FUNDAÇÃO EDUCACIONAL INACIANA “PE. SABÓIA DE MEDEIROS” (FEI)

CURSO CIÊNCIA DA COMPUTAÇÃO

PROJETO DE COMPILADORES

ENZO PACHECO PORFIRIO

FELIPE ORLANDO LANZARA

PEDRO HENRIQUE LEGA KRAMER COSTA

JOÃO VITOR GOVERNATORE

SÃO BERNARDO DO CAMPO – SP

2024

ENZO PACHECO PORFIRIO

1. **EXPRESSÕES REGULARES**

**EOI** (End of Instruction) → ;

**OB** (Open Bracket) → {

**CB** (Close Bracket) → }

**AND** (Logical Operator) → &&

**OR** (Logical Operator) → **||**

**NOT** (Logical Operator) → **!**

**LT** (Less Than) → **<**

**GT** (Greater Than) → **>**

**LE** (Less Equal) → **<=**

**GT** (Greater Than) → **>=**

**EQUAL** (Conditional Operator) → **==**

**ASSIGN** (Assign Operator) → **=**

**PLUS (Addition Operator) → +**

**SUB (Subtraction Operator) → -**

**MUL (Multiplication Operator) → \***

**DIV (Division Operator) → /**

**OP (Open Parenthesis) → (**

**CP (Close Parenthesis) → )**

**FLOAT** (Values) → [0-9]+.[0-9]+

**INT** (Values) → [0-9]+

**ID** (Symbols) → [a-zA-Z]+[a-zA-Z0-9\_]\*

TXT (Char Sequence) → “[^”]”

**RESERVED** (Reserved Words) → inteiro | decimal | texto | estado | real | barça | para | enquanto | se | senao | saida | entrada

1. **GRAMÁTICA**

**BLOCO →** IFELSE BLOCO | ATR BLOCO | DECL BLOCO | FOR BLOCO | WHILE BLOCO | PRINT BLOCO | ε

**DECL →** INT | FLOAT | STRING | BOOLEAN

**PRINT →** saida ( id | txt | EXP ) ;

**INT →** inteiro id INT’

**INT’→** ; | = INT’’

**INT’’ →** EXP ; | entrada( );

**FLOAT →** decimal id FLOAT’

**FLOAT’ →** ; | = FLOAT’’

**FLOAT’’→** EXP ; | entrada();

**STRING →** texto id STRING’

**STRING’ →** ; | = STRING’’

**STRING’’ →** txt ; | entrada();

**BOOLEAN →** estado id BOOLEAN’

**BOOLEAN’ →** ; | = BOOLEAN’’

**BOOLEAN’’ →** real ; | barça ;

**ATR →** id X

**X → =** Y

**Y →** EXP ; | txt ; | bool ;

**EXPI →** TI RI

**RI →** + TI RI | - TI RI | ε

**TI →** FI SI

**SI →** \* TI RI | / TI RI | ε

**FI →** ( EXPI ) | id | int

**EXPF →** TF RF

**RF →** + TF RF | - TF RF | ε

**TF →** FF SF

**SF →** \* TF RF | / TF RF | ε

**FF →** ( EXPF ) | id | NUM

**IFELSE →** se ( COND ) { BLOCO } senao { BLOCO }

**COND →** id OP COND’

**COND’ →** id | NUM

**OP →** < | > | <= | >= | == | !=

**NUM →** float | int

**NUMI →** int

**FOR →** para ( ATR\_FOR ; COND ; INC ) { BLOCO }

**ATR\_FOR →** inteiro id = (num | id)

**INC →** id = id OP\_MAT num

**OP\_MAT →** + | - | / | \*

**WHILE →** enquanto ( condição ) { BLOCO }

1. **INSTRUÇÕES E EXECUÇÃO**

Escreva seu código no arquivo chamado code, localizado no diretório principal do projeto. Esse arquivo será utilizado pelo script init.bash como entrada para o processo de tradução.

