دانشکده‌ی مهندسی کامپیوتر

بررسی ابزار JRefactory

گردآورنده:

الهه قصابانی

۸۹۷۲۳۱۹۸

استاد:

جناب آقای دکتر سعید پارسا

دی ۱۳۸۹

فهرست مطالب

۱. مقدمه

درخت های خلاصه نحوی یک ابزار پر قدرت هستند، که کامپایلرها برای مقاصد مختلف از جمله تولید کد میانی، از آن می‌توانند استفاده نمایند. بنابراین ما به دنبال ابزاری بودیم تا بتواند درخت خلاصه نحوی را نمایش دهد. طبق جستجوهایی که انجام شد، ابزارهایی با عناوین ذیل را برای تولید درخت خلاصه نحوی یافتیم:

۱- Javacc
۲- Langtools
۳- JRefactory tools

Javacc یک compiler generator است که قدرت زیادی دارد اما کار کردن با آن پیچیده است و قواعد خاص خود را دارد. توسط این ابزار شما برای هر گرامری قادر به تولید AST هستید. اما این تولید کد مستلزم این است که بر قواعد javacc کاملاً مسلط باشید.

Langtools نیز ابزار دیگری است که برای تولید درخت خلاصه نحوی استفاده می‌شود.

از میان این ابزارها، JRefactory ابزاری است پر قدرت که نصب و استفاده از آن راحت‌تر بوده و از طرفی دیگر، علاوه بر تولید AST، دارای قابلیت‌های دیگری نیز هست. اما با این همه یک نقطه‌ی ضعف آن نسبت به Javacc این است که Javacc قادر به تولید درخت خلاصه نحوی برای تمام گرامرهایی است که با قواعدش تعریف می‌شوند، درحالی که JRefactory درخت خلاصه نحوی را فقط برای انواع کدهای جاوا تولید می‌کند.

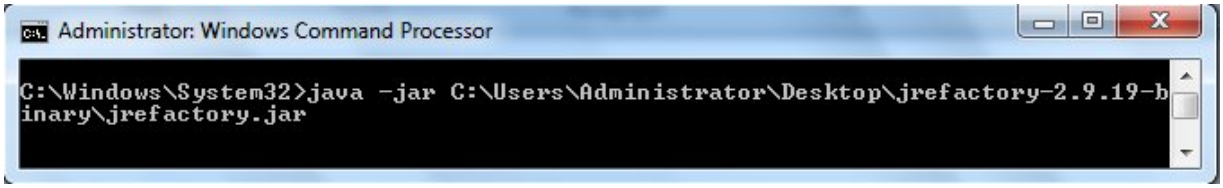
در این گزارش ما نصب و چگونگی کار با ابزار JRefactory را با ارائه‌ی مثال‌هایی کامل توضیح می‌دهیم. سپس نصب و راه‌اندازی ابزار Javacc را با ارائه‌ی یک مثال توضیح خواهیم داد. شرح ابزار Javacc بسیار فراتر از آنچه که در اینجا می‌خوانید می‌باشد.

۲. نصب و اجرای ابزار JRefactory

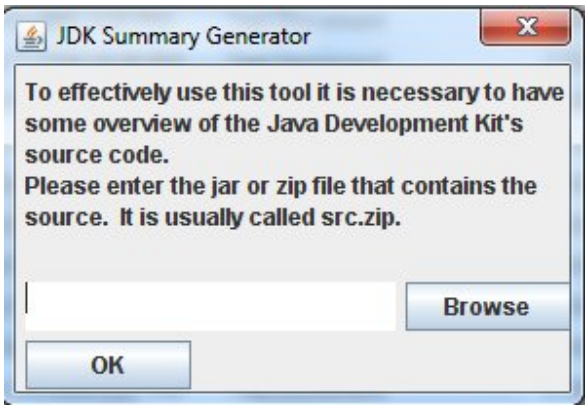
این ابزار از آدرس http://sourceforge.net/projects/jrefactory قابل بارگذاری است. ما بسته‌ی باینری این ابزار را از آدرس http://surfnet.dl.sourceforge.net/project/jrefactory/JRefactory/2.9.19/jrefactory-2.9.19-binary.zip دانلود نمودیم. پس از آن فایل zip. را از حالت فشرده خارج نمایید و محتویات آن را در یک

پوشه قرار دهید.

