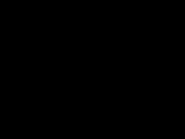


**Guía estudio para Examen de Pre-Certificación Bancomer**

[ Java]



1



Versión: [1.0]

Fecha de Publicación: Diciembre 17, 2012

Confidencialidad del documento

Todos los derechos reservados. El contenido de este documento es propiedad de Softtek® tiene derechos de autor registrados y es confidencial. Cualquier reproducción completa o parcial está estrictamente prohibida sin el permiso escrito de Softtek®.

Este documento está sujeto a cambios. Comentarios, correcciones o preguntas diríjanse al autor

2



# Sumario del Documento

**Propósito**

El propósito de este documento es proporcionar al usuario una herramienta de apoyo con temas claves que faciliten su integración a la cuenta de Bancomer. Contiene conceptos generales y presenta una serie de ejercicios de práctica incluyendo su resolución.

Este documento está dirigido a colaboradores con perfil Java.

**Tabla de Revisiones**

La siguiente tabla lista las revisiones hechas a este documento. Se usa para describir los cambios y adiciones cada vez que este documento es actualizado. La descripción deberá incluir tanto detalle como sea posible, así como los revisores que solicitaron los cambios.

|  |  |  |
| --- | --- | --- |
| Fecha | Autor | Descripción de Cambios |
| 10-Ago-2012 | Gerardo López Rojas | Versión inicial |
| 17-Dic-2012 | Juan Macedo Montes de Oca | Versión 2 |
| 01-Mar-2019 | Elizabeth Colin Vaughan | Eliminar la parte de documentos Bancomer, agregar temas JAVA: Colecciones, hilos, listas, Clases |

3



# Guía estudio para Examen Pre-Certificación Bancomer

Perfil Java

**Contenido**

[Sumario del Documento 3](#_TOC_250008)

[Propósito 3](#_TOC_250007)

[Tabla de Revisiones 3](#_TOC_250006)

[Guía estudio para Examen Pre-Certificación Bancomer 4](#_TOC_250005)

[Contenido 4](#_TOC_250004)

[Sección 1. Metodología Bancomer 5](#_TOC_250003)

[Sección 2. SQL](#_TOC_250002) 8

[Sección 3. Lógica 1](#_TOC_250001)5

[Sección 4. Java 1](#_TOC_250000)8

4

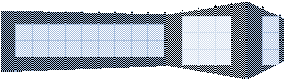


# Sección 1. Metodología Bancomer

## Objetivo.

Esta sección tiene la finalidad de presentar los diversos tipos de pruebas aplicadas dentro del ciclo de vida del desarrollo de software, lo cual permitirá interactuar con mayor fluidez dentro del equipo de trabajo.

5



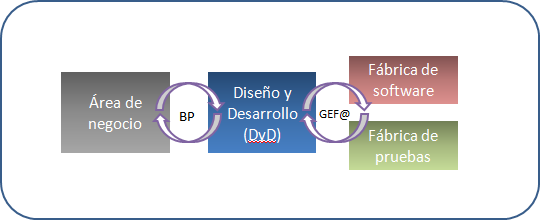
## Modelo de estructura organizacional proyectos Bancomer.

El siguiente diagrama muestra, de forma general, las diversas áreas que participan en el desarrollo de proyectos de software en Bancomer.

Consideraciones:

* **BP:** Business Partner o Socio de Negocio.
* **DyD:** Área de Diseño y Desarrollo.
* No todos los proyectos integran la participación de la Fábrica de pruebas, en este caso la responsabilidad de la ejecución de pruebas corresponderá al área del DyD

.

