

## **Objetivo**

O objetivo do Trabalho é a implementação de um interpretador para uma calculadora conforme a descrição abaixo.

## **Realização**

- O trabalho poderá ser realizado individualmente ou em grupos de até 3 alunos
- Dúvidas devem ser inicialmente discutidas no fórum de discussão disponível no sistema *moodle*, e, se necessário, trazidas para discussão em aula
- Apresentações: dias 28 e 30 de junho. Todos os componentes do grupo devem participar da apresentação
- Trabalhos que não compilarem/executarem serão desconsiderados
- Os trabalhos que configurarem algum tipo de fraude (cópia) serão anulados e atribuída a nota zero a todos os alunos envolvidos

## **Entrega Parcial - Análise léxica e sintática.**

**Data de entrega: 06 de junho**

**Peso: 30% da nota do projeto**

Implementação de analisadores léxico/sintático (utilizando os geradores Flex/Byacc) para a calculadora. Definição de um sistema de tratamento de erros para entradas inválidas, de forma a emitir mensagem adequada e retomar o processamento. Especificação da estrutura da Tabela de Símbolos e da Árvore Sintática Abstrata (ASA). Versão inicial da construção da ASA e do interpretador.

### **Entrega (conteúdo do arquivo, em formato ZIP, postado no moodle)**

- Especificações JFlex e ByaccJ implementadas e todos os arquivos fontes necessários à compilação e execução do exercício.
  - Obs. o grupo pode optar por utilizar a versão C++, ou C, dos geradores
- “Programas teste” que demonstrem o que foi implementado.
- Todos os arquivos devem estar claramente identificados com o nome, número de matrícula e e-mail de cada um dos componentes do grupo.
- Relatório, também identificado, no formato de um artigo (ver exemplo no moodle) contendo a descrição do trabalho realizado. Este relatório deverá ser cumulativo entre as duas entregas e na sua versão final deverá ter um relato completo do trabalho realizado. O relatório deve ser entregue em formato PDF junto aos demais artefatos solicitados. A falta de relatório ou formato incompatível com o solicitado podem penalizar a entrega em até um ponto na nota final.
- teste” que demonstrem o que foi implementado.
- Todos os arquivos devem estar claramente identificados com o nome, número de matrícula e e-mail de cada um dos componentes do grupo.
- Relatório, também identificado, no formato de um artigo (ver exemplo no moodle) contendo a descrição do trabalho realizado. Este relatório deverá ser cumulativo entre as diferentes entregas e na sua versão final deverá ter um relato completo do trabalho realizado. O relatório deve ser entregue em formato PDF junto aos demais artefatos solicitados. A falta de relatório ou formato incompatível com o solicitado podem penalizar a entrega em até um ponto na nota final.

## **Entrega final**

**Data de entrega: 20 de junho**

**Peso: 70% da nota do projeto**

Estender a parte inicial do trabalho para geração da TS e ASA para todas as construções solicitadas. Complementar a implementação do interpretador.

**Entregar:** mesmos itens listados na entrega parcial, com complementação do artigo.

### *Especificação da calculadora*

A calculadora a ser implementada é a calculadora “bc”, disponível em sistemas Linux, com várias simplificações para tornar o trabalho viável no prazo proposto. Mais informações em <https://www.gnu.org/software/bc/manual/bc.html> ou <http://sciencesoft.at/bc/>, entre outros links.

Para o projeto de compiladores, as seguintes regras devem ser aplicadas:

- Identificadores, usados para nomes de funções, argumentos e variáveis seguem a regra de formação da linguagem Java
- A calculadora trabalha apenas com os tipos numérico (double) e lógico (boolean), o tipo é inferido durante o processamento. Os valores numéricos são sempre apresentados com três casas decimais;
- Para nossa versão da calculadora não existem variáveis globais, logo argumentos e variáveis utilizadas no escopo de uma função são sempre locais a esta. Funções recursivas são permitidas o que exigirá um certo cuidado para utilização destes elementos;
- A calculadora contém três modos básicos de operação, a sintaxe específica de cada um pode ser encontrada na documentação, alterações são permitidas a critério do grupo, desde que explicadas no relatório entregue:
  - **Imediato:** A expressão é compilada (gera uma ASA) e executada imediatamente. Exemplo: “2^3+5”. A expressão é avaliada e seu resultado mostrado na tela.
  - **Atribuição:** a expressão é compilada, executada, e armazenada associada ao identificador. Exemplo: f = 2^b+5
  - **Declaração de função.** A expressão é compilada e armazenada em uma tabela de funções para uso posterior.  
Exemplos:
    - `define double (n) { return (2*n); }`
    - `define fat(n) { if (n==1) return 1; return n*fat(n-1) }`
    - `define f(n) { res = 1; for (i=1;i<=n;i++) res = res*i; return res; }`
- **Controle:**
  - `#help` – mostra um pequeno auxílio ao uso da calculadora (conteúdo a critério do grupo)
  - `#show “ident”` – apresenta os dados armazenados na tabela de valores/funções associadas ao identificador “ident”, no escopo corrente
  - `#show_all` – apresenta todos os dados armazenados nas tabelas de valores/funções
- Operadores válidos, valendo as regras de precedência e associatividade da linguagem Java:
  - Aritméticos: +, -, \*, /, ^ (potência, maior precedência e associativo à direita). Operandos devem ser numéricos e o resultado é numérico
  - Relacionais: >, >=, <, <=, !=. Operandos numéricos e o resultado lógico.
  - Lógicos: &&, ||, !. Operandos lógicos, resultado lógico
  - Atribuição: =, +=, \*=. Identificador do mesmo tipo do resultado da expressão.
- Comandos
  - Seleção (if)
  - Repetição (while e for)
  - Impressão (print)
- Definição de funções.
  - **Atenção!** A declaração e uso de funções é o grande diferencial da calculadora e então valerá 25% da nota final do trabalho!

### *Especificação da calculadora*

A calculadora a ser implementada é a calculadora “bc”, disponível em sistemas Linux, com várias simplificações para tornar o trabalho viável no prazo proposto. Mais informações em <https://www.gnu.org/software/bc/manual/bc.html> ou <http://sciencesoft.at/bc/>, entre outros links.

Para o projeto de compiladores, as seguintes regras devem ser aplicadas:

- Identificadores, usados para nomes de funções, argumentos e variáveis seguem a regra de formação da linguagem Java;
- A calculadora trabalha apenas com os tipos numérico (double) e lógico (boolean), o tipo é inferido durante o processamento. Os valores numéricos são sempre apresentados com três casas decimais;
- Embora possam ser declaradas variáveis “globais” (a partir de um comando de atribuição), os argumentos e variáveis utilizadas em uma função são sempre locais a esta. Funções recursivas são permitidas o que exigirá um certo cuidado para utilização destes elementos;
- A calculadora contém três modos básicos de operação, a sintaxe específica de cada um pode ser encontrada na documentação, alterações são permitidas a critério do grupo, desde que explicadas no relatório entregue:
  - Imediato: A expressão é compilada (gera uma ASA) e executada imediatamente. Exemplo: “2^3+5”. A expressão é avaliada e seu resultado mostrado na tela.
  - Atribuição: a expressão é compilada, executada, e armazenada associada ao identificador. Exemplo: f = 2^b+5
  - Declaração de função. A expressão é compilada e armazenada em uma tabela de funções para uso posterior. Exemplo: define doisN (n) { return (2\*n); }
  - Controle:
    - #help – mostra um pequeno auxílio ao uso da calculadora (conteúdo a critério do grupo)
    - #show “ident” – apresenta os dados armazenados na tabela de valores/funções associadas ao identificador “ident”
    - #show – apresenta todos os dados armazenados nas tabelas de valores/funções
- Operadores válidos, valendo as regras de precedência e associatividade da linguagem Java:
  - Aritméticos: +, -, \*, /, ^ (potência, maior precedência e associativo à direita). Operandos devem ser numéricos e o resultado é numérico
  - Relacionais: >, >=, <, <=, !=. Operandos numéricos e o resultado lógico.
  - Lógicos: &&, ||, !. Operandos lógicos, resultado lógico
  - Atribuição: =, +=, \*=. Identificador do mesmo tipo do resultado da expressão.
  - Condicional: “?:”. Expressão de teste lógica, outras duas do mesmo tipo.
  - Sequencia: “,”. Expressões são avaliadas sequencialmente e o resultado da lista é o resultado da última expressão avaliada.
- Comandos
  - Seleção (if)
  - Repetição (while e for)
  - Impressão (print)
- Definição de funções.
  - **Atenção!** A declaração e uso de funções é o grande diferencial da calculadora e então valerá 25% da nota final do trabalho!