دانشکده مهندسی کامپیوتر

معرفی و نحوه ی استفاده از ابزار Sonar
در جهت افزایش کیفیت کد برنامه

پژوهش درس مهندسی نرم افزار پیشرفته

در رشته مهندسی کامپیوتر - گرایش نرم افزار

دانشجویان:

محسن امیریان – مرتضی ذاکری – حمیدرضا احمدی

استاد راهنما:

دکتر سعید پارسا

اردیبهشت ماه ۱۳۹۶


بسم الله الرحمن الرحيم

فهرست مطالب

عنوان **صفحه**

فصل ۱:

معرفی Sonar Source

۱-۱- مقدمه

کیفیت کد برنامه موضوعی است که همزمان با اختراع نرم افزار مطرح شد.

از مشکلاتی که یک کد با کیفیت پایین می تواند ایجاد بکند می توان به سرعت پائین تیم توسعه، از بین رفتن برنامه، خرابی ها در زمان تولید، بدنامی برای شرکت و ... اشاره کرد.

SonarSource به دنبال ارائه ی راه حل هایی اســـت که بتواند قابلیت نگهداری، قابلیت اطمینان و امنیت نرم افزار را افزیش دهند.

۱-۱-۱- مشکل قابلیت نگهداری[1]

طبیعت نرم افزار به گونه ای اســـت که در طول زمان تغییر می کند، به این معنی که کدی که امروز ن.شـــته می شـــود، فردا بروز رســـانی می گردد. توانایی، هزینه و زمانی که برای این کار صـــرف می گردد مســـتقیماً با میزان قابلیت نگهداری نرم افزار ارتباط دارد. به عبارت دیگر، قابلیت نگهداری پائین به معنی سرعت پائین تیم توسعه خواهد بود.

قابلیت نگهداری شـــامل مفاهیم متعددی می شـــود، مانند پیمانه بندی، قابل فهم بودن، قابل تغییر بودن، تست پذیری و قابلیت استفاده مجدد.

۱-۱-۲- مشکل قابلیت اطمینان[2]

این مشـــکل به ایرادها[3] و یا کد هایی اشـــاره می کند که در زمان اجرای برنامه رفتارهای غیر قابل انتظار از خود بروز می دهند.

به طور کلی این مشکلات به شکل ایراد های برنامه نویسی مهمی هستند که می توانند باعث مختل شدن یک کسب و کار شوند.

اکثر این مشکلات از طریق آنالیز کردن دقیق کد و اجرای آزمایشی برنامه برای فهمیدن حالت متغییر ها در تمام نقاط برنامه، شناسایی می شوند.

1 - Maintainability Issue

2 - Riliability Issue

۳ - Bugs

۱-۱-۳- مشکل امنیت[1]

این مسئله به آسیب پذیری ها و یا نقص هایی در برنامه اشاره می کند که می تواند منجر به ا ستفاده از نرم افزار در م سیرهایی به غیر از م سیرهای مد نظر طراحان آن گردد.

آسیب پذیری های امنیتی مانند حملات SQL Injection و XSS می تواند به دلیل کد نویسی ضعیف و یا شیوه ی معماری نادرست باشد.

۱-۲-الگوی نشت آب[2] – رویکرد منحصر بفرد در ساخت نرم افزار بهتر

جلوی نشت (چکه کردن) آب را بگیرید، قبل از اینکه بخواهید کف اتاق را تمیز خشک کنید!

زیرا در غیر این صورت در وقت و هزینه خود را هدر می دهید.

در مبحث کیفیت کد برنامه نیز مسئله به همین شکل است. شما می باید شما می باید کیفیت کدهای بتازگی تغییر کرده و یا افزوده شده را قبل از هرچیز مدیریت کنید. زمانی که نشت را تحت کنترل خود بگیرید، کیفیت کد شروع به صورت خودکار بهتر خواهد شد.

۱-۳- محصولات

Sonar Source دارای دو محصول برای کنترل و بهبود کد برنامه است.

sonarlint	sonarqube
برای توسعه دهندگان	برای تیم ها و سازمان ها
درون محیط های برنامه نویسی	بازرسی مستمر کد
حل کردن مشکلات قبل از به وجود آمدن آنها	کنترل نقص های فنی

که در فصل های آتی با آن ها آشنا خواهیم شد.