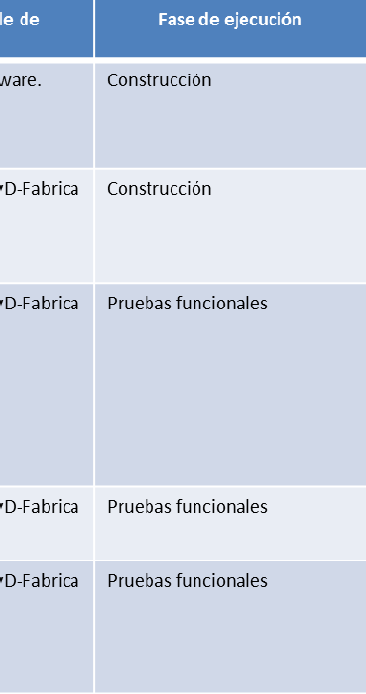
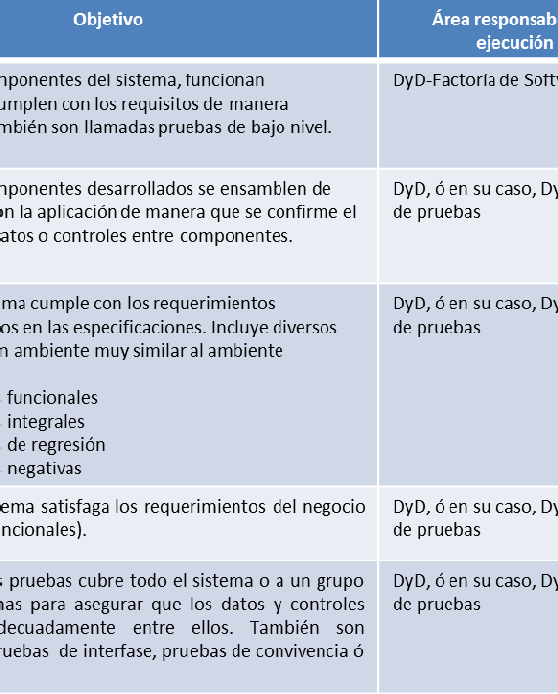
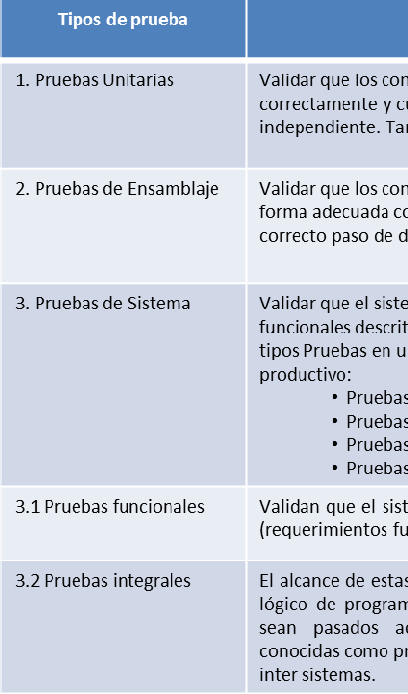


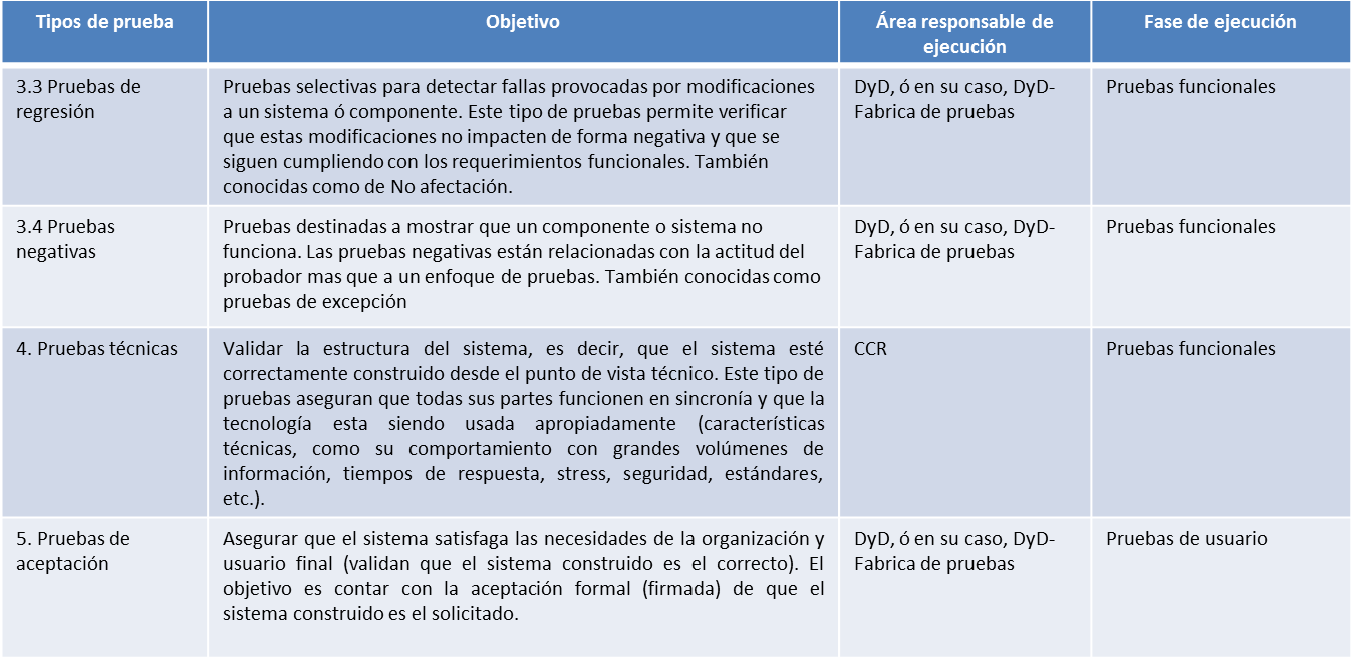
Quedan pocos proyectos con esta estructura, hoy en día se trabaja con la visión “agil”.

