

Programando con Pseudocódigo en Español (PPE)



Compiladores

Escuela de Ciencias de la Computación

Facultad de Ciencias

Universidad Nacional de Ingeniería

Jorge Luis Carhuarica Aguilar

Tomas J. Casas Rodriguez

Características del lenguaje

PPE (Programando con PseudoCodigo en Español) es un lenguaje de programación basado en la implementación de pseudocódigo en el cual se ve el paso a paso que sigue todo pseudocódigo con sentencias sencillas con palabras claves en español. Tiene pocas similitudes con lenguajes de programación conocidos como C, C++ o Java. Con el objetivo en el cual aquellos alumnos sin ninguna experiencia previa en la programación puedan aprender conceptos básicos con este lenguaje.

Este lenguaje reconoce funciones, condicionales (SI, SINO y SINO NO) y bucles (MIENTRAS y PARA) mediante procedimientos secuenciales, por lo tanto se le puede clasificar como procedural. Además está diseñado para realizar operaciones aritméticas y lógicas con enteros, decimales y exponenciales.

Delimitadores

Una de las características que toma es que la función principal, las funciones, los condicionales y los bucles están delimitados por una palabra y la misma palabra al pero con “fin” al final. Así el bucle while está delimitado por las palabras “mientras” y “fin_mientras” o, también, la función principal (main) está delimitado por las palabras “inicioprograma” y “finprograma”. Como se nota a diferencia de lenguajes como C, C++ o Java no se utilizan llaves.

Hay que agregar que los parámetros de las funciones; y la condición dentro de los bucles y condicionales están delimitados por las llaves “{” y “}”. Seguidos de “realizar” para dar inicio a las sentencias de las funciones, condicionales o bucles.

Palabras reservadas

Las palabras reservadas son aquellas que no pueden utilizarse como identificadores por tener un significado dentro del lenguaje de programación. En PPE las palabras reservadas son las siguientes:

PALABRA RESERVADA	TOKEN	USO
INICIOPROGRAMA	INICIO_PROGRAMA	Inicializa el programa (es la funcion principal “main”)
FINPROGRAMA	FIN_PROGRAMA	Finaliza el programa
INICIOFUNCION	INICIO_FUNCION	Inicializa una función
FINFUNCION	FIN_FUNCION	Finaliza una función
DEVUELVE	RETURN	Devuelve un valor en una función
SI	INICIO_IF	Inicializa un SI
FINSI	FIN_IF	Finaliza un SI
SINO	ELSE	Realiza las sentencias para casos donde no se cumplan SI o SINO SI
SINO SI	ELIF	Define otra condición
MIENTRAS	INICIO_WHILE	Inicializa un bucle mientras
FINMIENTRAS	FIN_WHILE	Finaliza un mientras
PARA	INICIO_FOR	Inicializa un PARA

FINPARA	FIN_FOR	Finaliza un PARA
REALIZAR	DO	Realiza las sentencias dentro de las funciones, condicionales y bucles

Patrones

Las reglas que sigue el lenguaje de programación PPE son:

NO TERMINAL	REGLA
main	declaracion_funcion INICIO_PROGRAMA instr FIN_PROGRAMA
declaracion_funcion	INICIO_FUNCION ID LLIZQ parametros_decl LLDER DO instr FIN_FUNCION declaracion_funcion
	INICIO_FUNCION ID LLIZQ LLDER DO instr FIN_FUNCION declaracion_funcion
	INICIO_FUNCION ID LLIZQ parametros_decl LLDER DO instr RETURN nivel1 PUNTCOM FIN_FUNCION declaracion_funcion
	INICIO_FUNCION ID LLIZQ LLDER DO instr RETURN nivel1 PUNTCOM FIN_FUNCION declaracion_funcion
parametros_decl	ID
	ID PUNTCOM parametros_decl
instr	var PUNTCOM instr
	bloque_if instr
	bloque_while instr
	bloque_for instr
	bloque_funcion instr
bloque_funcion	ID LLIZQ parametros_bloq LLDER PUNTCOM
	ID LLIZQ LLDER PUNTCOM
	ID ASIGNAR ID LLIZQ parametros_bloq LLDER PUNTCOM
	ID ASIGNAR ID LLIZQ LLDER PUNTCOM
parametros_bloq	term
	term PUNTCOM parametros_bloq