1 - Security Issue

۲ - Water Leak Paradigm

فصل ۲:

Sonar Lint

۲-۱- مقدمه

SonarLint افزونه ای است که در محیط برنامه نویسی[۱] مد نظر شما نصب می شود و به توسعه دهندگان کمک می کند که یک بازخورد در حین کدنویسی، بر روی ایرادهای جدید و مسائل کیفیتی درون کد داشته باشند.

۲-۲- محیط های برنامه نویسی

همانطور که درتصویر می‌بینیم، می تـوانیم از SonarLint در محیـــط های Visual Studio، Eclipse و یا IntelliJ IDEA استفاده کنیم.

در این تحقیق نحوه ی استفاده از SonarLint در محیط Visual Studio 2015 را بررسی خواهیم کرد.

۲-۳- SonarLint در محیط Visual Studio

SonarLint برای Visual Studio بر مبنای کامپایلر .NET[۲] نوشته شده است. این ابزار با استفاده از API های کامپایلر .NET. محیطی کاملاً یکپارچه برای کدنویسی و آنالیز کد فراهم می کند.

این ابزار رایگان و Open Source بوده و در گالری ویژوال استودیو قابل دسترسی است.

۲-۳-۱- نصب و راه اندازی

برای دریافت ابزار SonarLint می توانیم به وبسایت زیر مراجعه کرده و این ابزار را دانلود کنیم:

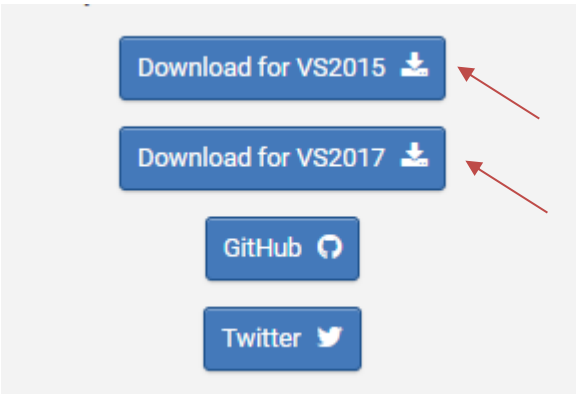
http://www.sonarlint.org/visualstudio/index.html

