Universidad Autónoma del Estado de México

UAEM

CU UAEM ZUMPANGO

CENTRO UNIVERSITARIO
UAEM ZUMPANGO

INGENIERÍA EN COMPUTACIÓN

Unidad de Aprendizaje:

PROGRAMACIÓN 1

Unidad de Competencia I:

ELEMENTOS DE UN PROGRAMA Y SU REPRESENTACIÓN EN PSEUDOCÓDIGO Y UML.

M. en C. Edith Cristina Herrera Luna

Agosto 2019

P R O G R A M A C I Ó N 1

+ OBJETIVO DE LA UNIDAD DE APRENDIZAJE:

Desarrollar programas orientados a objetos con interfaces textuales de usuario, aplicando análisis de sustantivos, diagramas de clase, arquitectura modelo-vista- controlador, y lenguaje de programación orientado a objetos para la resolución de problemas simples.

INTRODUCCIÓN

- El Ingeniero en Computación es el profesional que posee los conocimientos y habilidades en el desarrollo de sistemas computacionales, diseño y mantenimiento de hardware, comunicaciones y redes de computadoras así como en la administración de recursos computacionales.

- Una de las principales actividades del Ingeniero en Computación es la programación, cuyas bases deben ser adquiridas en su formación. La programación, como una parte de la informática también evoluciona continuamente. Hoy en día existen varios paradigmas de programación y entre los más conocidos y utilizados está el paradigma orientado a objetos, en el cual se centra este programa de la unidad de aprendizaje. La importancia de esta programación radica en que, favorece la creación de programas de calidad, fuerza en mantenimiento, en extensión y reutilización de programas.

INTRODUCCIÓN

- Está basada en el modo de pensar del hombre y en el modo de trabajar de la máquina, el elemento básico de esta programación no es solo la función sino un ente denominado objeto. Es importante recalcar que se retoman algunas herramientas importantes del paradigma estructurado como el diseño descendente, la modularidad y los registros.

- Esta unidad de aprendizaje tiene la finalidad de introducir al alumno al ámbito de la programación orientada a objetos en sus elementos básicos y proporcionarle los conocimientos necesarios y suficientes para que aplique el análisis de sustantivos, diseño de descendente, abstracción de datos y procedimientos para el desarrollo de algoritmos, haciendo uso de estructuras de datos lineales.

UNIDAD TEMÁTICA I:

ELEMENTOS DE UN PROGRAMA Y SU REPRESENTACIÓN EN PSEUDOCÓDIGO Y UML

OBJETIVO:

Realizar análisis de sustantivos y diseño descendente, determinando casos de uso, abstracción de procedimientos, diagramas de UML, estructuras de control, diagramas de flujo de datos, pseudocódigos y pruebas de escritorio para la elaboración de algoritmos simples.

PROGRAMACIÓN 1

CONTENIDO

1. Elementos básicos de la programación
   - 1.1 Tipos de datos
   - 1.2 Constantes y variables.
   - 1.3 Operaciones primitivas elementales.
2. Clases y objetos
   - 2.1 Abstracción y encapsulamiento.
   - 2.2 Objetos, clases, atributos y métodos.
   - 2.3 Ocultamiento y modificadores de acceso: público y privado.
   - 2.4 Instanciación de objetos
3. Diagramas UML: clases y secuencia
4. Estructuras de control

ELEMENTOS BÁSICOS DE LA PROGRAMACIÓN

PRO1 / ICO / ECHL

PROGRAMACIÓN 1

ELEMENTOS BÁSICOS DE LA PROGRAMACIÓN

+ ¿Qué es un programa?

Un **programa** es la implementación de un **algoritmo** en un determinado **lenguaje de programación**.

+ Pero… ¿qué es un algoritmo?

Es una secuencia de pasos *lógicos*, *precisos* y *finitos* que permiten resolver un problema.

Si se ejecuta varias veces un algoritmo (en iguales condiciones), se debe obtener para cada una de ellas el mismo resultado y se dice que es *determinista*.

