



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ZUMPANGO

PROGRAMACIÓN 1

EJERCICIOS Y PROGRAMAS

Ingeniería en Computación

Profesor:

M. en C. Edith C. Herrera Luna

PRESENTACIÓN DEL PROGRAMA DE ESTUDIOS

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. 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.

OBJETIVOS 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.

PRESENTACIÓN

La selección de ejercicios en el presente documento pretende sentar las bases y conocimientos introductorios a las materias de programación, ya que al ser la primera asignatura de esta área en la carrera de Ingeniería en Computación, se entiende que no todos los alumnos tienen estas bases, pero deben ser involucrados amigablemente y de una manera sencilla.

Los ejercicios propuestos van de un menor a mayor grado de dificultad e incorporan los conceptos previamente tratados en clase, inicialmente los ejercicios cubren un solo concepto, pero conforme se avanza en los ejercicios se incorporan dos o tres elementos para dar paso a programas o ejercicios más complejos. De tal forma que poco a poco el alumno toma nuevos conocimientos pero repasa los ya aprendidos en los primeros temas.

El programa de la asignatura contiene una parte de documentación y una de implementación o codificación, con el objetivo de que el alumno, practique esas dos fases de manera conjunta, para mejorar su desarrollo.

Se cubre la programación orientada a objetos, pero se tienen ejercicios introductorios a la programación (programación estructurada) para ayudar al alumno a cambiar su forma de abstracción necesaria para los diversos paradigmas de programación.

Se recomienda el uso del lenguaje C++ o Java para su codificación.

CONTENIDO

TIPOS DE DATOS DEL LENGUAJE	5
EJERCICIO 1	5
IDENTIFICADORES Y PALABRAS RESERVADAS	5
EJERCICIO 2	5
EXPRESIONES Y OPERADORES	6
EJERCICIO 3	6
EJERCICIO 4	7
EJERCICIO 5	8
EJERCICIO 6	8
ATRIBUTOS Y MÉTODOS	9
EJERCICIO 7	9
MODIFICADORES DE ACCESO	9
EJERCICIO 8	9
DIAGRAMAS DE CLASE	10
EJERCICIO 9	10
PROGRAMA 10	11
PROGRAMA 11	11
PROGRAMAS ESTRUCTURAS SELECTIVAS	12
PROGRAMA 12	12
PROGRAMA 13	13
PROGRAMA 14	13
PROGRAMA 15	13
PROGRAMA 16	13
PROGRAMAS ESTRUCTURAS ITERATIVAS	14
PROGRAMA 16	14
PROGRAMA 17	14
PROGRAMA 18	14
PROGRAMA 19	14
PROGRAMA 20	14

TIPOS DE DATOS DEL LENGUAJE

Ejercicio 1

Para cada uno de los datos de la siguiente tabla, anota del lado derecho el tipo de dato al que corresponde de acuerdo a los datos del lenguaje C++ / Java.

DATO	TIPO DE DATO
8, 25, 9, 500, 205, 18	Entero / short
'%', 'a', 'z', '5', '?', 'M',	
-10, -783, -202323, -586, -1245	
"casa", "amarillo", "tarea"	
8, 7.9, 4.56, 0.3, 0.895, 0.6	
"a", "B", "#", "C", "z", "-"	
'1', '5', '7', '0'	
0.00003453, 0.123235499786	
true, false	
"456", "178", "267", "9145"	

IDENTIFICADORES Y PALABRAS RESERVADAS

Ejercicio 2

Para cada identificador marca si es correcto no usarlo de acuerdo al lenguaje C y las reglas de identificadores. Además, si no es correcto escribe porqué y escribe un identificador valido.

