



UNIVERSIDADE FEDERAL DA BAHIA

Documentação QPortugol – Moodle

SALVADOR

05/2017

PIBITI - UFBA

Trabalho referente ao
projeto de iniciação científica da
Universidade Federal da Bahia
com intuito de desenvolver um
interpretador da pseudo-
linguagem Portugol para o
ambiente moodle

SALVADOR
05/2017

Sumário

1. Introdução
2. Características gerais
 - 2.1 Tipos de Dados
 - 2.2 Estruturas de Controle
3. Conhecendo a linguagem
 - 3.1 Olá mundo
 - 3.2 Variáveis
 - 3.3 Estruturas Condicionais
 - 3.4 Estruturas de Repetição
 - 3.4.1 Estrutura Enquanto
 - 3.4.2 Estrutura Para
 - 3.5 Funções Internas “leia” e “imprima”
4. Referências

1.Introdução

O QPortugol é um dialeto da pseudo-linguagem Portugol, que é amplamente utilizada para desenvolver algoritmos em português de forma simples e prática com o intuito de facilitar o ensino de algoritmos.

A proposta desse projeto foi disponibilizar uma implementação da linguagem Portugol fornecendo ferramentas que oferecem recursos de edição, compilação, execução e depuração de programas escritos nessa linguagem, de forma a favorecer estudantes que dão os primeiros passos no aprendizado de desenvolvimento de softwares, bem como professores que ensinam disciplinas relacionadas a computação. Portanto, seu foco é primariamente didático.

2.Características Gerais

A linguagem em si não difere fundamentalmente ou apresenta novidades em relação ao uso popular de português. Semelhante às linguagens como Pascal e C, é uma linguagem imperativa, com comandos de controle de fluxo, manipulação de dados e recursos de entrada e saída básicos. A linguagem é case sensitive, o que significa que, por exemplo, uma função chamada “leia” é diferente de uma função chamada “Leia”.

2.1 Tipos de dados

Tipos primitivos como inteiro e literal são suportados. Literais são tipos integrais, e não, tipos construídos pelo usuário. A tabela 2.1 apresenta os tipos com respectivos exemplos de dados:

Tipos primitivos	
Tipo	Exemplos
inteiro	19
real	2.5
caractere	'a'
literal	“uma frase”
lógico	verdadeiro

Tabela 2.1: Tipos de dados primitivos

2.2 Estruturas de controle

O conjunto de estruturas de controle são os mais primitivos. Uma estrutura condicional (se/senão) e duas estruturas de repetição (enquanto/para) são suportadas. Embora a estrutura “para” seja uma especialização de um laço simples como o “enquanto”, ela foi implementada, visto que uma variedade de livros e muitos professores os discutem.

Estruturas de controle	
Estrutura	Tipo
se/então/senão/	condicional
enquanto	repetição
para	repetição

Tabela 2.2: Estruturas de repetição

3. Conhecendo a linguagem

A proposta desse capítulo é mostrar os elementos da linguagem QPortugol usando programas como exemplos. Não é um capítulo sobre como programar, nem tem como intuito ensinar algoritmos. Portanto, é esperado que o leitor seja familiarizado com programação.

3.1 Olá Mundo

Nessa seção criaremos o nosso primeiro programa. O mesmo terá como propósito exibir na tela a frase “Olá Mundo”.

Exemplo 1: Olá Mundo

```
algoritmo OlaMundo;  
inicio  
    imprima ("Olá Mundo!");  
fim
```

3.2 Variáveis

Nessa seção será mostrado como a declaração de variáveis deve ser feita. Para tal, utilizaremos um bloco específico que deve aparecer logo após a declaração do algoritmo. O bloco é iniciado pela palavra-chave “variaveis” e termina com a palavra chave “fimvariaveis”. Pelo menos uma variável deve ser declarada dentro do bloco (embora o bloco em si seja opcional) e apenas um bloco em escopo global deve ser declarado.

