Check to see if the green screen photo height or width has a larger
249     percentage difference to ensure that the entire green screen photo
250     fits inside of the integration photo.
251     */
252     //===== Start calculating percentage difference ===
253
254     double heightPercentageDifference;
255     double widthPercentageDifference;
256     // Calculate height percentage difference.
257     heightPercentageDifference =
258         (double)originalIntegrationPhotoHeight/
259         originalGreenScreenPhotoHeight;
260     // Calcualte width percentage difference.
261     widthPercentageDifference =
262         (double)originalIntegrationPhotoWidth/
263         originalGreenScreenPhotoHeight;
264     // If photo is taller than wide.
265     if(heightPercentageDifference > widthPercentageDifference){
266         heightHasLargerPercentageDifference = true;
267     }
268     // If photo is wider or equal to height.
269     else{
270         widthHasLargerPercentageDifference = true;
271     }
272
273     //===== Start resizing green screen images =====
274
275     // If green screen image is taller than integration photo.
276     if(originalGreenScreenPhotoHeight > newIntegrationPhotoHeight &&
277        heightHasLargerPercentageDifference == true){
278         // Divides integration height to original height to get resize
279         // percentage.
280         greenScreenPhotoResizePercentage =
281         (double)newIntegrationPhotoHeight/
```

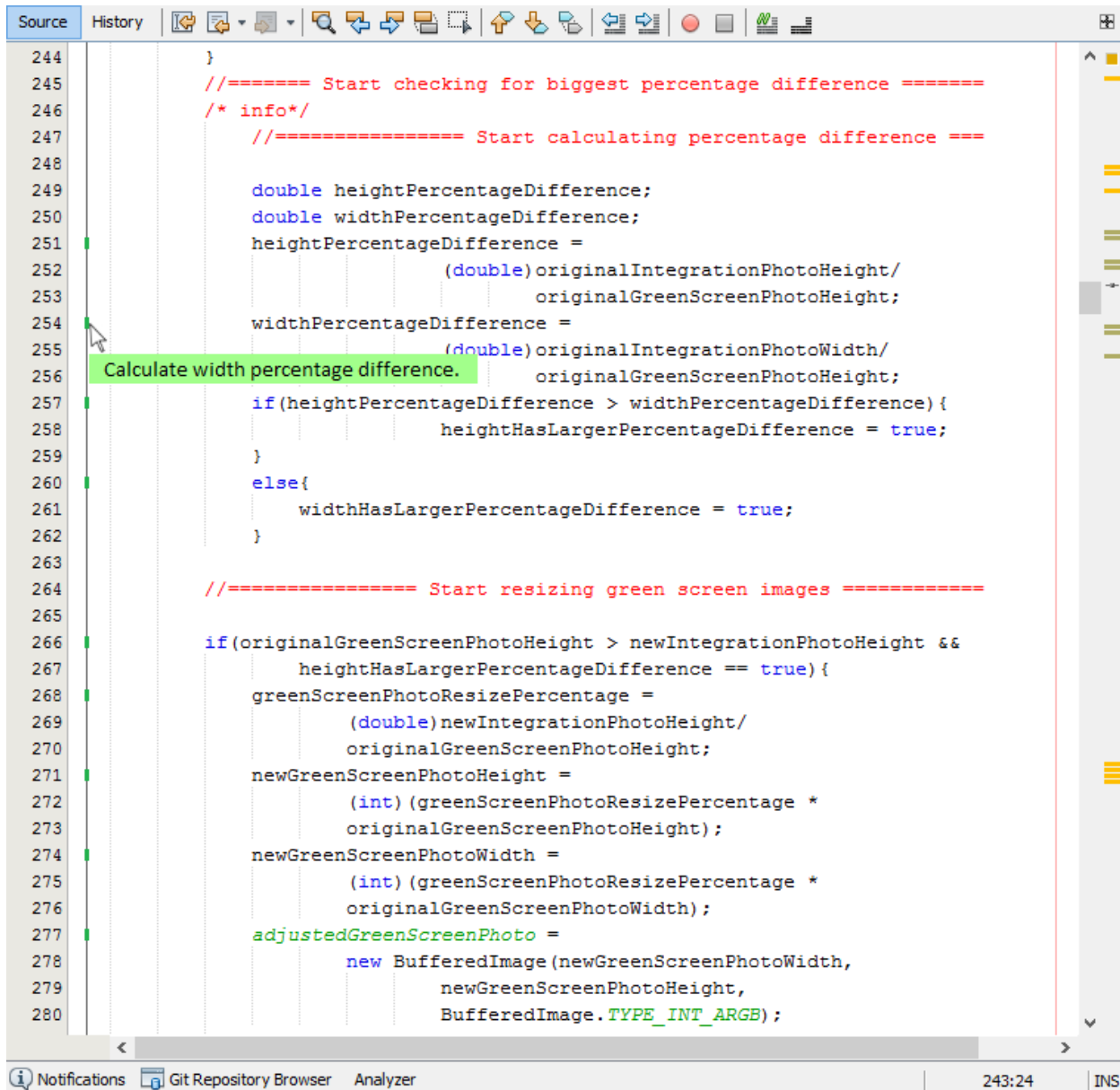
Code with hidden comments and comment markers.