IDENTIFICADOR	CORRECTO		¿POR QUÉ? (En caso de no ser valido)	IDENTIFICADOR VALIDO
	SI	NO		
agua	X		----	-----
Agua%6		X	No se puede usar el símbolo %	Agua6
76colores				
capacidad				
hora:salida				
altura_niño				
metro2				
_casa				

codigo barras				
paginas_				
#simbolo				
símbolo				
456hojas				
sitio4web				
apellido_Mat				
case				
Int				
integer				

EXPRESIONES Y OPERADORES

Ejercicio 3

Indica cuál es el resultado de evaluar (manualmente) cada una de las siguientes expresiones. Realiza cada paso para evaluar la expresión.

Sean $a=15$, $b=2$, $c= -5$ $d= 19$ y $e=-3$, evalúa cada expresión.

EJEMPLO:

$$\begin{aligned}
 a \% b + 12 / (3 - e) + 10 &= 15 \% 2 + 12 / (3 - -3) + 10 \\
 &= 15 \% 2 + 12 / 6 + 10 \\
 &= 1 + 12 / 6 + 10 \\
 &= 1 + 2 + 10 \\
 &= 3 + 10 \\
 &= 13
 \end{aligned}$$

- I. $6 + a / 5 * b$
- II. $a / b * b$
- III. $c / d / e * 3 \% 10$
- IV. $(-1) * (c - d \% 3 + 18 / b) * (1 / 3)$
- V. $7 - 6 / 3 + 2 * 3 / 2 - 4 / 2$
- VI. $5 * 5 (5 + (6 \% 2) + 1)$
- VII. $7 * 10 - 5 \% 3 * 4 + 9$

VIII. $(16 \gg 2) * (1 \ll 4)$

Ejercicio 4

Escribe las siguientes expresiones aritméticas como expresiones de computadora (algorítmicas o propias para el lenguaje Java).

La potencia puede hacerse con la función **pow()**, por ejemplo:

$$x^2 == \text{pow}(x, 2), (a - b)^3 == \text{pow}(a - b, 3).$$

EJEMPLO:

$$\sqrt{a^2 + 7}$$

$$\text{sqrt}(\text{pow}(a, 2) + 7)$$

I. $3x + y$

II. $\frac{x+y}{7}$

III. $\frac{8x+y}{z+17}$

IV. $\left(\frac{9}{5}\right) (^{\circ}\text{C} + 32)$

V. $\left(2x^2 + \frac{y}{3}\right) ((z - 1)^3)$

VI. $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$

VII. $\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$

Ejercicio 5

Evalúa cada una de las siguientes expresiones indicando paso por paso. Usa las tablas de verdad de los operadores AND, OR y NOT.

Toma en cuenta que son variables enteras (int) $x=9$, $y=3$, $z=5$; son variables reales (float) $m=4.5$, $n=-9.8$; $p=5.7$ y son variables tipo carácter (char) $a='A'$, $b='@'$, $c='+$

EJEMPLO:

$$\begin{aligned} & (x < y \ \&\& \ y < z) \ || \ (b == '@') \ || \ (n >= p) \\ = & (F \ \&\& \ V) \ || \ (V) \ || \ (F) \\ = & (F) \ || \ V \ || \ F \\ = & F \ || \ F \\ = & F = \underline{\underline{FALSO}} \end{aligned}$$

- I. $(x >= y \ \&\& \ x >= z \ \&\& \ x >= x)$
- II. $(m + 8) > p \ \&\& \ (x - y) > z$
- III. $!(m > z)$
- IV. $((x + y) != 5) \ || \ (b != c) \ || \ (a == b)$
- V. $(z >= m - 2) \ || \ (x < y) \ \&\& \ (m < p)$
- VI. $(x \% y) >= p \ \&\& \ !(m * y <= z * y)$

Ejercicio 6

Explica el funcionamiento de los operadores unarios :

++variable

variable++

-- variable

variable --

ATRIBUTOS Y MÉTODOS

Ejercicio 7

De las siguientes palabras, marca con una ✓ aquellas que consideres como atributos de una clase, marca con un ✗ las palabras que sea un método de una clase.

nombreAlumno	enrutamiento
construyeCasa	perro
calculaPromedio	correr
promedio	comer
ladrar	tiempo
tipoDeFuente	sol
tipoDeDocumento	rotación
horaDeAlarma	agua
sistemaOperativo	amanecer
velocidad	enrutamiento

