

- (1) O objetivo do trabalho é implementar um compilador para a Linguagem Karloff, usando tradução dirigida por sintaxe, na ferramenta Javacc. Para realizar esta tarefa, o aluno deve adicionar ações semânticas às regras sintáticas já implementadas em Javacc, para que seja gerada uma árvore sintática do programa Karloff sendo compilado. Em seguida, o programa deve percorrer a árvore sintática gerando código (**em qualquer linguagem**), semanticamente equivalente ao código Karloff original. As dicas de como implementar esse compilador estão na **vídeo aula: 12 - Tradução Dirigida Por Sintaxe usando o Javacc**.

As classes a serem usadas para a árvore sintática estão disponíveis para download no e-aula (arquivo **ArvoreSintatica.zip**). A explicação sobre as classes está no arquivo **Árvore Sintática.pdf**. **É obrigatório o uso dessas classes na construção da árvore sintática.**

O main do javacc, deve ficar mais ou menos assim:

```
// importar as classes da árvore sintática:

import ast.*;

public class Karloff {

    public static void main(String args[]) throws Exception{
        // abrir o arquivo passado por linha
        // de comando contendo o código em Karloff:

        FileInputStream fs = new FileInputStream(new File(args[0]));

        // Instanciar o parser da linguagem Karloff passando
        // como argumento o arquivo contendo o código
        //Karloff a ser processado:

        Karloff parser = new Karloff(fs);

        // Chamar a primeira regra do parser que irá
        // analisar o código e devolver a árvore sintática

        Prog arvore =parser.Karloff();

        // passar a árvore para o gerador de código
        // que deve gerar um arquivo com com o código
        // alvo na linguagem escolhida:

        geraCodigo(arvore, args[0])

    }

    public static void geraCodigo(Prog prog, String arquivo){??????}

}
```

Além do compilador escrito em Javacc (**Karloff.jj**) o aluno deve desenvolver **dois exemplos de programas** que usem uma grande parte das construções sintáticas disponíveis na linguagem.