ELEMENTOS BÁSICOS DE LA PROGRAMACIÓN

+ Como ya se menciono un **programa** es la implementación de un algoritmo en un determinado lenguaje de programación, esta formado por fragmentos individuales de código denominados **sentencias.**

+ Una sentencia representan operaciones del algoritmo y se van ejecutando en el orden establecido en el algoritmo.

- Da ejemplos de cuando se usa un algoritmo, identifica sus características y describe algunos de ellos
- Investiga como se realizaban programas de computo anteriormente.

ELEMENTOS BÁSICOS DE LA PROGRAMACIÓN

- Existen diferentes tipos de lenguajes de programación:
  - Lenguaje Maquina
  - Lenguaje Ensamblador
  - Lenguaje de Alto Nivel

prac1.c

```c
#include <stdio.h>
int main()
{
    printf("HOLA  MUNDO....");
    printf("Este es un codigo fuente");
    return 0;
}
```

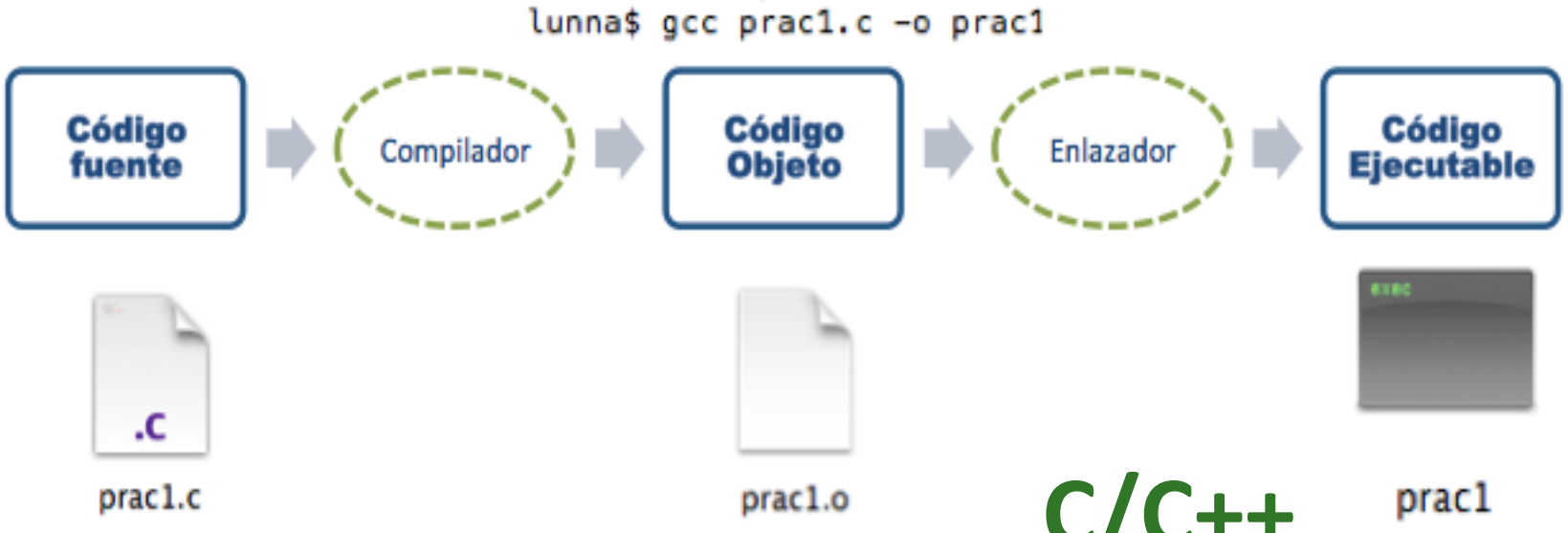
prac1.s

```
    .cstring
LC0:
    .ascii "HOLA  MUNDO....\0"
    .text
.globl _main
_main:
LFB3:
    pushq   %rbp
LCFI0:
    movq    %rsp, %rbp
LCFI1:
    leaq    LC0(%rip), %rdi
    movl    $0, %eax
    call    _printf
    movl    $0, %eax
    leave
    ret
LFE3:
    .section __TEXT,__eh_frame,coalesced,r
EH_frame1:
    .set L$set$0,LECIE1-LSCIE1
    .long L$set$0
LSCIE1:
```

- Código Fuente:
  - Es un algoritmo escrito en un lenguaje de programación específico y no puede ser ejecutado directamente por la computadora

¿CÓMO SE CREA UN PROGRAMA?