MODIFICADORES DE ACCESO

Ejercicio 8

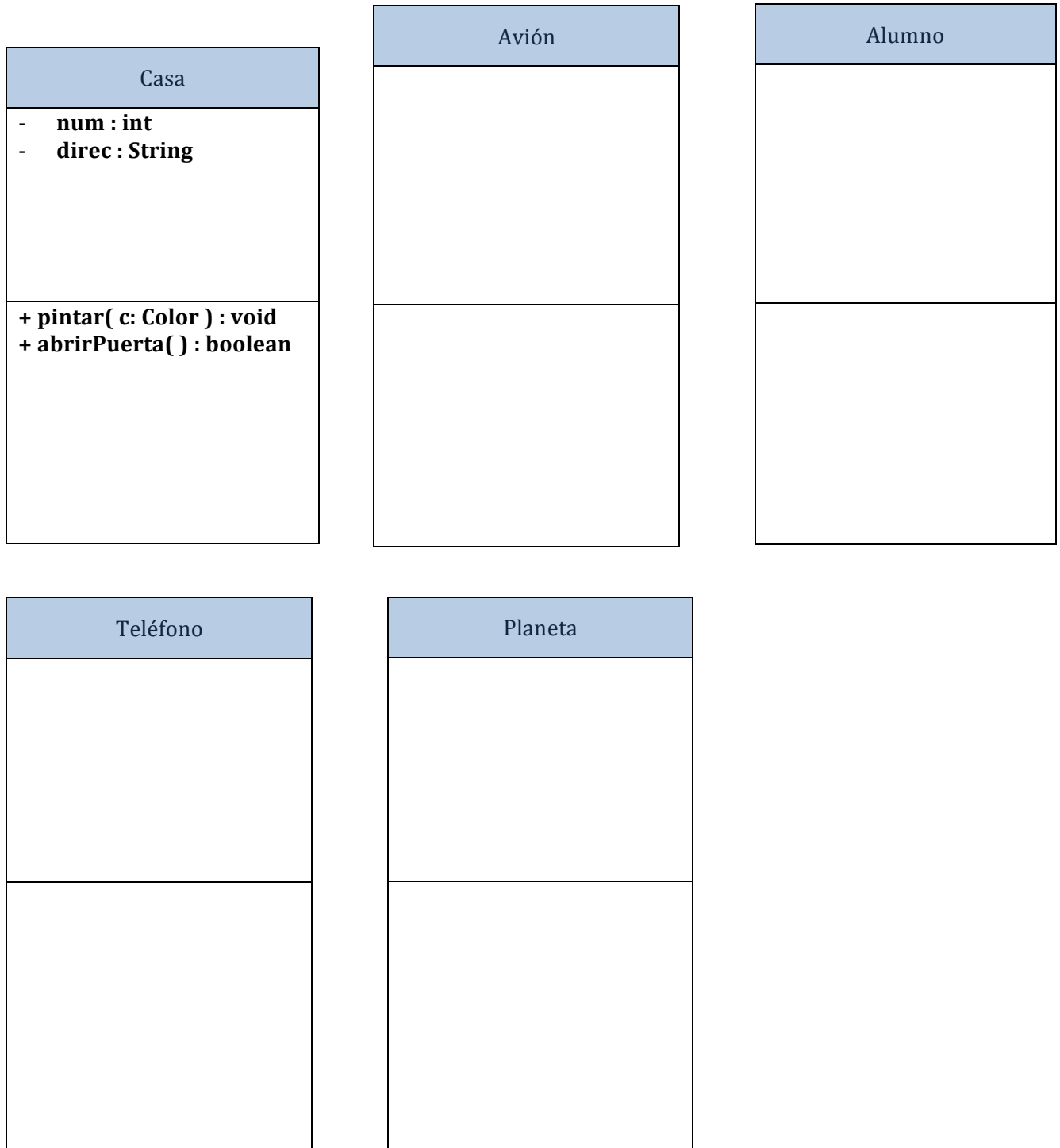
Investiga los modificadores de acceso siguientes, agrega el símbolo con el que se representan.

