# Orientações gerais

1. Será criada uma classe chamada TS para a tabela de símbolos
2. Serão criadas duas classes para o analisador sintático:
   1. Parser - classe abstrata de base para diferentes parsers
   2. PPR - Parser Preditivo Recursivo - implementação concreta
   3. No futuro será implementado um parser LL(1) na classe LL1
3. Utilizar o mesmo projeto do analisador léxico feito em aula

# 2. Tabela de Símbolos

1. Criar a classe TS que implementa a tabela de símbolos da LPD
   1. A classe conterá um atributo chamado “ts” do tipo HashMap ([ref.](https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html))
   2. As [operações](https://docs.google.com/presentation/d/1nspw5ug5VzcabxfiXiKrRra_7n_7HcKOugmFMMf_Ol0/edit#slide=id.g600943ddd4_0_85) insere e busca já existem em HashMap
   3. Será necessário criar métodos setAtributo e getAtributo
2. Criar (new) o objeto ts no construtor da classe TS
3. Implementar os métodos setAtributo e getAtributo
   1. Quais os parâmetros de cada um?
   2. Utilizar [esta referência](https://stackoverflow.com/a/2043485/2396999) para obter e setar o valor do atributo do objeto.

# 3. PPR - Parser Preditivo Recursivo para LPD

1. Criar uma classe chamada Main\_PPR.java com o método main que cria e executa o analisador sintático.

PPR ppr = new PPR("src/prog1.lpd");

ppr.parse();

1. Criar a classe abstrata Parser com os seguintes atributos:

TS ts;

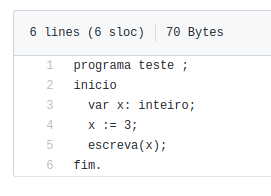
Lexer3 lexer;

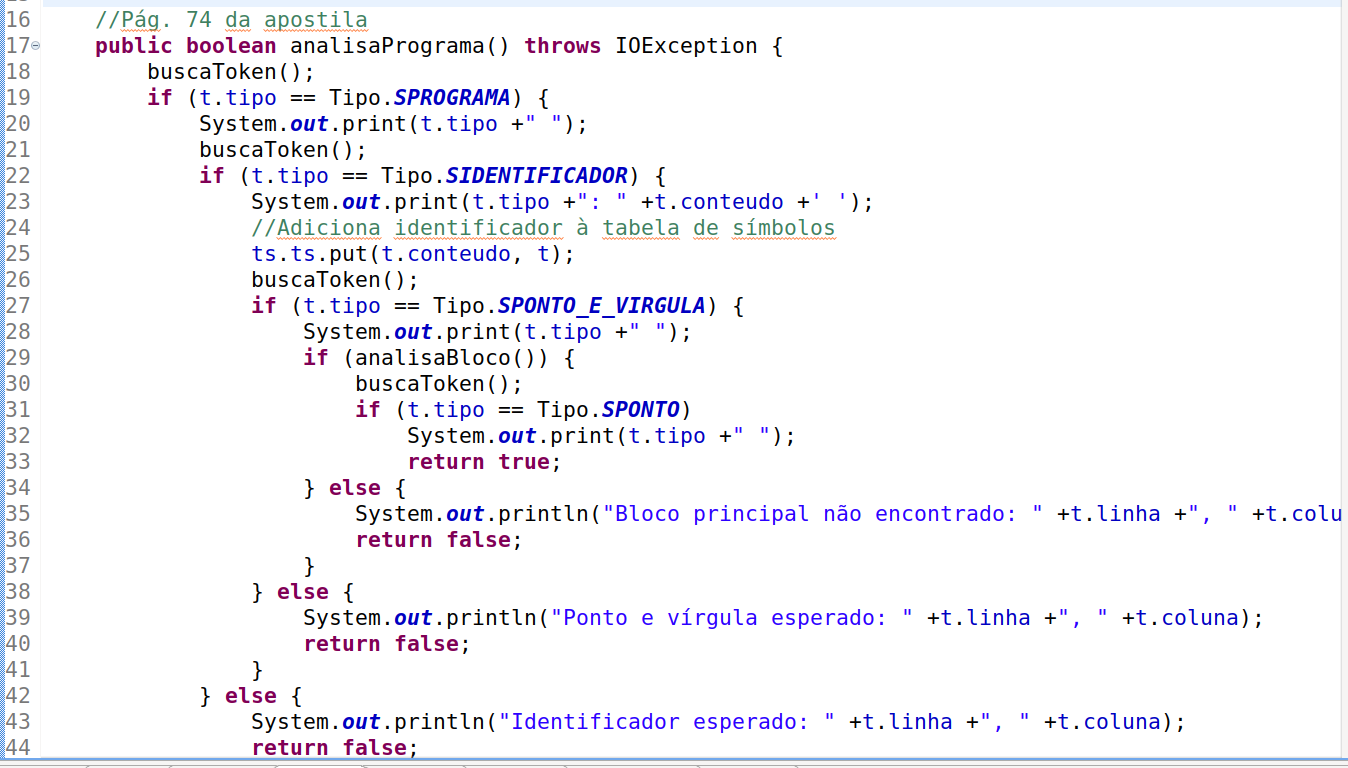
Token t;

1. O construtor de Parser deve criar os objetos ts e lexer.
2. Definir o método abstrato parse() que deve ser implementado pela(s) classe(s) concreta(s).
3. Na classe Parser implementar o método buscaToken().

t = lexer.buscaToken();

1. Criar classe PPR (Parser Preditivo Recursivo) que estende a classe Parser.
2. O construtor de PPR deve chamar o construtor da classe pai.
3. Implementar o método erro() em Parser, de maneira que receba uma string como parâmetro e escreva a string, a linha e a coluna no console, e retorne false.
4. O método parse() deve chamar método que inicia o analisador sintático (analisaPrograma)
5. Implementar os métodos que implementam o analisador sintático em PPR, conforme a página 74, de forma que reconheça o código de exemplo.
   1. analisaPrograma
   2. analisaBloco
   3. analisaEtapaDeclaracaoDeVariaveis





1. Tratar erros adequadamente: imprimir uma mensagem relevante e informar onde o erro ocorreu (linha e coluna).