6



## Tipos de pruebas de la metodología Bancomer.





7



# Sección 2. SQL

## Objetivo.

Esta sección tiene la finalidad de reforzar los conocimientos de SQL. Elaborando las sentencias en el lenguaje.

## Recomendaciones.

1. Lee detalladamente el código de los ejercicios.
2. Verificar que la sintaxis y la semántica del código sea correcta.
3. Elaborar las sentencias en SQL puro.

8



**Ejercicios de SQL – Parte 1.** Elaborar la sentencia de SQL en cada uno de los siguientes casos:

*Tiempo estimado para resolver los ejercicios: 20 minutos.*

1. Insertar en una tabla “PRODUCTO” el registro con los siguientes valores:

La clave del producto es “1342”, el nombre del producto es “LASER”, el precio es de “1250.00” y pertenece a la fabrica “4”.

|  |
| --- |
| TABLA PRODUCTO |
| ID\_PRODUCTO |
| NOMBRE |
| PRECIO |
| ID\_FABRICA |

1. De la Tabla PRODUCTO eliminar los elementos que cuyo precio sea entre 5 y 10.
2. En la tabla PRODUCTO se requiere actualizar el nombre por “Scanner” de los elementos que tengan en ID \_FABRICA igual a “4”.
3. Realizar una consulta de la tabla PRODUCTO y mostrar los nombres y precios de los productos ordenados de manera ascendente por nombre y descendente por precio.
4. Efectuar una consulta de la tabla PRODUCTO y mostrar el promedio del precio de todos los productos que en el campo nombre aparezcan impresora.

9



1. Realizar una consulta de las tablas TABLA\_EMPLEADO y EMPLEADO\_DETALLE y mostrar todos los datos de los empleados incluyendo su “IS” y “SALARIO”.

|  |
| --- |
| TABLA EMPLEADO |
| ID\_EMPLEADO |
| NOMBRE |
| APELLIDO |
| PUESTO |

|  |
| --- |
| EMPLEADO\_DETALLE |
| ID\_EMPLEADO |
| IS |
| SALARIO |
| FECHA\_INGRESO |
| DEPARTAMENTO |

1. Realizar una consulta de las tablas TABLA\_EMPLEADO y EMPLEADO\_DETALLE mostrar el “NOMBRE” y “APELLIDO” del empleado que tengan el máximo de los salarios y que en puesto tenga el valor la constante “ADM”.
2. Efectuar una consulta de las tablas TABLA\_EMPLEADO y EMPLEADO\_DETALLE que muestre el nombre y apellido de los empleados que tengan el máximo salario y que su puesto corresponda a ‘ADM’
3. Efectuar una consulta de las tablas TABLA\_EMPLEADO y EMPLEADO\_DETALLE que calcule el número de dias del empleado de nombre ”pepe” ha permanecido en la empresa y mostrar su nombre y el número de días.
4. De la tabla EMPLEADO\_DETALLE, calcular la nomina total de todos los empleados