اکنون از خط فرمان برنامه‌ی JRefactory را اجرا نمایید:

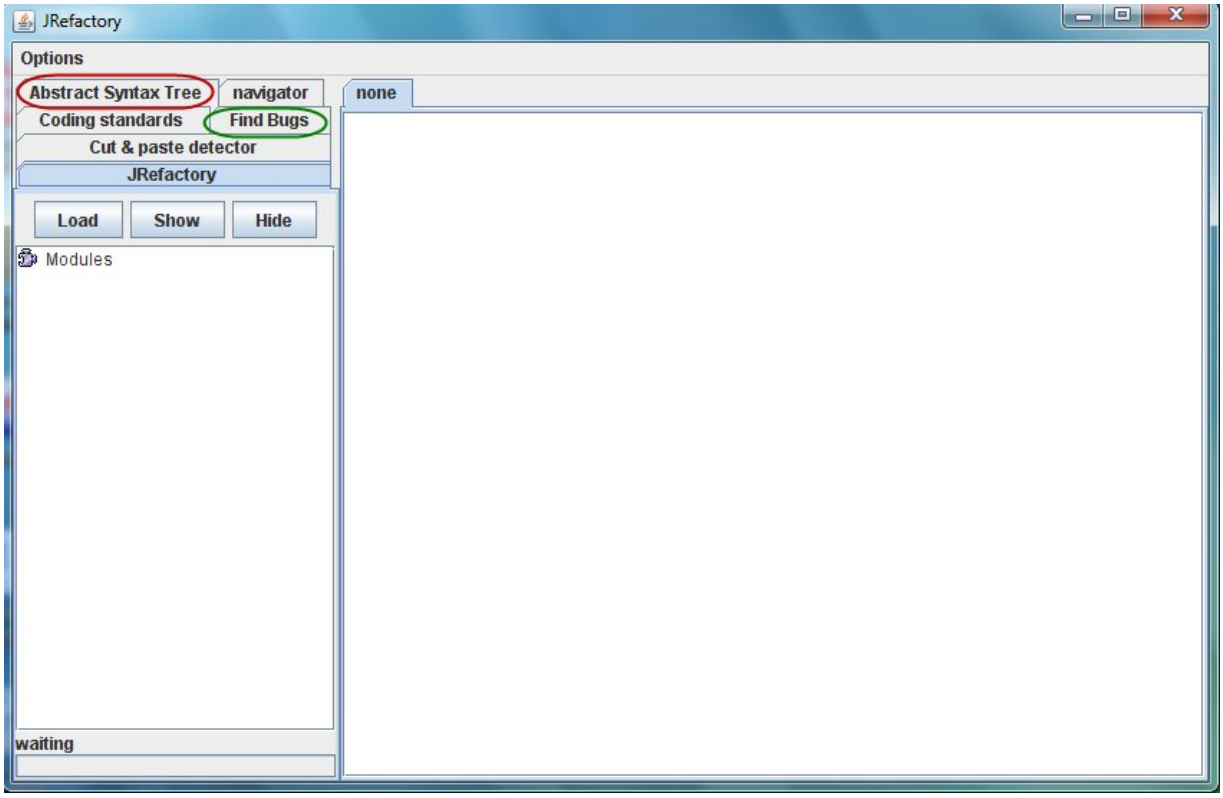


حال با پیغام زیر مواجه می‌شوید:



این پیغام از شما می‌خواهد تا در مسیری که ماشین مجازی جاوا (JDK) را نصب نموده‌اید، فایـل src.zip را مشخص کنید. اگر تمایلی به این کار ندارید، پنجره را ببندید. بعد از بستن پنجره، محیط برنامـه‌ی JRefactory را مشاهده خواهید نمود.

در شکل زیر نمایی از این برنامه را مشاهده می‌کنید.



