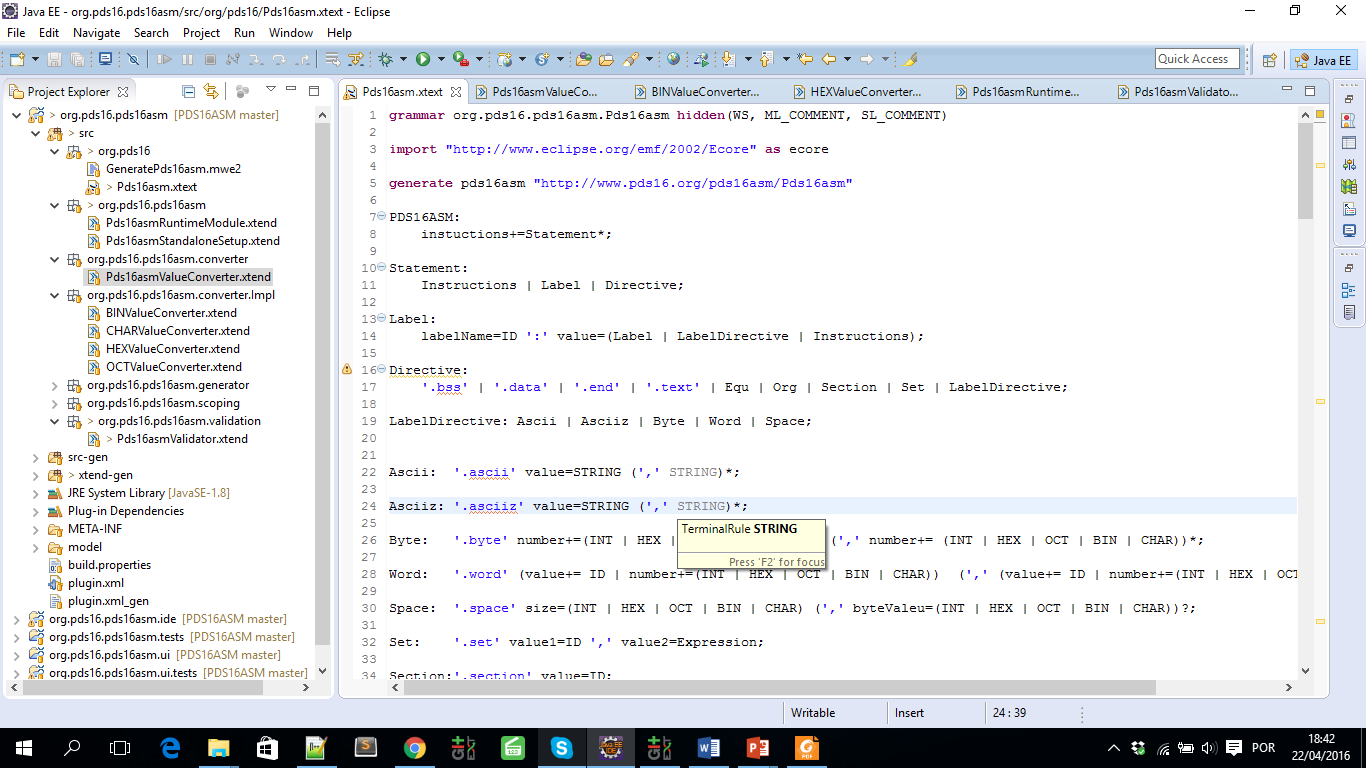
Xtext é uma framework para o desenvolvimento de linguagem de programação, as chamadas DSL, domain-specific languages. Com o Xtext é possível definir uma linguagem com toda a sua gramatica resultando uma infraestrutura que incluí parser, linker, typechecker, compilador e também a possibilidade de ter um editor através do Ecplise, Intellij IDEA e também através do browser.

Esta framework foi desenvolvida com o intuito de ser fácil de aprender e ser possível em poucos minutos descrever uma linguagem simples e também ser possível extrair o projeto em forma de um plugin para a sua portabilidade entre máquinas.

Esta framework foi posta em prática para a realização de um plug-in para uma linguagem de assembly: PDS16. Não havendo nenhum editor de texto para a mesma, comprometemos a usar esta biblioteca.

Para começar a trabalhar tivemos de instalar o plug-in da framework no nosso IDE neste caso o Ecplise. Após criar um novo projeto e definir o nome da linguagem e a sua extensão, podemos começar a desenvolver. O primeiro passo é definir a sintaxe da linguagem, ou seja definir as regras. Uma regra pode ser conjugada de várias formas através de palavras-chaves e de referências para outras regras. Como por exemplo:



Statement é uma regra que em que a sua definição é uma das referências para outra regra. Neste caso se virmos a regra Label podemos ver que a sua definição já contem palavras-chaves como “:” e um identificador “labelName” que é o tipo ID considerado um terminal. Terminal é também uma regra mas só que esta definida por uma sequência de caracteres também chamadas por tokens rules ou lexer rules. Um terminal pode retornar um tipo, por definição eles retornam sobre a forma de String: ecore::EString. Mas é possível converter o tipo de retorno para um típico especifico desque seja uma instancia de ecore::EDataType. Para isso é necessário implementar a interface IValueConverter e criar o respetivo converter de String para o tipo pretendido.



O primerio terminal, ID, começa com um caracter de ‘a’ a ‘z’ ou por um ‘\_’ seguindo de nenhum ou mais caracteres incluindo números. O terminal HEX é a definição de um numero hexadecimal, mas retornando um numero inteiro em vez da String. Para isso foi criado a classe “HEXValueConverter” que extende de AbstractLexerBasedConverter<?> onde é feito override do método “toValue” para fazer a conversão de String para Int.

Existem certas regras de uma linguagem que não podem ser definidas, logo essas tem que ser verificadas no ato da compilação. Mas tal como um editor de texto o Xtext permite que sejam feitas essas verificações ao decorrer da escrita do código indicando o erro. Os validadores da framework permitem analisar determinado conteúdo e indicar ao utilizador do erro, retirando essa função ao compilador, pois não é possivel compilar com erros de validações. No caso do nosso no trabalho verificamos os limites dos números conforme o tipo, por exemplo o offset 8 que só pode estar compreendido entre minimio valor a 8 bits com sinal e o máximo valor a 8 bits com sinal.