10



**Ejercicios de SQL – Parte 2.** Realizar las consultas requeridas a las tablas T.Empleados, T.Deptos y T.Sedes

*Tiempo estimado resolver para los ejercicios: 30 minutos.*

T.Empleados

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NUEMP** | **NOMBRE** | **DEPTO** | **PUESTO** | **TELEFONO** | **EXT** |  | **AÑOS** | **SALARIO** | **COMISION** |
| 10 | JORGE | 20 | GERENTE | 35453534 | 123 |  | 7 | 18375.5 | - |
| 20 | JOSE | 20 | VENDEDOR | 34534567 | 123 |  | 8 | 13240.4 | 612.5 |
| 30 | GUSTAVO | 38 | GERENTE | 34534678 | 124 |  | 5 | 17500 | - |
| 40 | ENRIQUE | 38 | VENDEDOR | 34533124 | 124 |  | 6 | 18000.6 | 856 |
| 50 | MARCO | 15 | GERENTE | 34538767 | 125 |  | 10 | 20659 | - |
| 60 | ISRAEL | 38 | VENDEDOR | 34539898 | 126 |  | - | 16808.2 | 650.25 |
| 70 | GERMAN | 15 | VENDEDOR | 34555687 | 127 |  | 7 | 16502.5 | 1115.2 |
| 80 | ALBERTO | 20 | EMPLEADO | 34234433 | 128 |  | - | 13504 | 128 |
| 90 | SAMUEL | 42 | VENDEDOR | 34588998 | 129 |  | 6 | 18200 | 1386 |
| 100 | RICARDO | 42 | GERENTE | 34534566 | 122 |  | 7 | 18352.34 | - |
| 110 | JUAN | 15 | EMPLEADO | 34538941 | 135 |  | 5 | 12502.6 | 506 |
| 120 | MIGUEL | 38 | EMPLEADO | 34535644 | 111 | - |  | 12954 | 180 |

T.Deptos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NUMDEPTO** | **NOMDEPTO** | **GERENTE** | **DIVISION** | **LOCALIZACION** |
| 10 | CENTRAL | 160 | CORPORATIVO | D.F. |
| 15 | CULIACAN | 50 | NORTE | SINALOA |
| 20 | NUEVO LEON | 10 | NORTE | MONTERREY |
| 38 | CHIHUAHUA | 30 | NORTE | CHIHUAHUA |
| 42 | TLAXCALA | 100 | CENTRO | TLAXCALA |
| 51 | MORELIA | 140 | CENTRO | MICHUACAN |
| 66 | TUXTLA | 270 | SUR | CHIAPAS |
| 84 | QUINTANA ROO | 290 | SUR | QUINTANA ROO |

T.Sedes

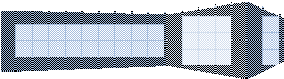
|  |  |  |  |
| --- | --- | --- | --- |
| **LOCALIZACION** | **ESTADO** | **PAIS** | **REGION GLOBAL** |
| D.F. | DF | MX | AMERICA |
| SINALOA | CULIACAN | MX | AMERICA |
| MONTERREY | NUEVO LEON | MX | AMERICA |
| CHIHUAHUA | CHIHUAHUA | MX | AMERICA |
| TLAXCALA | TLAXCALA | MX | AMERICA |
| MICHUACAN | MORELIA | MX | AMERICA |
| CHIAPAS | TUXTLA | MX | AMERICA |
| QUINTANA ROO | QUINTANA ROO | MX | AMERICA |
| PHOENIX | ARIZONA | EU | AMERICA |
| BARCELONA | BARCELONA | ES | EUROPA |

11



1. Obtener los totales de sueldos que tiene cada nombre de departamento
2. Obtener los totales de sueldos que tiene cada departamento del país MX
3. Obtener los nombres de los trabajadores que tienen de gerente a “Roberto”.
4. Obtener Nombre, Nombre de departamento, Gerente, Estado y País de los trabajadores de la división SUR.
5. Obtener Total de sueldos por división en las divisiones centro y sur
6. Obtener los nombres de los integrantes del departamento que tiene a la persona con el sueldo más bajo.
7. Obtener los totales de sueldos que tiene cada País
8. Obtener los nombres empleados a que estado, país, nombre de departamento y gerente que pertenecen, los cuales tengan comisión menor o igual .11

12



## Respuestas a Ejercicios de SQL – Parte 1.

1. Insert into producto (id\_producto, nombre, precio, id\_fabrica) values (1342, ‘impresora laser’, 1250, 4);
2. Delete \* from producto where precio between 5 and 10;
3. Update producto set nombre = ‘scanner’ where id\_fabrica = 4;
4. Select nombre, precio from producto order by nombre asc, precio desc;
5. Select avg (precio) as promedio from producto where nombre like ‘%impresora%’;
6. Select \* from empleado inner join empleado\_detalle on empleado.id\_empleado = empleado\_detalle.id\_empleado;
7. Select empleado.nombre, empleado.apellido max (empleado\_detalle.salaro) from empleado inner join empleado\_detalle on empleado.id\_empleado = empleado\_detalle. Id\_empleado where empleado.puesto = ‘adm’;
8. Select emplado.nombre, empleado.apellido max(empleado\_detalle.salario) from emplado inner join empleado\_detalle on empleado.id\_empleado=empleado\_detalle.id\_empleado

where empleado.puesto=’adm’

1. Select nombre, datediff(“y”,fecha\_ingreso,now()) from empleado, empleado\_detalle

where nombre=”pepe”