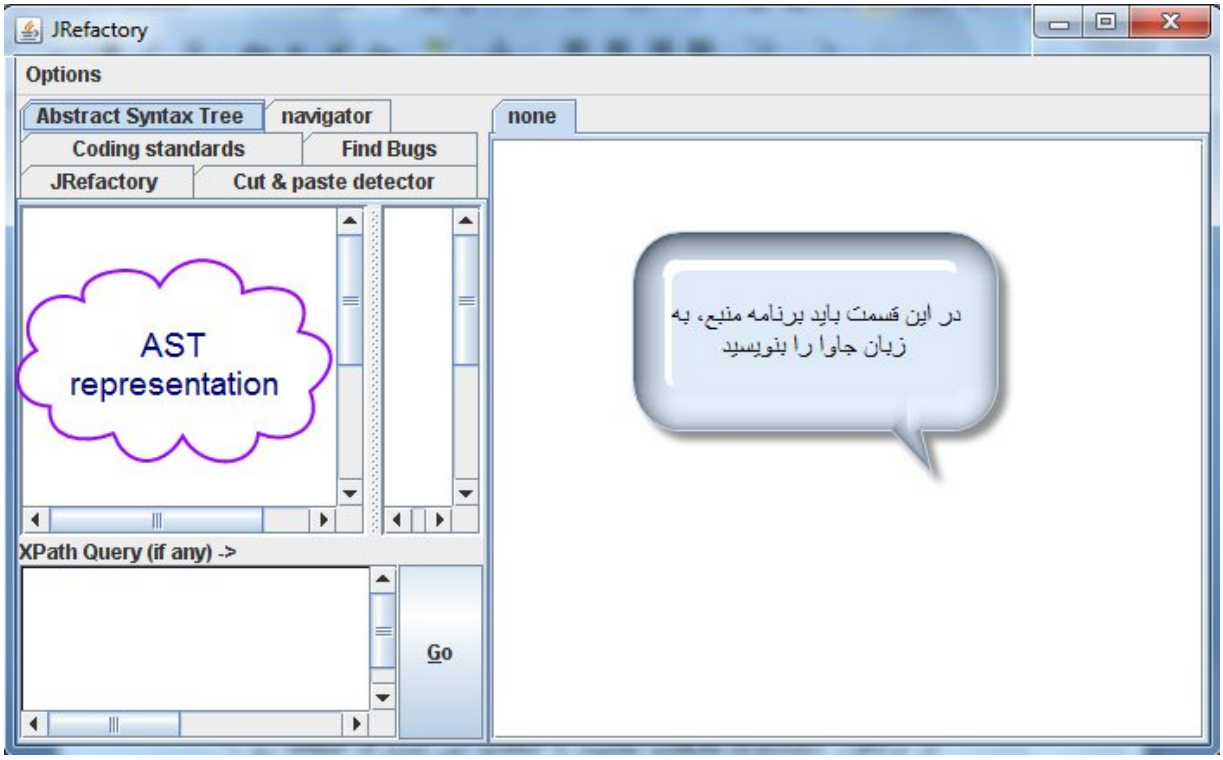
همانطور که در شکل فوق مشاهده می‌نمایید، این ابزار دارای بخش‌های متعددی است که مهمترین آنها عبارتند از:

- Abstract Syntax Tree
- Find Bugs

که در این گزارش به بررسی این دو بخش می‌پردازیم.

۳. تولید درخت خلاصه نحوی توسط ابزار JRefactory

در نرم‌افزار JRefactory بخش Abstract Syntax Tree را انتخاب نمایید. با انتخاب این سربرگ صفحه زیر را مشاهده خواهید نمود:



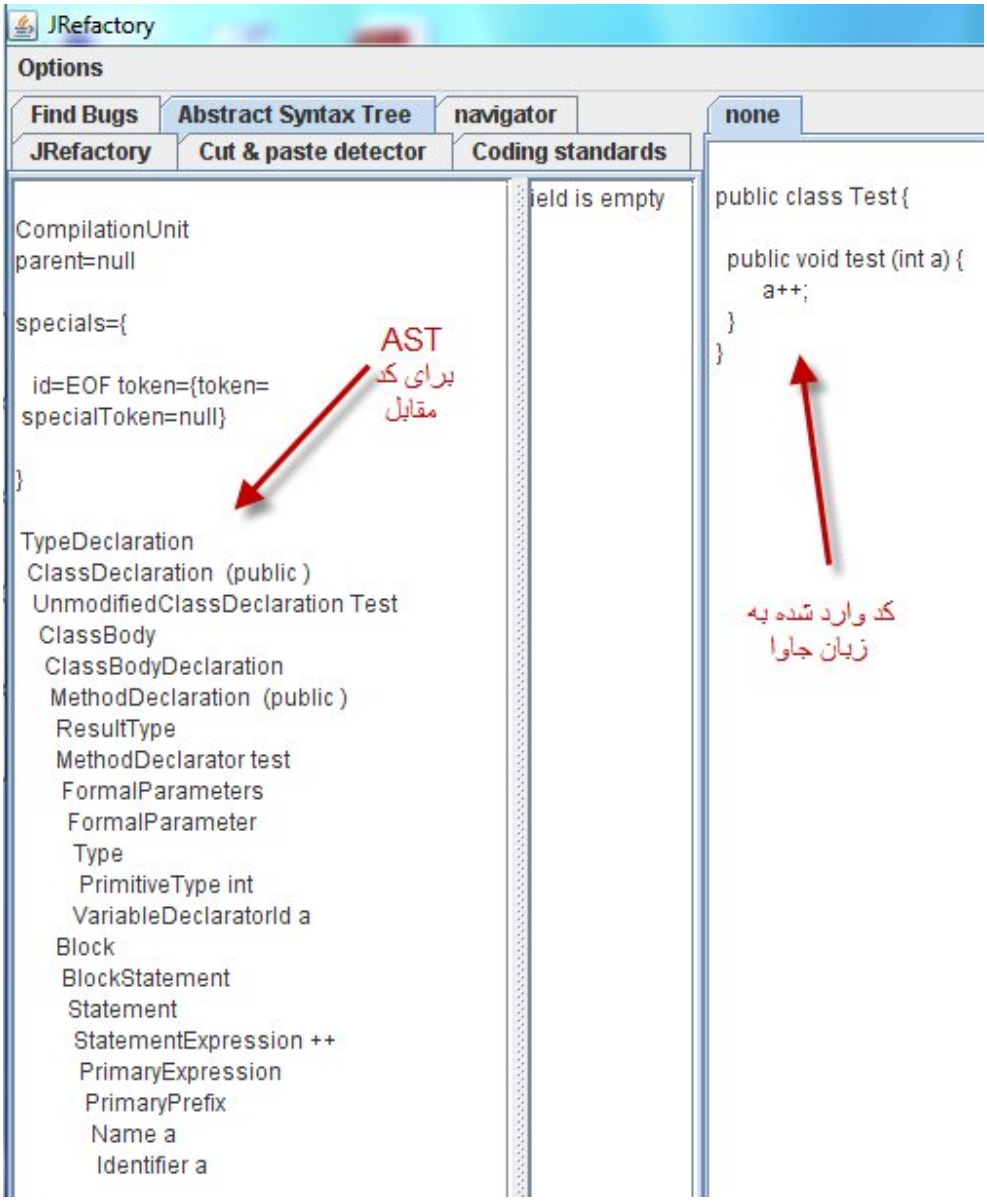
همانطور که در شکل فوق مشاهده می‌نمایید، در قسمت none، باید برنامه منبع به زبان جاوا را وارد کنید. پس از آن با زدن دکمه‌ی "Go" درخت خلاصه نحوی در مکان علامتگذاری شده روی شکل، نمایش داده خواهد شد. در ادامه مثال‌هایی را اجرا می‌کنیم.

این ابزار درخت خلاصه نحوی را برای برنامه هایی که به زبان جاوا نوشته شده‌اند، رسم می‌کند. همانطور که می‌دانید تمام برنامه‌ها در جاوا با یکی از کلمات کلیدی زیر آغاز می‌شوند:

"abstract"، "import"، "interface"، "package"، "public"، "strictfp"، "@"، "final"، "class"، "protected"، "private"، "enum".

بنابراین چناچه در کد نوشته شده قواعد نحوی زبان جاوا را رعایت نکرده باشید، با پیغام رخداد خطا و شماره‌ی سطر و ستون خطای نحوی رخداده و پیغام "Was expecting one of..." مواجه خواهید شد. پس به خاطر داشته باشید که برای مشاهده درخت خلاصه نحوی، کدی که می‌نویسید باید عاری از خطای نحوی باشد.

