UNIVERSIDADE FEDERAL FRONTEIRA SUL
CIÊNCIA DA COMPUTAÇÃO – 3º Período
ALUNOS: Emerson Martins, Leonardo Vargas
DATA: 15/05/15

TRABALHO DE PROGRAMAÇÃO I :

Implementar um programa em java que interprete uma linguagem de programação criada pelos alunos. O interpretador deve ser capaz de analisar e reagir corretamente as seguintes situações no arquivo fonte que ele esteja interpretando:

- ✪ Declaração de variáveis;
- ✪ Atribuição de valor a variável;
- ✪ Expressões com no minimo dois operandos;
- ✪ Operações de adição, subtração, divisão e multiplicação;
- ✪ Laço;
- ✪ Comando de saída (ex: Mostrar algo na tela);
- ✪ Controlador de fluxo (ex: If);
- ✪ Aninhamento de comandos (ex: ifs dentro de ifs, laços dentro de laços).

Sobre a linguagem:

A linguagem chama-se **Toon World** e é definida e especificada pelos seguintes comandos:

Arquivo Especifico:

Para a utilização do interpretador é necessário um arquivo na forma “Arquivo.toon”.

Inicio e fim do programa:

Não é necessário ter nenhum comando especifico de começo ou fim do programa, somente o código.

Uso do terminador:

O único caractere obrigatório nos comandos é o terminador. Para funções diferentes de laços e condições (será explicado posteriormente) o terminador é o simbolo “?”.

★ Exemplo de programa:

“código” ?
“código” ?

Declaração de variáveis:

A linguagem suporta os tipos Int, Double e String. No momento da declaração é permitida a atribuição de valores usando o simbolo “<)” depois do nome da variável. É possível atribuir um valor usando uma operação aritmética. No caso de Strings são usadas “”(aspas) para a atribuição. Para declarar uma variável sem definir seu valor é necessário apenas usar o terminador após o nome da mesma.

★ Exemplo de programa:

```
Int A <) 10?
Int B <) 12*2?
Double C?
String J <) “sou uma string”?
```

Principais operadores:

Os principais operadores são + (soma), - (subtração), / (divisão), * (multiplicação) e % (Mod). Não é possível fazer mais de uma operação por vez.

Existem ainda outras operações como:

<)	Atribuição, atribui à uma variável o valor desejado.
<<	Menor, compara números ou variáveis e retorna verdadeiro ou falso.
>>	Maior, compara números ou variáveis e retorna verdadeiro ou falso.
==	Igual, compara números ou variáveis e retorna verdadeiro ou falso.
|=	Diferente, compara números ou variáveis e retorna verdadeiro ou falso.
>|	Maior ou Igual, compara números ou variáveis e retorna verdadeiro ou falso.
<|	Menor ou Igual, compara números ou variáveis e retorna verdadeiro ou falso.
?	Terminador das linhas que não definem inicio ou fim de função If, four e while.
{	Inicio de expressão,utiliza a mesma da ideia de “(” em outras linguagens.
}	Fim de expressão,utiliza a mesma da ideia de “)” em outras linguagens.
[	Abre escopo, define o inicio de uma função if, four ou while.
]	Fecha escopo, define o fim de uma função if, four ou while.

Função WENN:

Tem a mesma ideia de IF em outras linguagens. Funciona da seguinte maneira: “WENN{cond} [” dentro das chavetas vai a condição a ser analisada. Retorna verdadeiro ou falso. Há também a possibilidade de usar um “] SONST [” (Mesma ideia de else). A cada fechamento de escopo é necessário colocar um w(wenn) ou s(sonst) ao lado da chaveta.

★ Exemplo de programas:

```
WENN{A<<B}[              se A é menor que B
        “código”
]W

WENN{A>>B}[              se A é maior que B
        “código”
]W

WENN{A>|B}[              se A é maior ou igual a B
        “código”
]W SONST [
        “código”
]S

WENN{A<|B}[              se A é menor ou igual a B
        “código”
]W SONST [
        “código”
]S
```

Função VOOR

Tem a mesma ideia de FOR em outras linguagens. Funciona da seguinte forma: “VOOR{}[” Dentro das chavetas é necessário ter 3 “comandos”. O 1º de atribuição a variável, o 2º de condição de execução (do mesmo modo da função WENN) e o 3º também de atribuição de variável (var++ ou var--).

Para separar esses comandos é necessário o uso do terminador.

★ Exemplo de programa:

```
Int A?
VOOR{a<)0? A<<1? a++?}[
        “código”
]V
```

Função GIRAGIRA

Tem a mesma ideia de WHILE em outras linguagens. Funciona da seguinte forma: “GIRAGIRA{cond}[” Dentro das chavetas estará a condição a ser analisada (do mesmo modo da função WENN). Retorna verdadeiro ou falso.

★ Exemplo de programa:

```
Int A <) 0?
GIRAGIRA{A<<10}[
        “código”
        “código”
]G
```

Função PRINT

Essa função tem a mesma ideia do comando System.out.println() em Java. Funciona de 2 maneiras:
“PRINT{}?” Imprime o que esta dentro das chavetas sem quebrar linha.
“PRINTLN{}?” Imprime o que esta dentro das chavetas quebrando a próxima linha.

Dentro das chavetas também existem 2 maneiras de funcionamento:
“PRINT{“sou um teste”}?” Imprime o que estiver entre as “” exatamente como estiver escrito.
“PRINT{“Valor de A: ” + A}?” Imprime o texto seguido do valor de A. Pode ser feito quantas vezes forem necessárias.

★ Exemplo de programa:

```
Int A <) 5?

PRINTLN{“oi ” + A + “você”}?        Imprime: oi 5voce (e quebra a próxima linha)
PRINT{A}?                           Imprime: 5
```

Função DIZPRAMIM

Essa função tem a mesma ideia de scanf em C. Ela pode ser usada para as variáveis de tipo Int e Double. Funciona da seguinte maneira:
“DIZPRAMIM(#NomeVariavel)?” .

★ Exemplo de programa:

```
Int A?

DIZPRAMIM(#A)?          valor sera atribuído a variável A.
```

Função DARKSIDE

Essa função imprime as informações de todas as variáveis (nome,valor,posição de memoria). Funciona de duas maneiras:
Se executada sem parâmetros mostra TODAS as variáveis.
Se executada com parâmetros mostra APENAS os próprios parâmetros.

★ Exemplo de programa:

```
Int A <) 10?
Int B <) 20?
Int C <) 30?

DARKSIDE{ }?          mostra as informações de A,B e C.
DARKSIDE{A,B}?        mostra as informações de A e B.
```

Linguagem Flexível

É possível utilizar sintaxe flexível no código desde que o comando inteiro fique na mesma linha. Isso torna a identação possível fazendo com que a compreensão do programa se torne mais fácil.

★ Exemplo de programa:

```
Int A <) 0?
Int B <) 5?
GIRAGIRA{A<<B}[
        A++?
        WENN{A == 2}[
                PRINT{“Entrou no WENN”}?
                A--?
        ]W
]G
```