1. Select sum(empleado\_detalle.salario) as total from empleado\_detalle

13



## Respuestas a Ejercicios de SQL – Parte 2.

* 1. Select b.nomdepto, sum (a.sueldo) from empleado a, depto b

where b.numdepto = a.numdepto Group by nomdepto

* 1. Select b.numdepto, sum (a.sueldo) from empleado a, deptos b, sedes c

where b.numdepto = a.numdepto and c.localización = b.localización and c.pais = ‘MX’

Group by b.numdepto

* 1. Select a.nombre

from empleado a, depto b

where a.depto = (select d.numdepto from empleado c, deptos d where c.nombre = “roberto” and d.gerente = c.numempleado )

* 1. Select a.nombre, b.nomdepto, b.gerente, c.estado, c.pais from empleado a, depto b

where b.division = ‘SUR’ and c.localizacion= b.localización and a.numdepto = b.numdepto

* 1. Select b.division, sum(a.sueldo) from empleado a, depto b

where b.division in ( ‘CENTRO’, ´SUR´) and a.numdepto = b.numdepto Group by b.division

* 1. Select b.nombre from empleados b

where b.depto = (select a.depto from empleados a Group by a.depto having min(a.sueldo)

* 1. Select b.pais, sum(a.sueldo) from empleado a, depto b, pais c

where b.localizacion = c.localizacion and a.numdepto = b.numdepto Group by c.pais

.

* 1. Select a.nombre, b.nomdepto, c.gerente, c.estado, c.pais

from empleado a, depto b 14

where a.comision < =.11 and c.localizacion= b.localización and b.numdepto =

a.numdepto



# Sección 3. Lógica

## Objetivo.

Esta sección tiene la finalidad practicar el uso de operadores lógicos, permitiendo fortalecer habilidades para la abstracción de problemas.

## Recomendaciones.

1. Lee detalladamente la descripción de los ejercicios.
2. Toma en cuenta el uso adecuado de las tablas de verdad para cada operador.

|  |  |  |  |
| --- | --- | --- | --- |
| **Conjunción** | | | |
| V | & | V | **V** |
| F | & | V | **F** |
| V | & | F | **F** |
| F | & | F | **F** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Disyunción** | | | |
| V | || | V | **V** |
| F | || | V | **V** |
| V | || | F | **V** |
| F | || | F | **F** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Implicación** | | | |
| V |  | V | **V** |
| F |  | V | **V** |
| V |  | F | **F** |
| F |  | F | **V** |

15



## Ejercicios de Lógica.

*Tiempo estimado para los ejercicios: 20 minutos.*

1. **De acuerdo a los valores de A y B determina el resultado de las siguientes expresiones:** A = falso B = verdadero

1. [(A || B) && (B && A)]

2. [(A && A) || (B || B)]

3. [(A && A) && (B || B)]

1. **De acuerdo a los valores de A,B y C identifica cuál de las siguientes expresiones es verdadera:** A=falso B= verdadero C= falso

a) ((AB) && (CA) ) && (BC)

b) ((B || C)(A && B)) || (B || C)

## ¿Cuáles de las siguientes sentencias son siempre falsas, de acuerdo a los siguientes valores? A= 7 B=10 C=1 D=8

a) (B>C) && (C<A) && (B>A) && (C>A)

b) (C<A) && (C>B) || (B<A) && (C<B)

c) (D>A) || (C>B) && (C<B) || (B<D)

d) (C>A) && (D>B) || (C>A) || (B>D)

## ¿Cuál es el resultado de la siguiente sentencia, Verdadera o Falsa?

a) (FV) && (V && V) || (V(F || F && V || F) )

## ¿Cuál de las siguientes sentencias es verdadera? tomando en cuenta los siguientes valores: A=-1, B=6,C=2,D=1

a) (D>A)||(C>B)&&(B<A)&&(D>C)

b) (D<C)&&(A<D)||(C<B)&&(D<A)

c) (B>C)||(A<D)&&(A>D)||(C>B)

## Si tienes hambre, entonces comes. Si comes, entonces te lavaste las manos, por lo tanto, si tienes hambre te lavaste las manos.

* 1. Verdadero
  2. Falso

16



## Respuestas a ejercicios de Lógica.

1. Respuesta 1: Falso Respuesta 2: Verdadero Respuesta 3: Falso
2. b
3. a,b
4. Verdadera
5. b
6. a

17



# Sección 4. Java

## Objetivo.

Esta sección tiene la finalidad de reforzar los conocimientos del lenguaje Java y Javascript.