۳,۱. ارائه مثال

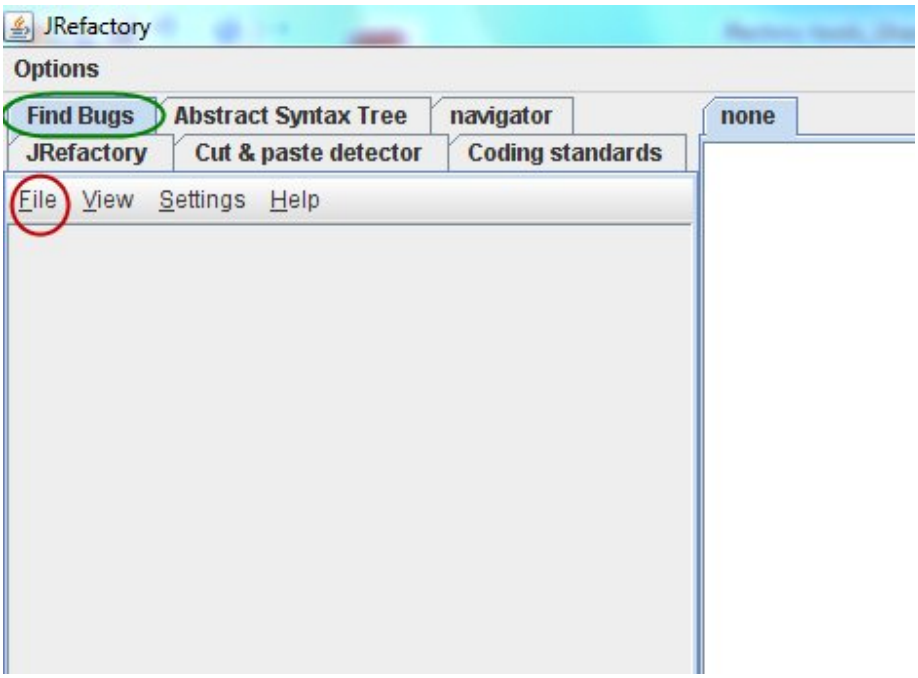


یک قابلیت این ابزار این است که روی درخت تولید شده می‌توان پرسش‌های XPath وارد نمود و جستجو انجام داد. برای این کار باید پرسش خود را در بخش XPath Query وارد نمایید.

به دلیل اینکه کدی که می‌نویسیم هرقدر طولانی‌تر شود، درخت تولیدی هم بزرگتر می‌شود، امکان ارائه‌ی مثال‌های بیشتری با استفاده از snapshot برنامه وجود ندارد. اما به ترتیبی که ارائه شد می‌توانید برنامه را اجرا و از آن استفاده نمایید.

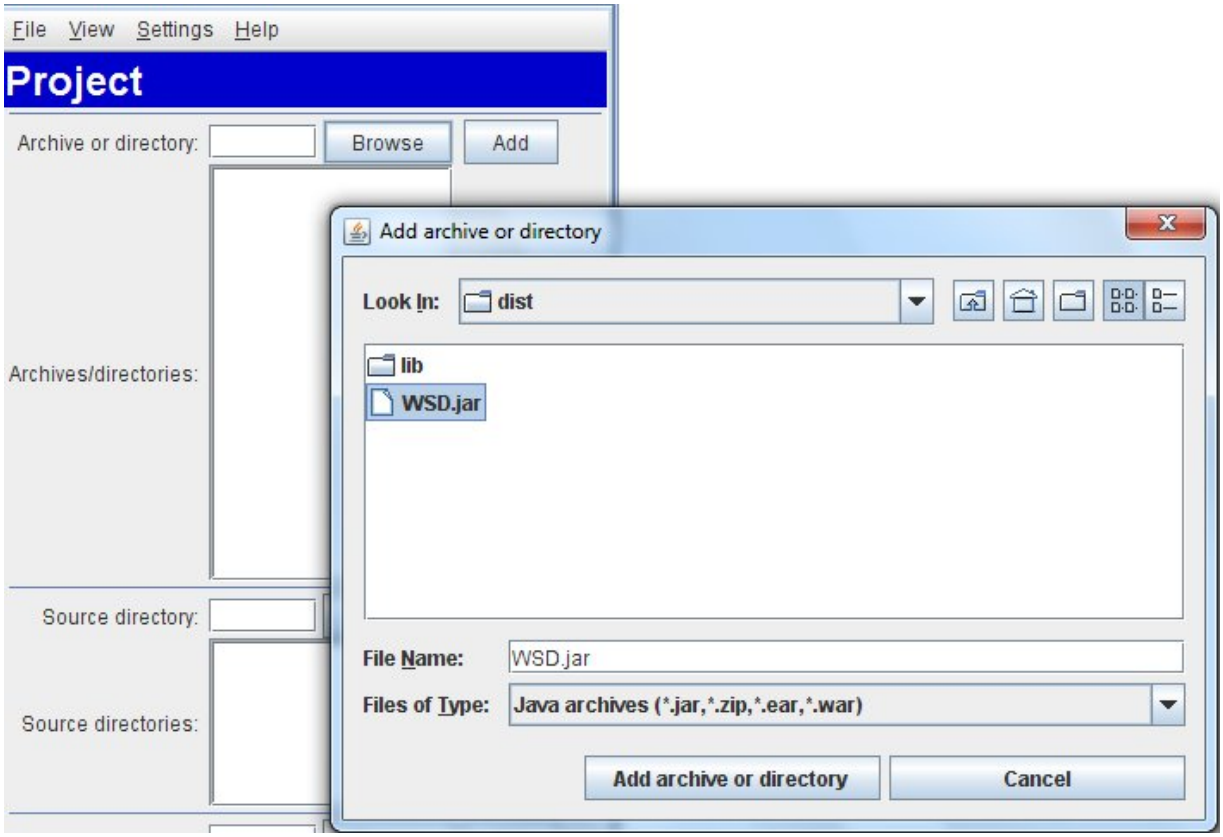
۴. خطایابی برنامه‌های جاوا با استفاده از JRefactory