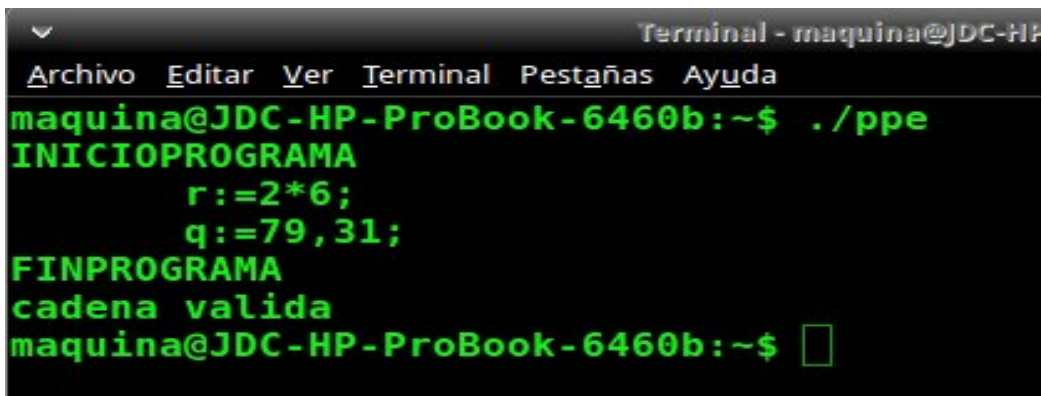
bloque_if	INICIO_IF LLIZQ nivel1 LLDER DO instr FIN_IF
	INICIO_IF LLIZQ nivel1 LLDER DO instr ELSE instr FIN_IF
	INICIO_IF LLIZQ nivel1 LLDER DO instr bloque_elif FIN_IF
	INICIO_IF LLIZQ nivel1 LLDER DO instr bloque_elif ELSE instr FIN_IF
bloque_elif	ELIF LLIZQ nivel1 LLDER DO instr
	ELIF LLIZQ nivel1 LLDER DO instr bloque_elif
bloque_while	INICIO_WHILE LLIZQ nivel1 LLDER DO instr FIN_WHILE
bloque_for	INICIO_FOR LLIZQ var PUNTCOM nivel1 PUNTCOM var LLDER DO instr FIN_FOR
var	ID ASIGNAR nivel1
nivel1	nivel2 PARIZQ nivel1 PARDER nivel2
	nivel2
nivel2	nivel2 MAS nivel3
	nivel2 MENOS nivel3
	nivel3
nivel3	nivel3 POR nivel4
	nivel3 ENTRE nivel4
	nivel3 RESIDUO nivel4
	nivel4
nivel4	nivel4 POTENCIA nivel5
	nivel5
nivel5	nivel5 MENOR nivel6
	nivel5 MAYOR nivel6
	nivel5 MENOR_IGUAL nivel6
	nivel5 MAYOR_IGUAL nivel6
	nivel5 IGUAL nivel6
	nivel5 OR nivel6
	nivel5 AND nivel6
	nivel6
nivel6	NOT term
	term
term	ENT
	DEC
	SCI
	ID

Gramática

A continuación se muestra unos ejemplos aplicando toda la gramática expuesta en el lenguaje de programación PPS:

Código

El código fuente de PPE es el siguiente:



```
Terminal - maquina@JDC-HP
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
maquina@JDC-HP-ProBook-6460b:~$ ./ppe
INICIOPROGRAMA
      r:=2*6;
      q:=79,31;
FINPROGRAMA
cadena valida
maquina@JDC-HP-ProBook-6460b:~$
```

```
Terminal - maquina@JDC-HP-ProBook-6460b:~$  
Archivo Editar Ver Terminal Pestañas Ayuda  
maquina@JDC-HP-ProBook-6460b:~$ ./ppe  
INICIOFUNCION aumento{a} REALIZAR  
    SI {a>0} REALIZAR  
        x:=9;  
    SINO  
        x:=0;  
    FINSI  
    DEVUELVE x;  
FINFUNCION  
INICIOPROGRAMA  
    x:=aumento{2};  
FINPROGRAMA  
cadena valida  
maquina@JDC-HP-ProBook-6460b:~$
```

```
Terminal - maquina@JDC-HP-ProBook-6460b:~$  
Archivo Editar Ver Terminal Pestañas Ayuda  
maquina@JDC-HP-ProBook-6460b:~$ ./ppe  
INICIOFUNCION mediaAritmetica{r;q} REALIZAR  
    x:=(r+q)/2;  
    DEVUELVE x;  
FINFUNCION  
  
INICIOFUNCION sumar{r;q} REALIZAR  
    x:=r+q;  
FINFUNCION  
  
INICIOPROGRAMA  
    r:=4;  
    q:=((3*r)/5)+5^3-4,98;  
    x:=mediaAritmetica{r;q};  
    sumar{r;q};  
FINPROGRAMA  
cadena valida  
maquina@JDC-HP-ProBook-6460b:~$
```

```
maquina@JDC-HP-ProBook-6460b:~$ ./ppe
INICIOFUNCION suman{n} REALIZAR
    x:=0;
    PARA {i:=1;i<=n;i:=i+1} REALIZAR
        x:=x+i;
    FINPARA
    DEVUELVE x;
FINFUNCION

INICIOFUNCION sumarMientras{n} REALIZAR
    x:=0,0;
    i:=0;
    MIENTRAS {i<=n} REALIZAR
        x:=x+i;
        i:=i+1;
    FINMIENTRAS
    DEVUELVE x;
FINFUNCION

INICIOPROGRAMA
    r:=suman{n};
    q:=sumarMientras{n};
    MIENTRAS {r<=q} REALIZAR
        r:=r+2,5;
    FINMIENTRAS

    m:=1,0;
    PARA {l:=1;l<=10;l:=l+1} REALIZAR
        m:=m*i;
    FINPARA
FINPROGRAMA
cadena valida
maquina@JDC-HP-ProBook-6460b:~$
```