## Recomendaciones.

1. Leer las siguientes Ligas:
   * Modificadores de Campos
     + [https://docs.oracle.com/javase/tutorial/reflect/member/fieldModifiers.html](https://docs.oracle.com/javase/tutorial/reflect/member/fieldModifiers.html%20)
   * Modificadores de metodos
     + https://docs.oracle.com/javase/tutorial/reflect/member/methodModifiers.html
   * Modificadores de clase
     + <https://docs.oracle.com/javase/specs/jls/se8/html/jls-8.html#jls-8.1.1>
   * Modificadores de acceso.
     + <http://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html>
   * Clases anidadas (incluye clases internas).
     + [http://docs.oracle.com//javase/tutorial/java/javaOO/nested.html](http://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html)
   * Colecciones-Listas
     + <http://docs.oracle.com/javase/tutorial/collections/interfaces>
   * Hilos
     + <http://docs.oracle.com/javase/tutorial/essential/concurrency>
   * Paquetes
     + <http://docs.oracle.com/javase/specs/jls/se8/html/jls-7.html>
2. Lee detalladamente el código de los ejercicios.
3. Verificar que la sintaxis y la semántica del código sea correcta.
4. Contesta las preguntas



## Ejercicios de Java.

*Tiempo estimado para los ejercicios: 25 minutos.*

## ¿Cuál es el resultado que se obtiene al ejecutar el siguiente código?:

import java.util.\*;

public class EjemArrayList{

public static void main (String[] args){ ArrayList nombres = new ArrayList();

System.out.println("Elementos al inicio: " + nombres.size()); nombres.add("Ana");

nombres.add("Bea"); nombres.add("Dalia"); nombres.add("Pedro"); nombres.add("Bea");

System.out.println("Contenido: " + nombres); System.out.println("Elementos: " + nombres.size()); System.out.println(nombres.contains("Ana")); nombres.remove("Ana"); System.out.println("Contenido: " + nombres); System.out.println("Elementos: " + nombres.size()); System.out.println(nombres.contains("Ana")); nombres.add(3, "Ana"); System.out.println("Contenido: " + nombres);

System.out.println("SubLista(1,4):" + nombres.subList(1,4));

}

}

## ¿Cuál es el resultado que se obtiene al ejecutar el siguiente código?:

public class Saludo extends Hola{ Saludo(String frase){

super();

System.out.println("Hola Javier");

}

public static void main(String argv[]){ Saludo s = new Saludo("palabradoce");

}

}

public class Hola{

private Hola(String mensaje){ System.out.println("HolaJuan");

}

Hola(){}

}

19



## ¿Cuál es el resultado que se obtiene al ejecutar el siguiente código?:

public class EjemploOperador { static int *a* = 4;

static int *b* = 2;

public static void main(String args[]){ if( *suma*(*a*,*b*) || *resta*(*a*,*b*) ){

System.*out*.println("Entra con--- suma(a,b) || resta(a,b) 1");

}

if( *resta*(*a*,*b*) || *suma*(*a*,*b*) ){

System.*out*.println("entra con--- resta(a,b) || suma(a,b) 2");

}

if( *suma*(*a*,*b*) && *resta*(*a*,*b*) ){

System.*out*.println("entra con --- suma(a,b) && resta(a,b) 3");

}

if( *resta*(*a*,*b*) && *suma*(*a*,*b*) ){

System.*out*.println("entra con --- resta(a,b) && suma(a,b) 4 ");

}

if( *suma*(*a*,*b*) | *resta*(*a*,*b*) ){

System.*out*.println("Entra con--- suma(a,b) | resta(a,b) 5");

}

if( *resta*(*a*,*b*) | *suma*(*a*,*b*) ){

System.*out*.println("entra con--- resta(a,b) | suma(a,b) 6");

}

if( *suma*(*a*,*b*) & *resta*(*a*,*b*) ){

System.*out*.println("entra con --- suma(a,b) & resta(a,b) 7");

}

if( *resta*(*a*,*b*) & *suma*(*a*,*b*) ){

System.*out*.println("entra con --- resta(a,b) & suma(a,b) 8 ");

}

}

public static boolean suma( int a, int b){ System.*err*.println("Suma true"); int r = a + b;

System.*out*.println(" la suma es : " + a + b + " r: " + r ); return true;

}

public static boolean resta(int a, int b){ System.*err*.println("resta false"); int r;

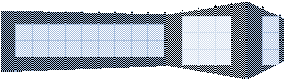
r = a-b;

System.*out*.println(" la resta es : " + r ); return false;