ابزار JRefactory دارای بخشی به نام "Find Bugs" بوده که قادر به خطایابی پروژه‌های نوشته شده به زبان جاوا است. این خطایابی با استفاده از تحلیل استاتیک کد انجام می‌شود. برای کار با این قسمت، به بخش Find Bugs رفته و از منوی File، New Project را انتخاب نمایید.

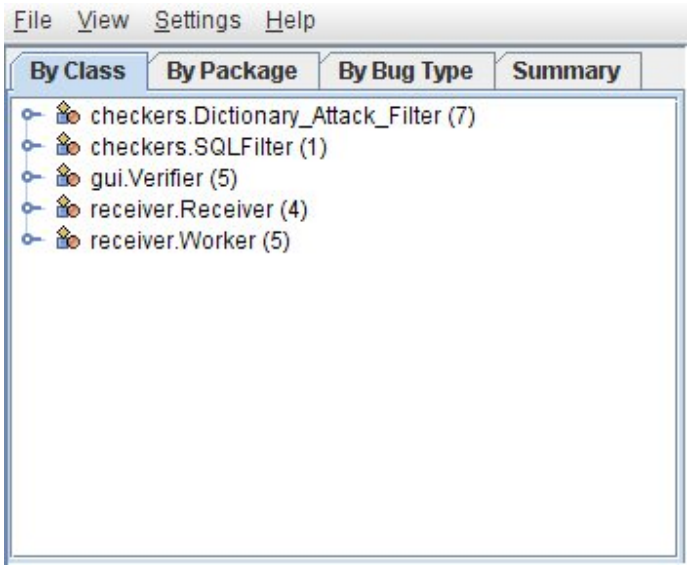


بعد از انتخاب New Project صفحه زیر باز می‌شود:

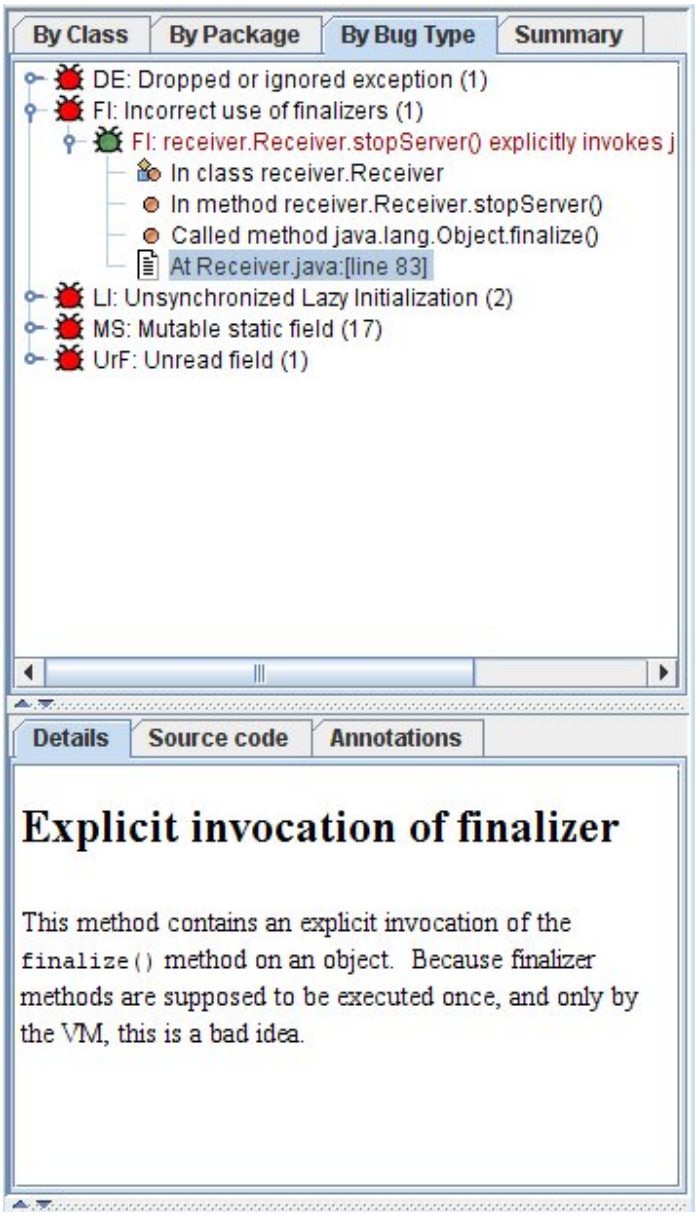
در قسمت اول (Archive or directory)، فایلی که قصد خطایابی آن را دارید وارد نمایید. مثلاً ما یک فایل jar. را وارد نموده‌ایم. بعد از وارد نمودن فایل مورد نظر، دکمه‌ی "Find Bugs!" را بزنید تا کد تحلیل شود. نتایج تحلیل برای شما روی صفحه نمایش داده می‌شود.