---

۱ - IDE

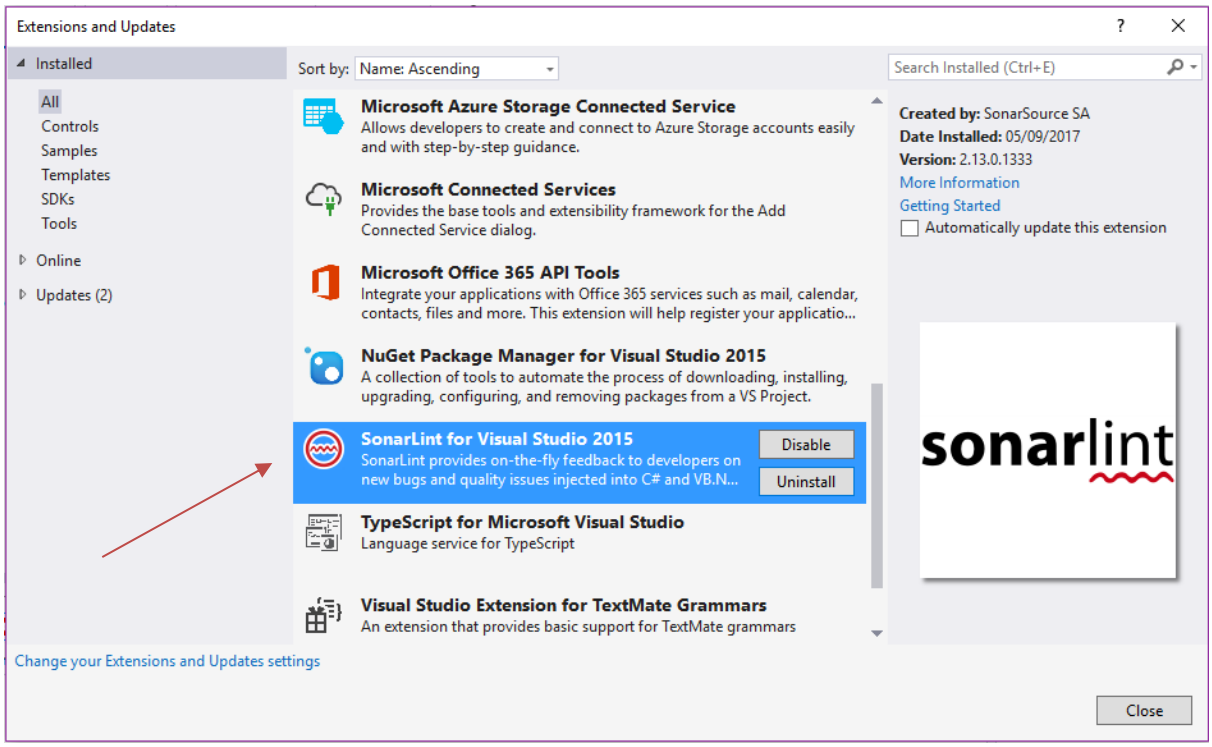
۲ - Roselyn

پس از ورود به این وبسایت می توانیم با توجه به نسخه‌ی ویژوال استودیوی خود، فایل نصبی مد نظر را دانلود کنیم.



همانطور که می‌بینیم برای استفاده از این ابزار می‌باید حتماً از یکی از نسخه های VS 2015 و یا VS2017 استفاده کنیم.

پس از اتمام عملیات نصب، وارد محیط ویژوال استودیو شده و وارد بخش زیر می‌شویم:
Tools => Extentions and Updates…

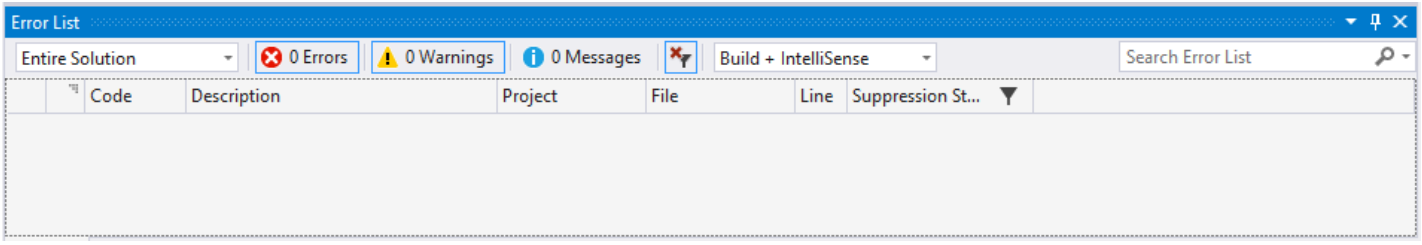


همانطور که مشاهده می‌کنیم این ابزار به درستی نصب شده و در این بخش قابل مشاهده است.
با استفاده از گزینه های Disable و Uninstall می‌توانیم آن را غیرفعال و یا حذف کنیم.

۲ـ۳ـ۲ـ نحوه ی استفاده

نحوه‌ی استفاده از این ابزار را در محیط C#.NET بررسی خواهیم کرد.

ویژوال ا ستودیو در حالت معمول (هنگامی که ابزار SonarLint بر روی آن نصب نیست) دارای پنجره ای با عنوان Error List اســت. در این صــفحه می‌توانیم خطاها و یا هشــدارهایی که در هنگام کدنویســی و یا کامپایل کردن برنامه به وجو آمده اند را مشاهده کنیم:

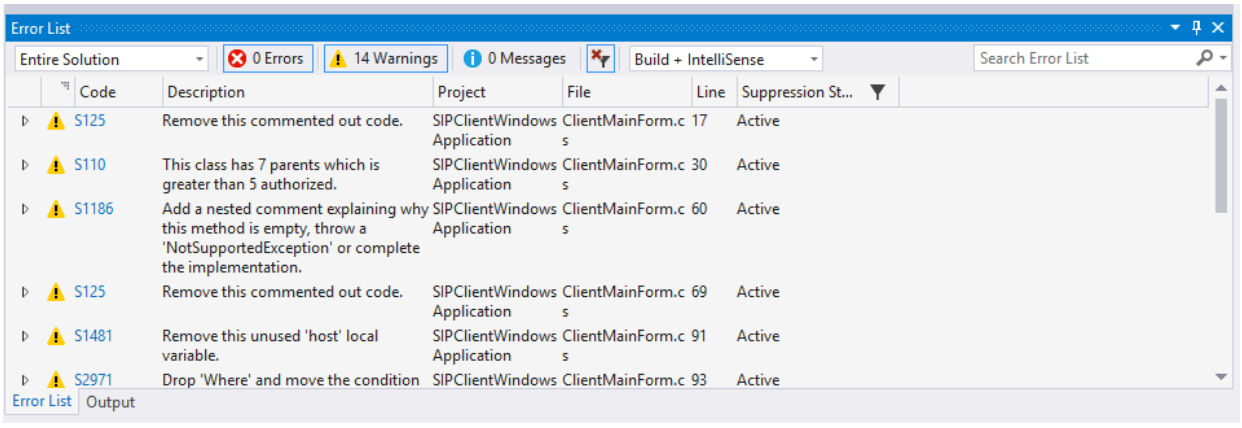


همانطور که می‌بینیم این پنجره دارای بخشــی با عنوان Errors و بخشــی هم با عنوان Warnings اســت. Error ها آن دســته از خطاهایی هســتند که در صــورت برطرف نشــدن، برنامه قابل اجرا نخواهد بود. اما Warning ها هشدارهایی هستند که در صورت وجود، امکان اجرای برنامه وجود دارد. اما بهتر ا ست برطرف گردند.

به عنوان مثال درصــورتی که از متغیری تعریف نشــده اســتفاده شــود، به عنوان Error خواهد بود. اما اگر متغیری تعریف شده ولی از آن استفاده نگردد در برنامه، این موضوع در بخش Warning مطرح خواهد شد.

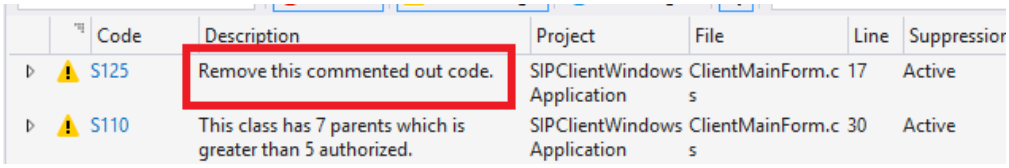
پس از نصــب ابزار SonarLint، تعداد Warning های اعلام شــده در پنجره‌ی بالا معمولاً از حالت معمول بیشتر خواهد بود. این به این دلیل است که این ابزار به شکل دقیق تری کد برنامه را آنالیز کرده و در راستای بهبود هرچه بیشتر کیفیت کد، پیشنهادات بیشتری را در بخش Warnings به برنامه نویسان ارائه می‌کند.

در مثال زیر نمونه‌ای از Warning های ایجاد شده توسط این ابزار را مشاهده می‌کنیم:

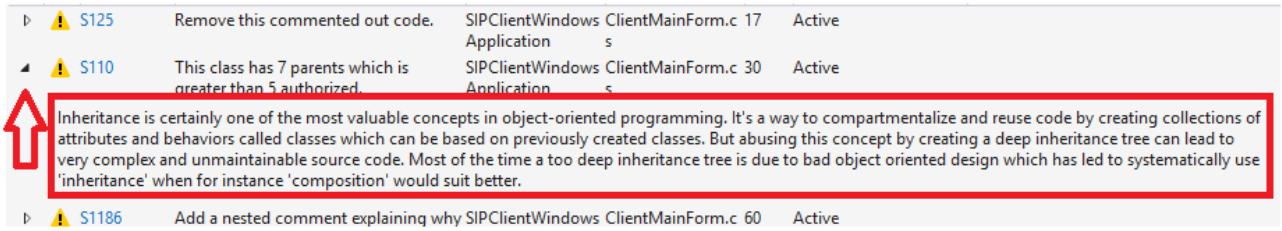


در تصویر بالا مشاهده می‌کنیم که ۱۴ Warning شناسایی شده اند. در کنار هریک از آن‌ها یک کد (مثلاً S125) نوشته شده که نشان دهنده‌ی نوع آن Warning است. همچنین توضیح مختصری از هر Warning را