Exemplo 2: Declaração de Variáveis

```
algoritmo teste_variaveis;  
variaveis  
    x:inteiro;  
    nome:literal;  
fimvariaveis  
inicio  
fim
```

3.3 Estruturas condicionais

Nesse programa, é possível utilizar as estruturas condicionais se/ entao/ senao. Essas são estruturas que impõem uma condição ao algoritmo utilizando expressões, que são avaliadas para que uma decisão seja tomada (repetir execução, selecionar bloco de instruções, etc). Qualquer expressão pode ser avaliada como expressão lógica. Expressões numéricas de valor “0” são avaliadas como falso. Demais valores numéricos são avaliados como verdadeiro. Valores literais nulos, da mesma forma, são avaliados como falso, e demais textos, como verdadeiro.

Exemplo 3: Estruturas condicionais

```
algoritmo teste_condicao;
variaveis
    idade : inteiro;
    nome : literal;
fimvariaveis
inicio
    imprima("Digite seu nome:");
    nome := leia();
    imprima(nome + ", digite sua idade:");
    idade := leia();
    se idade >= 18 entao
        se idade < 60 entao
            imprima("adulto!");
        senao
            imprima("ancião" + "!");
        fimse
    senao
        imprima("menor" + "!");
    fimse
fim
```

3.4 Estruturas de repetição

Nesse programa é possível utilizar as estruturas de repetição para/ enquanto. Essas são estruturas que impõem a repetição de um bloco de instruções dentro do algoritmo.

3.4.1 Estrutura “enquanto”

A estrutura “enquanto” é a mais básica e comum das estruturas de repetição. Nessa estrutura um conjunto de

instruções é executado repetidamente enquanto o valor lógico de uma dada expressão for avaliado como “verdadeiro”.

Exemplo 4: Estrutura enquanto

```
algoritmo fatorial;
variaveis
    res : inteiro;
    fat : inteiro;
    x : inteiro;
fimvariaveis
inicio
    imprima("Digite um número:");
    x := leia();
    fat := x;
    res := 1;
    enquanto x <> 0 faca
        res := res * x;
        x := x - 1;
    fimenquanto
    imprima("fatorial de " + fat + " é igual a " + res);
fim
```

3.4.2 Estrutura “para”

A estrutura “para” é uma especialização da estrutura “enquanto”. Sua sintáxe é simples e determina um número finito de repetições as quais um bloco de instruções deve executar.

Exemplo 5: Estrutura para

```
algoritmo fatorial;
variaveis
    res : inteiro;
    fat : inteiro;
    x : inteiro;
fimvariaveis
inicio
    imprima("Digite um número:");
    fat := leia();
    res := 1;
    para x de fat ate 1 passo -1 faca
        res := res * x;
    fimpara
    imprima("fatorial de " + fat + " é igual a " + res);
fim
```


3.5 Funções Internas

O QPortugol oferece duas funções internas: “leia” e “imprima”, que permitem uso básico de entrada e saída, respectivamente.

A função “leia” não recebe argumentos e retorna o valor lido da entrada padrão (“STDIN”), o que significa, geralmente, ler os dados que o usuário digitar no teclado, seguido do caractere “nova linha” (em geral, associado a tecla “Enter” no teclado). O tipo de dado retornado por “leia” é implicitamente convertido para o tipo primitivo exigido no contexto em que ela é usada.

A função “imprima” recebe um número variável de argumentos de qualquer tipo primitivo, sendo que pelo menos um argumento deve ser passado. Os valores passados como argumentos são convertidos para texto, concatenados por um caractere “+” na ordem definida e enviados para “STDOUT” (em geral, associado ao display ou monitor). Não há retorno de valor para esta função.

Exemplo 6: Funções leia e imprima

```
algoritmo leia_imprima;
variaveis
    nome : literal;
fimvariaveis
inicio
    imprima("Olá, qual o seu nome?");
    nome := leia();
    imprima("Seja bem vindo ao portugol, " + nome);
fim
```

5. Referências

SILVA, Thiago. **G-Portugol: Manual da v1.0.** Disponível em <<http://www.slideshare.net/GabrielFaustino/manual-de-portugol>>. Acesso em 21/05/2017.