پس از اینکه تحلیل انجام گرفت، خطاهای یافته شـده در چهـار دسـته‌ی by class، by package، by bug و type و summary خلاصه‌بندی می‌شوند. به شکل زیر توجه نمایید:



چناچه روی هر پیغام خطا کلیک نمایید، در قسمت Details شرح آن خطا را نیز مشاهده خواهید نمود:



توانایی این ابزار در یافتن خطاها و نوع خطاهایی که تشخیص می‌دهد مانند ابزار PMD است که قبلاً گزارش مربوط به PMD را مشاهده نموده‌اید. اما اغلب کاربران اتفاق نظر دارند که Find Bugs از PMD قوی‌تر عمل می‌نماید.

۵. نصب و راه اندازی ابزار Javacc

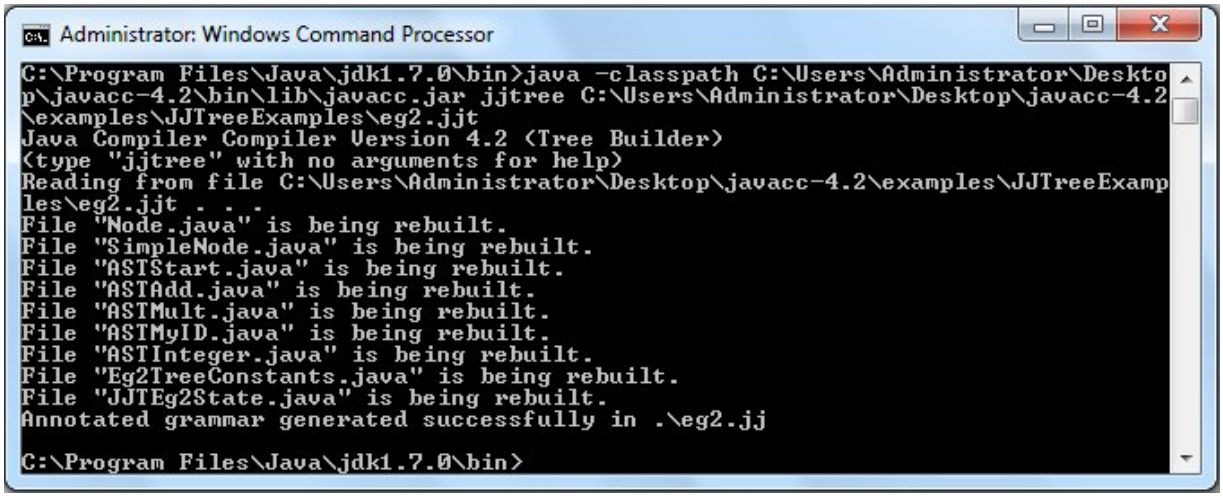
برای نصب محتویات فایل zip. مربوطه را extract نمایید. سپس به مسیر bin/lib/ بروید. برنامه‌ی Javacc.jar را مشاهده می‌نمایید. Javacc.jar دارای سه کلاس اجرایی javacc، jjtree و jdoc می‌باشد. برای تولید درخت‌های خلاصه نحوی، ما نیاز به استفاده از دو کلاس jjtree و javacc داریم.

استفاده از Javacc را با یک مثال که از پوشه‌ی examples انتخاب نموده‌ایم، شرح می‌دهیم. مثال ذیل از پوشه-ی JJTreeExamples انتخاب شده است.