- public
- private
- protected
- por default

DIAGRAMAS DE CLASE

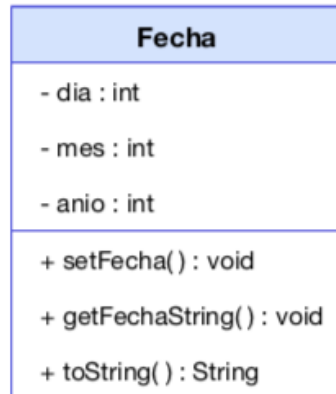
Ejercicio 9

Llena los siguientes diagramas de clase colocando entre 3 y 4 atributos y métodos en cada uno de ellos.



PROGRAMA 10

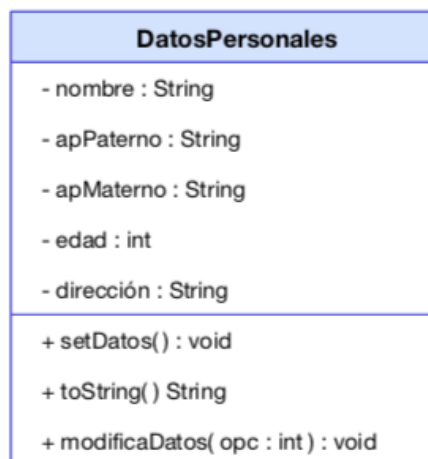
Realiza el código fuente correspondiente al siguiente diagrama.



- El método `getFechaString` imprime la fecha pero el mes desplegado como nombre y no como numero de mes.
- Los métodos `set_____() : void` piden la información para cada atributo
- Los métodos `toString() :` devuelven toda la información del objeto como una cadena de texto.

PROGRAMA 11

Realiza el código fuente correspondiente al siguiente diagrama.



- Los métodos `set_____() : void` piden la información para cada atributo
- Los métodos `toString() :` devuelven toda la información del objeto como una cadena de texto.
- El método `modificaDatos` cambia un atributo específico de clase de acuerdo a un parámetro que recibe como entrada, por ejemplo, se puede convenir que si la opción dada es 10 cambie el nombre, si es 20 el `apPaterno`, si es 30 `apMaterno` y así sucesivamente.

PROGRAMAS ESTRUCTURAS SELECTIVAS

Codifica los siguientes programas:

PROGRAMA 12

Construir un programa que admita las dimensiones de dos rectángulos y diga cual de ellos tiene mayor área.

```
1  #include <stdio.h>
2
3  int main(){
4      float a, b, c, d, area1, area2;
5      printf("\nCalcula el área mayor de 2 rectangulos");
6      printf("\nIngresa la base y altura del rectángulo 1");
7      scanf("%f", &a);
8      scanf("%f", &b);
9      printf("\nIngresa la base y altura del rectangulo 2");
10     scanf("%f %f", &c, &d);
11
12     area1 = a*b;
13     area2 = c*d;
14
15     printf("\nArea1: %.3f \tArea2: %.2f", area1, area2);
16
17     if( area1 > area2 )
18         printf("\nArea del rectangulo 1 mayor\n");
19     else{
20         if( area2 > area1 )
21             printf("\nArea del rectangulo 2 mayor\n");
22         else
23             printf("Los rectangulos tienen"
24                 "la misma area\n");
25     }
26     return 0;
27 }
```

PROGRAMA 13

En una sola clase, crea métodos estáticos que:

- Admitan un numero entero e indique si es múltiplo de 2, de 3 o de 5.
- Admitan un numero entero e indique si es múltiplo de 2, de 3 y de 5 simultáneamente.
- Admitan un numero entero e indique si es múltiplo de 2 pero no de 3.

PROGRAMA 14

Programa que calcula el perímetro de las figuras: Circulo, cuadrado, polígono regular o triangulo, en base a la elección del usuario (construir un menú).