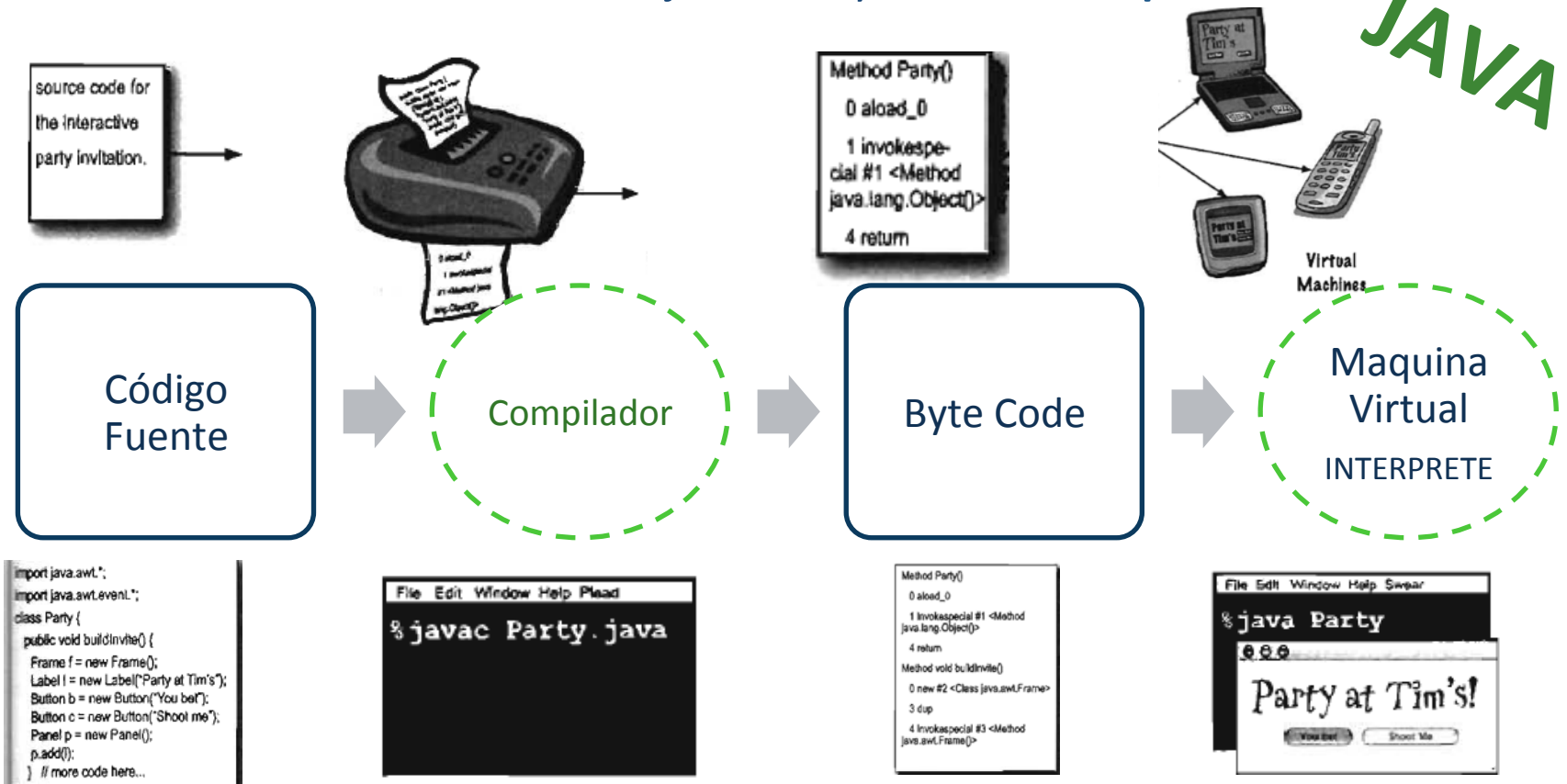
+ Algunos lenguajes de programación son **compilados**, pero….