در خط فرمان عبارتی که در شکل مشاهده می‌نمایید را وارد کنید:

در شکل فوق خط زرد رنگ دستور اجرایی برنامه جاوا است. قسمت بنفش، آدرس مکانی است که فایل javacc.jar در آن قرار داده شده است، خط سبز رنگ نشان می‌دهد که از کلاس jjtree استفاده می‌کنیم و نهایتاً بخش صورتی رنگ آدرس فایل با پسوند jjt. را نشان می‌دهد. در این فایل گرامر مربوط به زبانی که قصد مشاهده‌ی AST آن را داریم، باید با قواعد javacc نوشته شده باشد.

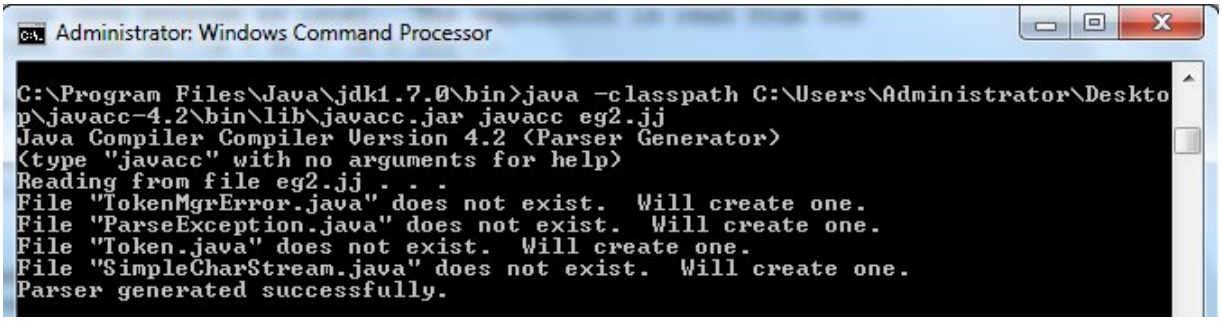
پس از نوشتن دستور فوق کلید Enter را بزنید تا پیغام زیر را مشاهده کنید:



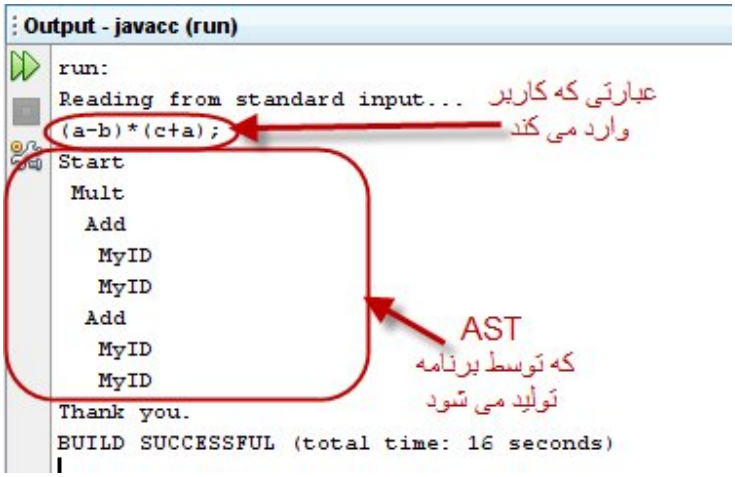
این پیغام کلاس‌هایی را که لیست کرده، در مسیری که jdk نصب است، تولید می‌نماید.

در قدم بعد باید دستور زیر (که در شکل نشان داده شده است) را وارد نمایید:

%Javacc_Home% \javacc.jar javacc filename.**jj**



اکنون برنامه برای شما فایل منبع جاوا با نام مثلاً eg2.java را ساخته است، با دستور javac، این فایل را کمپایل نمایید، سپس با دستور java فایل class. ساخته شده را اجرا کنید. ما این برنامه را با استفاده از netbeans کامپایل و اجرا نموده‌ایم. در زیر خروجی برنامه را مشاهده می‌کنید. این برنامه یک عبارت ریاضی دریافت نموده و AST آن را رسم می‌کند:



توجه: برای تولید برنامه خودکار که هر عبارتی را بخواند و AST نظیر آن را رسم کند، باید قواعد نرم‌افزار javacc را فرا گرفته و مطابق دستورالعمل‌های آن، فایل jjt. را بنویسید. آنگاه می‌توانید مطابق راهنمایی‌های ارائه شده، درخت خلاصه نحوی برای عبارات مطابق گرامر را مشاهده نمایید.

ما در این گزارش فرض را بر این قرار دادیم که فایل jjt. را در اختیار داریم.