}

}

20



## ¿Cuál es el resultado que se obtiene al ejecutar el siguiente código?:

public interface FiguraGeometrica{ public int calculaArea(int b, int h); public void imprimeNombre();

}

Public class Cuadrado implements FiguraGeometrica{ Public void imprimeNombre(){

System.out.println(“ cuadrado”);

}

}

## ¿Cuál es el resultado que se obtiene al ejecutar el siguiente código?:

public class FactrialNum {

public static void main(String[] args) { int fac = 1;

int n = 5;

for(int i = 1; i <= n; i++){ fac \*= i ;

}

System.*out*.print(“Resultado: ”+fac);

}

}

## ¿Cuál es el resultado que se obtiene al ejecutar el siguiente código?:

public class Constructores{

static{ System.out.println(“Estático”); } public Constructores{

System.out.println(“Constructor”); } public static void main(String args[]) {

Constructores c = new Constructores(); }

}

21



## Cuando compila un programa escrito en el lenguaje de programación Java el compilador convierte el código fuente en código independiente de la plataforma que una máquina virtual Java pueda entender. ¿Cómo se denomina este código independiente de la plataforma?

## ¿Qué es lo primero que debería comprobar si ve el siguiente error en tiempo de ejecución?

Exception in thread "main" java.lang.NoClassDefFoundError: HelloWorldApp.java.

## Al declarar el método *main*, ¿qué modificador debe indicarse primero, public o static?

## ¿Qué parámetros define el método *main*?

## ¿Qué es un servidor Web?

* 1. Es una máquina que esta prendida todo el tiempo.
  2. Es un servicio que implementa protocolos.
  3. Es un servicio de FTP para publicar archivos en Internet.
  4. Ninguna de la anteriores

## ¿Qué es un cliente HTTP?

* 1. Una persona que compra productos en una tienda online.
  2. Un programa que se propaga como un virus.
  3. Es lo mismo que un servidor Web.
  4. Un programa que se conecta o solicita información al servidor, normalmente mediante un navegador Web.

## ¿Qué se utiliza para maquetar una página Web?

* 1. Ruby on Rails.
  2. XHTML y CSS.
  3. MySQL, Oracle, SQLServer.
  4. PHP, Java, Ruby.

22



## Algunas librerías o frameworks de JavaScript son:

* 1. Spring, Hibernate.
  2. Cake PHP
  3. jQuery, Mootools, Ext JS.
  4. Ajax

## ¿Es posible programar orientado a objetos en JavaScript?

* 1. Si
  2. No

## ¿Qué significa JSON?

* 1. Java Server Object Notation
  2. Java Script Object Notation
  3. John Stuart Ole Nat
  4. Ninguna de los anteriores

## Si ejecutas el siguiente código, ¿cuál será el mayor y menor tiempo de inserción?

* 1. El menor tiempo de inserción es en LinkedHashSet y mayor en HashSet
  2. El menor tiempo de inserción es en HashSet y mayor en TreeSet
  3. El menor tiempo de inserción es en HashSet y mayor en LinkedHashSet
  4. El menor tiempo de inserción es en LinkedHashSet y mayor en TreeSet

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | final Set<Integer> hashSet = new HashSet<Integer>(1\_000\_000);  final Long startHashSetTime = System.currentTimeMillis();  for (int i = 0; i < 1\_000\_000; i++) {      hashSet.add(i);  }  final Long endHashSetTime = System.currentTimeMillis();  System.out.println("Time spent by HashSet: " + (endHashSetTime - startHashSetTime));    final Set<Integer> treeSet = new TreeSet<Integer>();  final Long startTreeSetTime = System.currentTimeMillis();  for (int i = 0; i < 1\_000\_000; i++) {      treeSet.add(i);  }  final Long endTreeSetTime = System.currentTimeMillis();  System.out.println("Time spent by TreeSet: " + (endTreeSetTime - startTreeSetTime));    final Set<Integer> linkedHashSet = new LinkedHashSet<Integer>(1\_000\_000);  final Long startLinkedHashSetTime = System.currentTimeMillis();  for (int i = 0; i < 1\_000\_000; i++) {      linkedHashSet.add(i);  }  final Long endLinkedHashSetTime = System.currentTimeMillis();  System.out.println("Time spent by LinkedHashSet: " + (endLinkedHashSetTime - startLinkedHashSetTime)); |

## Este tipo de colección define una sucesión de elementos, sí admite duplicados. A parte de los métodos heredados de Collection, añade métodos que permiten mejorar: Acceso posicional a elementos, búsqueda de elementos, iteración sobre elementos, rango de operación.