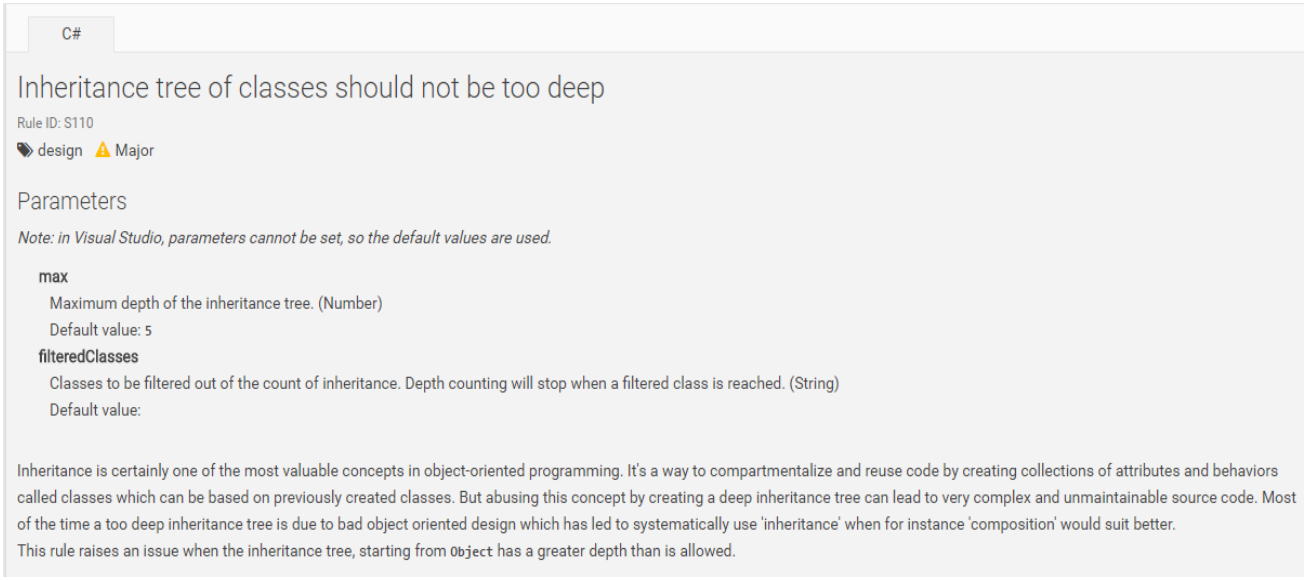
در کنار آن مشاهده می‌کنیم.



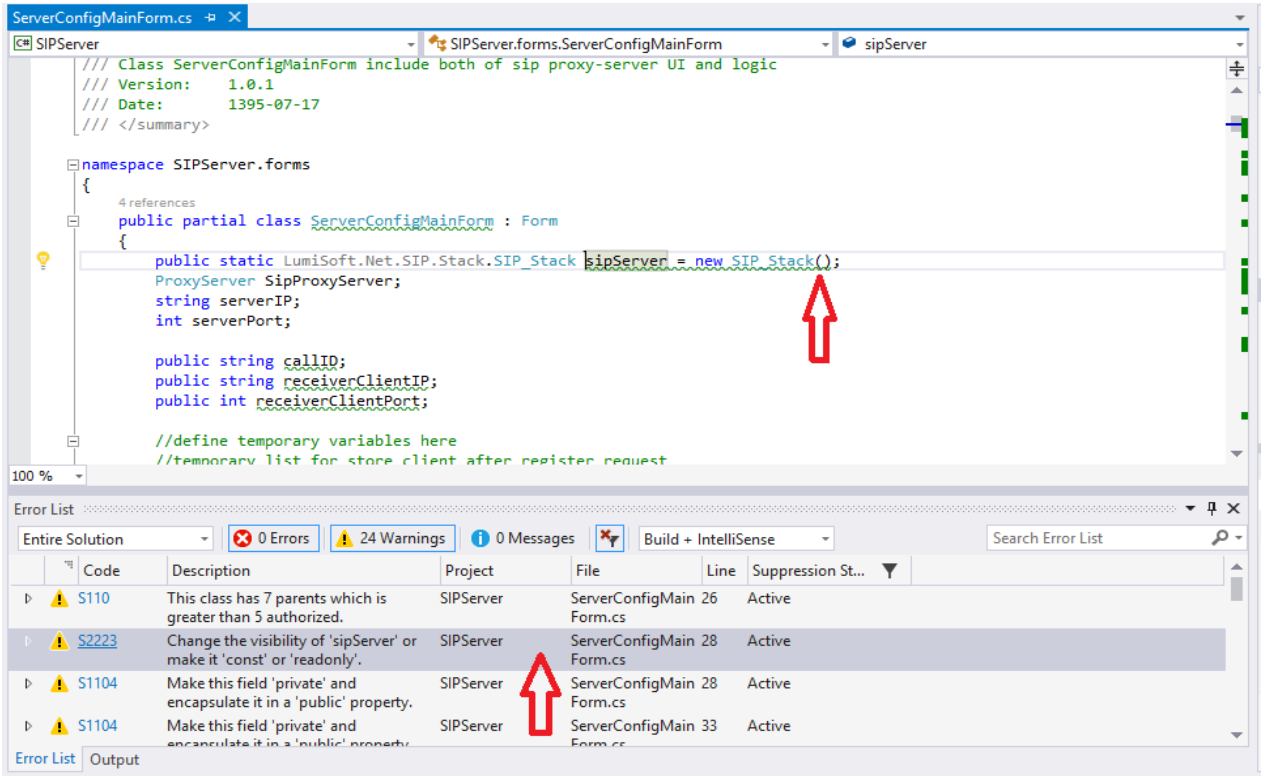
با کلیک کردن بر روی مثلث کنار هر کد، می‌توان توضیحات بیشتری در مورد آن Warning مشاهده کرد:



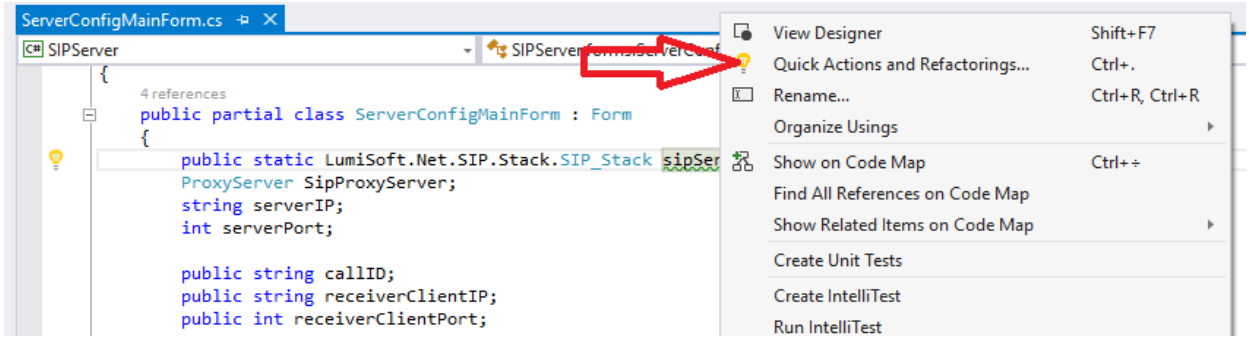
اگر مایل به مطالعه‌ی بیشتری درمورد یک Warning هستید، می‌توانید بر روی کد آن (مثلاً S110 در تصویر بالا) کلیک کنید. با این کار وارد وبسایت SonarLint شده و در مورد آن Warning می‌توانید توضیحات مفصل تری را مشاهده کنید:



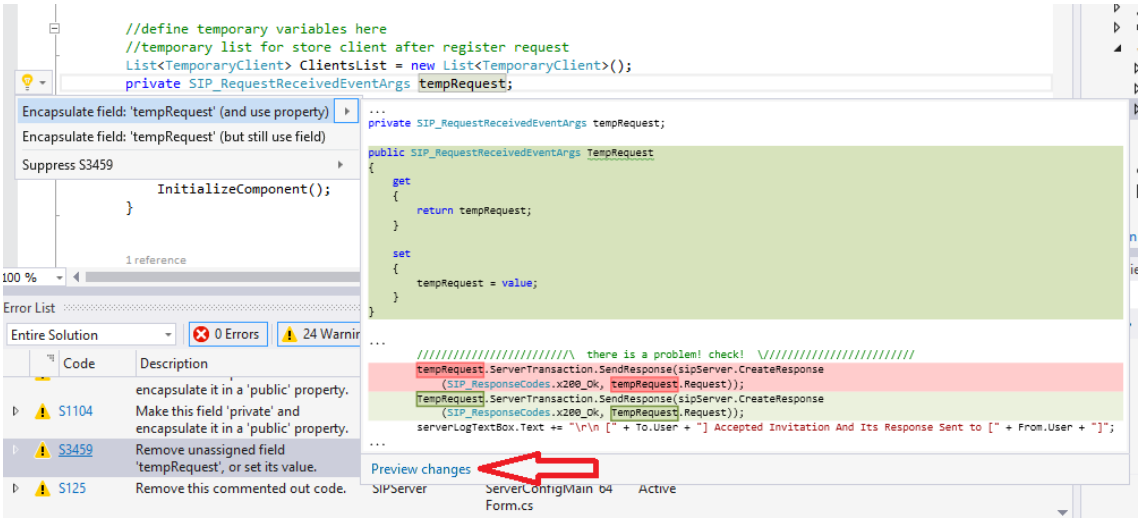
پس از اینکه در مورد یک Warning مطالعه کرده، می‌توانیم آن را برطرف کنیم. برای این منظور می‌توانیم در پنجره‌ی Error List بر روی آن Warning دابل کلیک کنید تا به بخشی از کد که مربوط به آن است منتقل شوید:

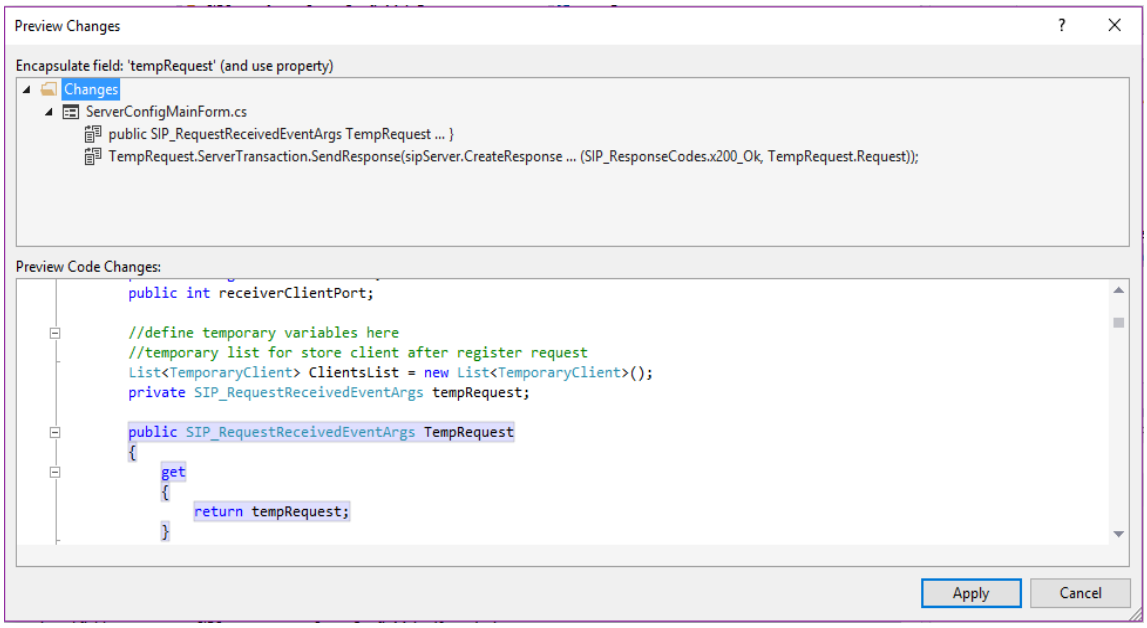


پس از اینکه به محل مورد نظر منتقل شدیم، مشاهده می‌کنیم که زیر بخشی از کد برنامه خط سبز رنگ کشیده شده است. این بخش همان بخش مد نظر SonarLint است که می‌باید ویرایش گردد. می‌توانید خودتان این بخش را ویرایش کنید، اما SonarLint خود پیشنهاداتی برای حل این مشکلات ارائه می‌کند. برای استفاده از این پیشنهادات بر روی همان بخش هایلایت شده از کد کلیک راست کرده و گزینه‌ی Quick Actions and Refactorings… را انتخاب می‌کنیم:



پس از انتخاب این گزینه، SonarLint پیشنهاداتی را ارائه می‌دهد. پس از انتخاب هر کدام و کلیک بر گزینه‌ی Preview Changes، می‌توانید پیش نمایش تغییراتی که بر روی کد برنامه صورت خواهد گرفت را مشاهد کنید:





با انتخاب گزینه‌ی Apply، تغییرات بر روی کد برنامه صورت می‌گیرد.