PROGRAMA 15

Pide un nombre al usuario, almacénalo en una sola variable y posteriormente imprime a) el nombre, b) el apellido paterno ó c) el apellido materno, de acuerdo a la elección del usuario.

PROGRAMA 16

Crea un programa que usando la clase Math de llama y métodos estáticos indique:

- El mayor de 2 números
- El menor de 3 números
- Demuestre el uso de la función pow

```
1 import java.util.Scanner;
2 public class Programa6
3 {
4     public static void Mayor()
5     {
6         Scanner x=new Scanner(System.in);
7         int a,b,c;
8         System.out.println("Ingresa 2 num");
9         a=x.nextInt();
10        b=x.nextInt();
11        c=Math.max(a,b);
12        System.out.println("El mayor es: "+c);
13    }
14
15    public static void Menor()
16    {
17        Scanner x=new Scanner(System.in);
18        float i,j,y,z,k,r;
19        System.out.println("Ingresa 3 num");
20        i=x.nextInt();
21        j=x.nextInt();
22        y=x.nextInt();
23        z=Math.min(i,j);
24        k=Math.min(y,z);
25        System.out.println("El menor es: "+k);
26    }
}
```

```

27
28 public static void Potencia()
29 {
30     Scanner x=new Scanner(System.in);
31     int i,j,y,k,r;
32     double z, p;
33
34     System.out.println("Ingresa 4 num");
35     i=x.nextInt();
36     j=x.nextInt();
37     y=x.nextInt();
38     k=x.nextInt();
39     z=Math.pow(i,y);
40     r=Math.max(j,k);
41     p=Math.sqrt(r);
42     System.out.println("La Potencia es: "+z);
43     System.out.println("La raiz es: "+p);
44 }
45 //METODO
46
47 public static void main(String[]args)
48 {
49     Programa6.Mayor();
50     Programa6.Menor();
51     Programa6.Potencia();
52 }//MAIN
53 }

```

PROGRAMAS ESTRUCTURAS ITERATIVAS

Codifica los siguientes programas:

PROGRAMA 16

Construir un programa que indique el número de vocales que tiene una frase ingresada por el usuario.

PROGRAMA 17

Programa que imprima los primeros N números de la secuencia Fibonacci.

PROGRAMA 18

Programa que calcula el máximo común divisor de dos números positivos.

PROGRAMA 19

Programa que invierte los dígitos de un número dado.

PROGRAMA 20

Programa que imprime un triangulo de tamaño N con un carácter dado por el usuario.

BIBLIOGRAFÍA

Básica:

Cairó Osvaldo (2005). Metodología de la programación. Algoritmos, diagramas de flujo y programas. Alfaomega Grupo Editor ISBN 970-15-1100-X

Guardati Buemo Silvia. (2016). Estructuras de datos básicas programación orientada a objetos con java. Alfaomega Grupo Editor.

López Goytia, José Luis y Ángel Gutiérrez González. (2014). Programación orientada a objetos con C++ y Java. Un acercamiento disciplinario. Grupo Editorial Patria.

López José, Gutiérrez Ángel. (2017). Programación orientada a objetos con C++ y Java. Grupo Editorial Patria

López Román, Leobardo. (2013). Metodología de la programación orientada a objetos. Alfaomega Grupo Editor. 2a Ed.

Martínez, Román y Elda Quiroga. (2004). Estructuras de datos. Referencia práctica orientada a objetos. Thomson Learning.

Complementaria:

Corona Nakamura María Adriana, Ancona Valdez María de los Ángeles. (2011) Diseño de Algoritmos y su Codificación en C. Mc Graw Hill.

Joyanes Aguilar, Luis. (2008). Fundamentos de Programación: Algoritmos, Estructuras de datos y objetos. McGraw Hill. 4a Ed.

Tamayo Silva F LIX, Tamayo Silva Feliz Manual. (2012). Fundamentos de Lógica de Programación. Eae Editorial Academia Española.