

 <p>UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA <i>Departamento de Engenharia Informática</i></p>	<p>Projecto de Compiladores</p> <p>2010/11- 2º semestre</p> <p>Licenciatura em Engenharia Informática</p> <p>Data de Entrega: 6 de Junho</p>
<p><u>Nota Importante:</u> A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior e futuro profissional licenciado. Qualquer tentativa de fraude leva à anulação da componente prática tanto do facilitador como do prevaricador.</p>	

Compiladores

Compilador para a linguagem PJAVA

Tema

Neste projecto pretende-se o desenvolvimento de um compilador da linguagem PJAVA, que consiste num muito pequeno subconjunto de funcionalidades inspiradas na linguagem JAVA. Considera-se apenas a vertente procedimental do JAVA: os programas da linguagem PJAVA são constituídos por uma única classe (a classe principal), contendo necessariamente um método main, e podendo conter outros métodos e atributos, todos eles estáticos e (possivelmente) públicos. Não se pretende explorar a vertente orientada a objectos.

Os programas em PJAVA devem poder incluir dados e realizar operações sobre esses dados. Entre as funcionalidades a implementar encontram-se:

- tipos de dados elementares (*int, float, char*, etc.);
- declaração de variáveis;
- definição e invocação de funções/métodos;
- expressões envolvendo
 - operações aritméticas (+, -, *, etc.);
 - operações relacionais (=, <, >, <=, etc.);
 - operações lógicas (!, &&, //, etc.);
- instruções de:
 - atribuição (=, +=, -=, etc.)
 - controlo (de selecção e repetição: *if, for, while, return*)
 - saída (*System.out.println*)

O significado de um programa em PJAVA será o mesmo que o seu significado em JAVA. Por exemplo, o seguinte programa calcularia o máximo divisor comum de dois inteiros, e apresentaria esse valor no écran:

```

class gcd_rec {
    public static void main(String[] args) {
        int a, b;
        a = 315;
        b = 1932;
        System.out.println(gcd(a,b));
    }
    public static int gcd(int a, int b) {
        if (a == b || b == 0)
            return a;
        else if (a == 0)
            return b;
        else if (a > b)
            return gcd(a % b, b);
        else /* b < a */
            return gcd(a, b % a);
    }
}

```

Pormenores sobre a sintaxe podem ser encontrados em <http://www.oracle.com/technetwork/java/index.html>.

Deverá ter em conta o seguinte:

- Não esqueça que, com detecções de erros, as mensagens enviadas ao programador deverão ser o mais esclarecedoras possível.
- O código final do seu compilador deverá estar numa versão de C reduzido (ver restrições abaixo).
- O seu compilador deverá receber um ficheiro fonte (de extensão *.pja*) e produzir um ficheiro “.c” correspondente. A visualização dos resultados poderá ser feita utilizando o gcc.

No seu código final C, não poderá em caso algum violar as seguintes regras:

- Todo o programa estará escrito dentro de uma única função *main()*, excepto, eventualmente instruções *include* de que necessite.
- Não poderá existir qualquer bloco de código (rodeado por “{ }”) para além do próprio da função *main*.
- Não poderá fazer chamadas de funções externas, a não ser *printf* e *malloc*.
- Não poderá utilizar qualquer instrução de ciclo (*for*, *while* ou *do*) ou *switch*.
- Para fazer redireccionamento de fluxo, deverá utilizar a instrução *goto*<label>
- A instrução *if* não poderá contemplar a parte “else”. Ou seja, será sempre *if*<condição> <acção>.

Extras

Para este projecto, está reservada uma componente importante de extras. Considera-se um *extra* qualquer extensão à linguagem proposta (respeitando a condição de o resultado continuar a ser válido em Java) ou ao compilador que o valorize. São alguns exemplos:

- Outros tipos de dados
- Acesso ao argumentos passados na linha de comandos

- Instruções de leitura
- Leitura e escrita com ficheiros
- Etc.

Note que estes extras são só exemplos e não têm todos necessariamente o mesmo grau de dificuldade.

Fases

O projecto terá metas intermédias em que haverá defesas. Deverão ser consideradas também como guias ou linhas reguladoras para evitar o “descarrilamento” do projecto de desenvolvimento do compilador. As metas serão:

1. Ficheiro lex, gramática da linguagem, especificação da sintaxe abstracta, construção da árvore respectiva e análise sintática completa – Dia 18 de Abril
2. Análise semântica (tabelas de símbolos, detecção de erros de semântica) – Dia 16 de Maio
3. Código Final e Relatório- Dia 6 de Junho

As entregas serão feitas no moodle. A entrega final do projecto deverá ser até ao dia **6 de Junho**. Cada grupo deverá nesta meta entregar **na secretaria até às 17h ou no cacifo do professor até às 23h59** os seguintes documentos:

- Todo o software criado (em CD)
- O relatório, que deverá conter a especificação da gramática, da sintaxe abstracta, e descrição geral do projecto.

Defesa e grupos

Durante as semanas seguintes às entregas, proceder-se-á à defesa do trabalho, sendo as das duas primeiras metas realizadas nas TPs. É fortemente encorajada a formação de grupos de dois alunos e que sejam ambos da mesma turma prática. No entanto o princípio do equilíbrio é fundamental. Alunos que se sintam mais confortáveis com trabalho individual poderão fazê-lo.

Avaliação

Neste trabalho, pretende-se que o aluno obtenha conhecimentos práticos com alguma profundidade sobre a construção de compiladores. Pretende-se também estimular a criatividade e autonomia, pelo que extensões à linguagem que confirmem um aumento de potencialidades serão favorecidas.

Bibliografia

- | |
|---|
| <p>. <i>Processadores de Linguagens</i>. Rui Gustavo Crespo. IST Press. 1998</p> <p>. <i>Modern compiler implementation in C</i>. A. Appel. Cambridge Press. 1998.</p> <p>. <i>A Compact Guide to Lex & Yacc</i>. T. Niemann. http://epaperpress.com/lexandyacc/epaperpress.</p> <p>. Manual do yacc em Unix (comando “man yacc” na shell)</p> <p>. <i>Lex & Yacc</i>. John R. Levine, Tony Mason and Doug Brown. O’Reilly. 2004</p> <p>. Manuais da linguagem C (por exemplo, <i>Linguagem C</i>, Luís Damas, FCA. 1999)</p> |
|---|