Este projeto utiliza um script de inicialização em Bash (init.bash) para compilar e traduzir o código-fonte para C++ ou Java, conforme a opção escolhida. Siga os passos abaixo para executar o projeto.

Certifique-se de que você possui oAmbiente Bash instalado (Linux, macOS, ou Git Bash no Windows).

Abra o terminal bash no diretório raíz do projeto e siga os exemplos abaixo:

Exemplo de execução para tradução em Java:

* ./init.bash -j

Exemplo de execução para tradução em C++:

* ./init.bash -c

**Nota:** As flags -c e -j são exclusivas, então escolha uma delas conforme a linguagem de destino desejada.

Executando os comandos acima seu código será compilado e executado automaticamente no terminal aberto.

1. **CARACTERÍSTICAS DA LINGUAGEM**

Essa linguagem é fortemente tipada, o que significa que cada variável deve ser declarada com um tipo específico antes do uso. Os tipos de dados suportados são numéricos (inteiros e decimais), lógicos e textuais. A sintaxe e semântica são inspiradas em linguagens tradicionais como C e Java, com algumas simplificações.

### **Tipo de Dados**

* **inteiro**: Representa números inteiros, similar ao tipo int em linguagens como C.
* **decimal**: Representa números com ponto flutuante, similar ao tipo float.
* **texto**: Usado para armazenar sequências de caracteres, similar ao String.
* **estado**: Um tipo booleano para valores lógicos, permitindo representar valores como verdadeiro e falso (substituídos aqui por real e barça).

### **Palavras Reservadas**

A linguagem define palavras reservadas para funções e operadores específicos. Essas palavras não podem ser usados como identificadores de variáveis ou funções, pois representam instruções de controle ou tipos:

* **Tipos**: inteiro, decimal, texto, estado (booleans).
* **Controle de Fluxo**: se,senao (condicional if-else).
* **Laços**: para (loop for), enquanto (loop while).
* **Entrada e Saída**: saida (para exibir valores) e entrada (para capturar valores).

### **Operadores**

* **Matemáticos**: Inclui os operadores básicos +, -, \* e / para realizar cálculos numéricos.
* **Comparação**: Operadores como <, >, <=, >=, == permitem comparar valores e são amplamente usados em expressões condicionais.
* **Lógicos**: && (AND), || (OR) e ! (NOT) são usados para combinar condições em instruções condicionais e laços.

### **Declaração e Atribuição de Variáveis**

As variáveis devem ser declaradas com seu tipo específico, e a atribuição é realizada com o operador =.

Exemplo de declaração e atribuição:

inteiro contador = 0;

decimal preco = 19.99;

texto saudacao = "Olá, mundo!";

estado isActive = real;

Exemplo de declaração e atribuição para as palavras reservadas saida e entrada:

saida("Digite sua idade:");

inteiro idade = entrada();

**Nota:** A variável só pode armazenar valores de entrada se for declarada e inicializada no momento da captura, como no exemplo acima.

1. **EXEMPLOS E EQUIVALÊNCIAS**

### **Estrutura Condicional (se e senao)**

Na linguagem do projeto:

inteiro idade = entrada();

se (idade >= 18) {

saida("Você é maior de idade.");

} senao {

saida("Você é menor de idade.");

}

Em C++:

#include <iostream>

using namespace std;

int main() {

int idade;

cout << "Digite sua idade: ";

cin >> idade;

if (idade >= 18) {

cout << "Você é maior de idade." << endl;

} else {

cout << "Você é menor de idade." << endl;

}

return 0;

}

### 

### 

### 

### 

### 

### **Estrutura de Repetição (for)**

Na linguagem do projeto:

para (inteiro i = 0; i < 5; i = i + 1) {

saida("Iteração número: ");

saida(i);

}

Em Java:

public class IteracaoExemplo {

public static *void* main(String[] *args*) {

for (*int* i = 0; i < 5; i++) {

System.out.println("Iteração: " + i);

}

}

}