```
244     }
245     //===== Start checking for biggest percentage difference =====
246     /* info*/
247     //===== Start calculating percentage difference ===
248     Check to see if the green screen photo height
249     or width has a larger percentage difference to
250     ensure that the entire green screen photo fits
251     inside of the integration photo.
252     heightPercentageDifference =
253         (double)originalIntegrationPhotoHeight/
254         originalGreenScreenPhotoHeight;
255     widthPercentageDifference =
256         (double)originalIntegrationPhotoWidth/
257         originalGreenScreenPhotoHeight;
258     if(heightPercentageDifference > widthPercentageDifference){
259         heightHasLargerPercentageDifference = true;
260     }
261     else{
262         widthHasLargerPercentageDifference = true;
263     }
264     //===== Start resizing green screen images =====
265
266     if(originalGreenScreenPhotoHeight > newIntegrationPhotoHeight &&
267        heightHasLargerPercentageDifference == true){
268         greenScreenPhotoResizePercentage =
269             (double)newIntegrationPhotoHeight/
270             originalGreenScreenPhotoHeight;
271         newGreenScreenPhotoHeight =
272             (int) (greenScreenPhotoResizePercentage *
273                 originalGreenScreenPhotoHeight);
274         newGreenScreenPhotoWidth =
275             (int) (greenScreenPhotoResizePercentage *
276                 originalGreenScreenPhotoWidth);
277         adjustedGreenScreenPhoto =
278             new BufferedImage(newGreenScreenPhotoWidth,
279                 newGreenScreenPhotoHeight,
280                 BufferedImage.TYPE_INT_ARGB);
```

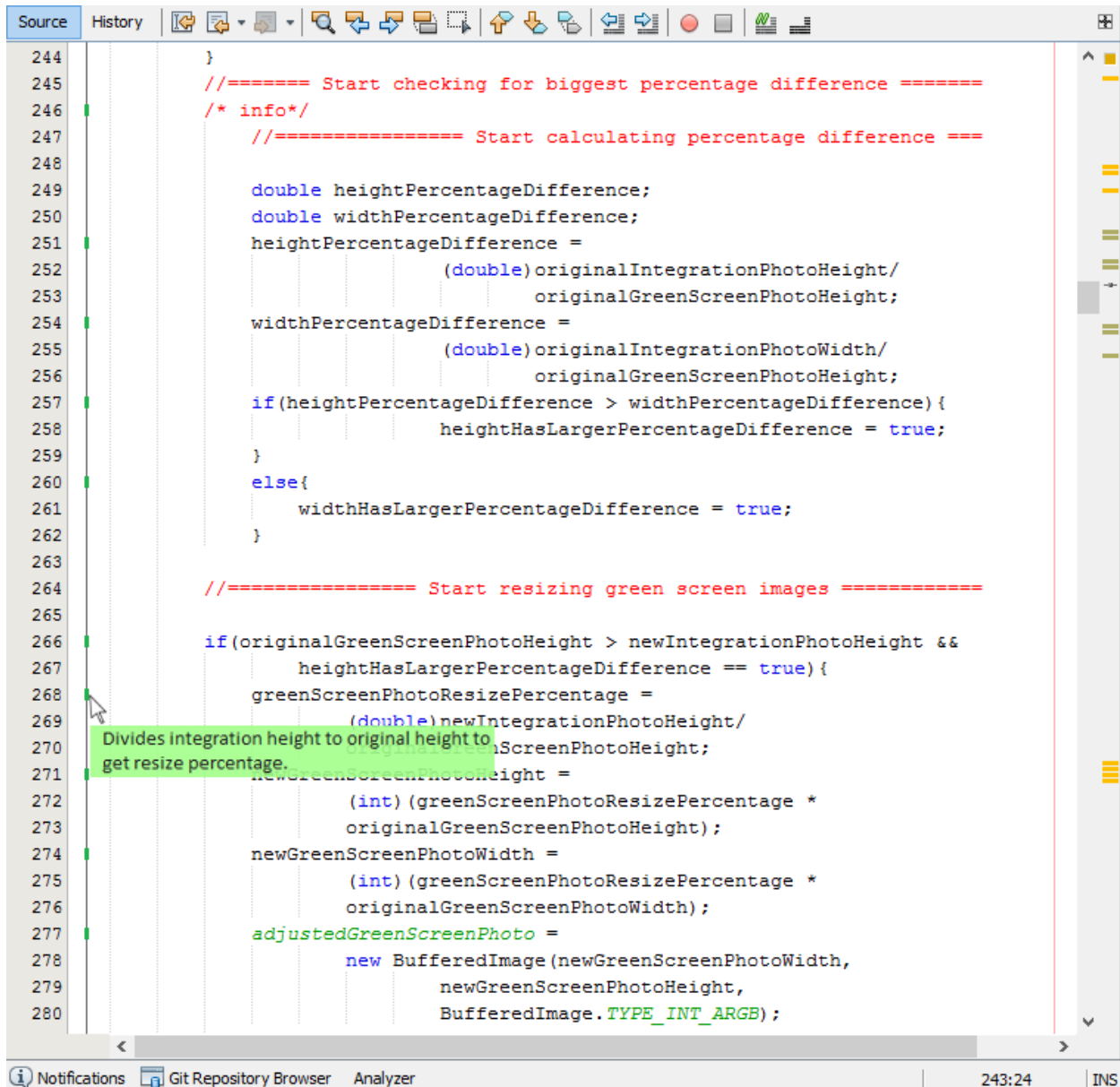
NetBeans interface elements: Source, History, Git Repository Browser, Analyzer, 243:24, INS.