Realiza el programa “Hola mundo”, verifica los archivos originados por la compilación.

¿CÓMO SE CREA UN PROGRAMA?

+ Actualmente debido a sus ventajas, la mayoría son **interpretados**.



Party.java

Party.class

- Imágenes del Libro: "Head First Java" (Ver Referencias)

TIPOS DE DATOS

+ Ya que las computadoras usan un lenguaje binario (0 y 1), es necesario que todos los elementos de un programa sean traducidos a un formato binario.

+ **Un tipo de dato especifica la forma en que se va a emplear un cierto numero fijo de bits para representar internamente una determinada clase de información.**

+ Los tipos básicos o propios del lenguaje se conocen como ***tipo de dato primitivo*** o ***atómicos*.**

- Investiga los tipos de datos primitivos, tamaño en bytes y rangos de datos.

TIPOS DE DATOS

- En C/C++ se tiene el dato primitivo **bool**, mientras que en Java el dato primitivo es **boolean**.

Tipo Dato	Bits / Bytes	Menor Dato	Mayor Dato
byte			
short			
int			
long			
float			
double			
char			
bool / boolean			

CONSTANTES Y VARIABLES

+ **Identificador:** Es el nombre que se le da a un elemento dentro de un programa. Esta formado por una serie de caracteres, dígitos y guiones de subrayado.

1. Debe iniciar con una letra del alfabeto ingles. Posteriormente pueden ir letras, dígitos y/o guiones de subrayado (en cualquier orden).
2. No se pueden usar símbolos especiales.
3. El identificador no puede ser una *palabra reservada*.
4. Las letras mayúsculas y minúsculas son diferentes.

- Investiga las palabras reservadas del lenguaje

CONSTANTES Y VARIABLES

- CONSTANTE:
  - Un dato que no cambia su valor durante el tiempo de ejecución de un programa. Generalmente esta acompañado de un identificador, pero puede ser representado solo con un valor.

- VARIABLE:
  - Un dato que puede cambiar su valor durante el tiempo de ejecución de un programa.
  - Es un fragmento de memoria del cual se conoce su posición (el byte en que comienza), su extensión (el numero de bytes que ocupa en total) y tipo (el esquema de codificación que se ha utilizado en su construcción).

CONSTANTES Y VARIABLES

+ ¿Cuántas constantes y variables identificas en el siguiente código?

```c
#include <stdio.h>
#include <math.h>

int main(){

    float r, area, perim;
    float* a, b, c;

    printf("\nIngresa el radio del circulo ");
    scanf("%f", &r);

    a = &r;
    b = &area;
    c = &perim;

    *c = 2.0f * 3.1415f * (*a);
    *b = 3.1415 * pow( (*a), 2);

    printf("\nPerímetro: %f ", *c);
    printf("\nArea: %f \n", *b);

    return 0;
}
```

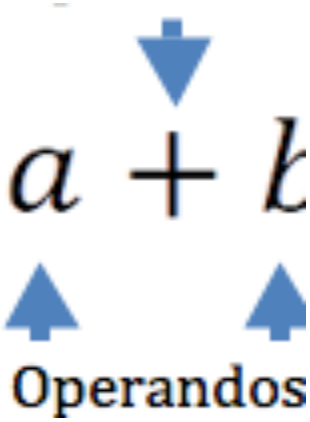
OPERACIONES PRIMITIVAS ELEMENTALES

- **EXPRESIÓN**: Es una combinación de constantes, variables y operadores construida de tal modo que produce un resultado.

- **OPERADORES**: Son símbolos del lenguaje que permiten especificar operaciones, van acompañados de uno o más operandos.

Se pueden clasificar de acuerdo al número de operandos que los acompañan:

- Operador Unario: un solo operando. var++;
- Operador Binario: dos operandos. var1 * var2;



OPERACIONES PRIMITIVAS ELEMENTALES: OPERADORES

Según la operación que realizan pueden clasificarse en :

- Asignación
- Manipulación de bits
- Aritméticos
- Relacionales
- Lógicos

- Investiga las jerarquía de operadores
- Investiga las reglas para evaluar expresiones

OPERACIONES PRIMITIVAS ELEMENTALES: OPERADORES

- Operadores de asignación:
  - Se usan para asignar/dar un valor a una variable.

