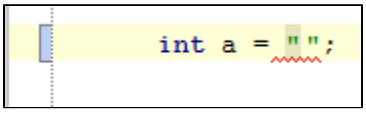
Suppressing Errors

One of very effective ways to maintain high quality of code in MPS is the instant on-the-fly code analysis that highlights errors, warnings or potential problems directly in code. Just like with other code quality reporting tools, it is essential for the user to be able to mark false positives so that they are not reported repeatedly. MPS now provides the language developers with a customizable way to suppress errors in their languages. This functionality was used to implement Suppress Errors intention for BaseLanguage:
One place where this feature is also useful are the generators, since type errors, for example, are sometimes unavoidable in the templates.

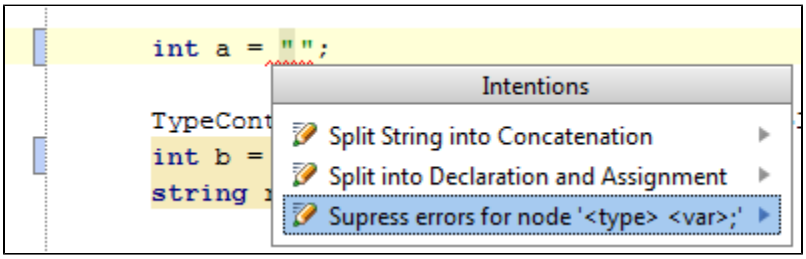
If a node is an instance of a concept, which implements the ISuppressErrors interface, all issues on this node and all its children won't be shown. For example, comments in BaseLanguage implement ISupressErrors. It is also possible to define child roles, in which issues should be suppressed, by overriding the boolean method suppress(node<> child) of the ISupressErrors in terface.
Additionally, if a node has an attribute of a concept that implements ISuppressErrors, issues in such node will be suppressed too. There is a convenience default implementation of an ISuppressErrors node attribute called SuppressErrorsAttribute. It can be applied to only those nodes that are instances of ICanSuppressErrors.

An example of using the SuppressErrorsAttribute attribute and the corresponding intention.

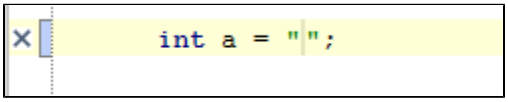
There is an error in editor:



BaseLanguage Statement implements ICanSuppressErrors, so the user can apply the highlighted intention here:



Now the error isn't highlighted any longer, but there is a newly added cross icon in the left pane. The SuppressErrorsAttribute c an be removed either by pressing that cross or by applying the corresponding intention



Previous Next