Example 2: Highlighting another marker.



```
244     }
245     //===== Start checking for biggest percentage difference =====
246     /* info*/
247     //===== Start calculating percentage difference ===
248
249     double heightPercentageDifference;
250     double widthPercentageDifference;
251     heightPercentageDifference =
252         (double)originalIntegrationPhotoHeight/
253         originalGreenScreenPhotoHeight;
254     widthPercentageDifference =
255         (double)originalIntegrationPhotoWidth/
256         originalGreenScreenPhotoHeight;
257     if(heightPercentageDifference > widthPercentageDifference){
258         heightHasLargerPercentageDifference = true;
259     }
260     else{
261         widthHasLargerPercentageDifference = true;
262     }
263
264     //===== Start resizing green screen images =====
265
266     if(originalGreenScreenPhotoHeight > newIntegrationPhotoHeight &&
267        heightHasLargerPercentageDifference == true){
268         greenScreenPhotoResizePercentage =
269             (double)newIntegrationPhotoHeight/
270             originalGreenScreenPhotoHeight;
271         newGreenScreenPhotoHeight =
272             (int) (greenScreenPhotoResizePercentage *
273                 originalGreenScreenPhotoHeight);
274         newGreenScreenPhotoWidth =
275             (int) (greenScreenPhotoResizePercentage *
276                 originalGreenScreenPhotoWidth);
277         adjustedGreenScreenPhoto =
278             new BufferedImage(newGreenScreenPhotoWidth,
279                             newGreenScreenPhotoHeight,
280                             BufferedImage.TYPE_INT_ARGB);
```

Example 3: Moving down the to the next comment marker.



The screenshot shows the NetBeans IDE interface. The main editor window displays a Java code file with the following content:

```
244     }
245     //===== Start checking for biggest percentage difference =====
246     /* info*/
247     //===== Start calculating percentage difference ===
248
249     double heightPercentageDifference;
250     double widthPercentageDifference;
251     heightPercentageDifference =
252         (double)originalIntegrationPhotoHeight/
253         originalGreenScreenPhotoHeight;
254     widthPercentageDifference =
255         (double)originalIntegrationPhotoWidth/
256         originalGreenScreenPhotoHeight;
257     if(heightPercentageDifference > widthPercentageDifference){
258         heightHasLargerPercentageDifference = true;
259     }
260     else{
261         widthHasLargerPercentageDifference = true;
262     }
263
264     //===== Start resizing green screen images =====
265
266     if(originalGreenScreenPhotoHeight > newIntegrationPhotoHeight &&
267         heightHasLargerPercentageDifference == true){
268         greenScreenPhotoResizePercentage =
269             (double)newIntegrationPhotoHeight/
270             originalGreenScreenPhotoHeight;
271         newGreenScreenPhotoHeight =
272             (int) (greenScreenPhotoResizePercentage *
273                 originalGreenScreenPhotoHeight);
274         newGreenScreenPhotoWidth =
275             (int) (greenScreenPhotoResizePercentage *
276                 originalGreenScreenPhotoWidth);
277         adjustedGreenScreenPhoto =
278             new BufferedImage(newGreenScreenPhotoWidth,
279                 newGreenScreenPhotoHeight,
280                 BufferedImage.TYPE_INT_ARGB);
```

A green highlight is placed over the comment on line 269: `Divides integration height to original height to get resize percentage.`. A mouse cursor is positioned over this highlight. A suggestion box appears, indicating the comment can be moved to the next comment marker.

The bottom status bar shows "Notifications", "Git Repository Browser", "Analyzer", "243:24", and "INS".