    Pseudocódigo: variable → expresión

    Lenguaje C / C++ / Java variable = expresión ;

- Operadores de Manipulación de bits:
  - Operadores que permiten trabajar a nivel de bits en un dato:

    <<, >>, &, !, |,

- ¿Cómo funcionan los operadores de corrimientos y para que son utilizados?

OPERACIONES PRIMITIVAS ELEMENTALES: OPERADORES

+ Operadores Aritméticos

Operador	
Suma	+
Resta	-
Multiplicación	*
División	/
Modulo	%

Pseudocódigo

variable ← variable + variable

Lenguaje C / C++ / Java

variable = variable + variable ;

OPERACIONES PRIMITIVAS ELEMENTALES: OPERADORES

+ Operadores Lógicos:

A	B	A AND B	A OR B	NOT A
		A && B	A || B	!A
V	V	V	V	F
V	F	F	V	
F	V	F	V	V
F	F	F	F	

+ Operadores Relacionales:

Operador	
Mayor	>
Mayor o igual	>=
Menor	<
Menor o igual	<=
Igual	==
Diferente	!=

NOTA: Observa que para asignar un valor a una variable se usa un solo signo '=', mientras que para hacer una comparación u operación relacional se usan dos signos '=='.

OPERACIONES PRIMITIVAS ELEMENTALES: FUNDAMENTOS DEL LENGUAJE

Antes de usar una variable debe ser **declarada**, es decir, se avisa al compilador el tipo de dato de la variable y su identificador.

Una variable puede ser **inicializada** con un valor al ser declarada.

Sintaxis:

Declaración

tipo_de_dato identificadorVar1, identificadorVar2;

Ej.
```
int dato1, dato2, dato3;
double resultado;
float gastoTotal, gastoDiario;
```

Inicialización

tipo_de_dato variable = valor;

Ej.
```
char letra = 'a';
double suma = 8.0, resta = 9.5;
```

Asignación

variable1 = valor;

Ej.
```
calificacion = 7.8;
```

variable1 = expresión;

Ej.
```
calificacion = 5.0 + 3.5 ;
```

variable1 = variable2;

Ej.
```
calificacion = promedio;
```

SALIDA/IMPRESIONES EN PANTALLA

Una instrucción para desplegar mensajes en pantalla en el lenguaje…

+ C

printf(“Mensaje entre comillas dobles ”);

printf(“%modificadorDeFormato”, variable);

% [formato] [ancho] [.precisión]

formato:

-	Justificado a la izquierda
+	Preceder el valor con un signo + ó -
(espacio)	Predece los valores positivos con un espacio
0	Llena la longitud del valor con ceros
#	Si la salida es en formato octal precede con 0. Si la salida es en formato hexadecimal precede con 0x

ancho y precisión: Número entero.

OPERACIONES PRIMITIVAS ELEMENTALES: FUNDAMENTOS DEL LENGUAJE

+ C++

```
cout<< “Mensaje entre comillas dobles ” ;
cout<< “mensaje y ” << variable << endl;
cout << variable ;
```

+ Java

```
System.out.print( “Mensaje ” + variable );
System.out.println( “Mensaje y salto de línea” );
System.out.println( variable );
System.out.printf( “Salidas con formato, como en C:  %d”, varInt );
```

ENTRADA/LECTURA DE DATOS

Una instrucción para leer datos desde teclado/entrada estándar es:

+ C

```
scanf( “%modFormato”, &variable );
scanf( “%modFormato %modFormato”, &variable, &variable);
```

+ C++

```
cin>> variable;
cin>> variable1 >> variable2 >> variableN;
```

OPERACIONES PRIMITIVAS ELEMENTALES: FUNDAMENTOS DEL LENGUAJE

- Java

Importar: import java.util.Scanner;

Crear objeto Scanner:

```
Scanner identif = new Scanner( System.in );
```

Lectura de datos:

```
varInt = identiif.nextInt( );
varFloat = identif.nextFloat( );
…
varString = identif.next( );
varString = identif.nextLine( );
```