* 1. MAP
  2. SET
  3. LIST
  4. STREAM API

## ¿La siguiente afirmación es verdadera o falsa? Los hilos se pueden crear utilizando los mecanismos: - Extender la clase Thread, - Implementar la interfaz Runnable.

## Las clases anidadas permiten declarar una clase dentro de otra y así reducir el número de clases y ficheros del proyecto, la clase anidada puede declararse como static, en este caso la clase anidada se denomina anidada estática (Static Nested) en caso contrario ¿como se denomina?

1. Anidada no estática (Non static nested)
2. Local (local classes)
3. Anónima (anonymous classes)
4. Interna (inner classes)

**Si se tiene el siguiente código:**

package packlibros;

class Libros {

private String titulo;

private String autor;

private int anio;

Libros(String t, String a, int d){

titulo=t;

autor=a;

anio=d;

}

void mostrar(){

System.out.println(titulo);

System.out.println(autor);

System.out.println(anio);

System.out.println();

}

}

package packlibros;

class DemoLibro {

public static void main(String[] args) {

Libros libros[]=new Libros[5];

libros[0]= new Libros("Aprendiendo a Programar en Java","Sonia Jaramillo",2015 );

libros[1]= new Libros("Estructuras de Datos en Java","Mark Allen Weis",2013 );

libros[2]= new Libros("Fundamentos de programación Java","Ricardo Marcelo Villalobos",2012);

libros[3]= new Libros("Introducción a la Programación Orientada A Objetos","Franciso Aragón Mesa",2014 );

libros[4]= new Libros("Java Cómo Programar","Paul Deitel",2017);

for (int i=0; i<libros.length;i++) libros[i].mostrar();

}

}

## ¿Cuál sería la salida?

## Una vez que DemoLibro es parte del paquete packlibros, ¿se puede utilizar la línea de comando “java DemoLibro”?

## 

23



## Respuestas a ejercicios de Java.

1. La salida del código es:

Elementos al inicio: 0

Contenido: [Ana, Bea, Dalia, Pedro, Bea]

Elementos: 5 true

Contenido: [Bea, Dalia, Pedro, Bea]

Elementos: 4 false

Contenido: [Bea, Dalia, Pedro, Ana, Bea] SubLista(1,4):[Dalia, Pedro, Ana]

Explicación: Cuando se trabaja con acumuladores se tomar en cuenta los cambios que se realicen son permanentes y no existe un tipo de recuperación de la información.

1. La salida del código es: Hola Javier

Explicación: La palabra reservada “*super*” hace que sea ejecutado el constructor de la superclase.

24



1. La salida del código es:

Suma true resta false Suma true Suma true resta false resta false Suma true resta false resta false Suma true Suma true resta false resta false Suma true

la suma es : 42 r: 6

Entra con--- suma(a,b) || resta(a,b) 1

la resta es : 2

la suma es : 42 r: 6

entra con--- resta(a,b) || suma(a,b) 2

la suma es : 42 r: 6 la resta es : 2

la resta es : 2

la suma es : 42 r: 6 la resta es : 2

Entra con--- suma(a,b) | resta(a,b) 5

la resta es : 2

la suma es : 42 r: 6

entra con--- resta(a,b) | suma(a,b) 6

la suma es : 42 r: 6 la resta es : 2

la resta es : 2

la suma es : 42 r: 6

25



1. Error al compilar

Explicación: Una clase debe implementar todos los métodos de la interfaz sino al compilar marcara un error. En este caso solo se implementa un método por lo tanto existe el error. Las cabeceras de los métodos (identificador y número y tipo de parámetros) deben aparecer en la clase tal y como aparecen en la interfaz implementada.

1. La salida del código es: Resultado: 120

Explicación: Multiplica Fac por i y lo asigna a Fac, otra forma de hacerlo es: fac = fac \* i ; El resultado es: 120, la función principal de este código es sacar el Factorial de 5.

1. La salida del código es:

Estático Constructor

Explicación: Al instanciar el objeto de una clase primero se ejecuta el bloque estático y después el constructor del objeto.

1. Bytecode
2. Comprobar el classpath. No se puede encontrar su clase.
3. Pueden estar en cualquier orden, pero el convenio es public static.

26



1. El método main define un solo parámetro, normalmente denominado args, cuyo tipo es un array de objetos String.
2. Ninguna de las anteriores
3. Un programa que se conecta o solicita información al servidor, normalmente mediante un navegador Web.
4. XHTML y CSS.
5. jQuery, Mootools, Ext JS.
6. No
7. Java Script Object Notation.
8. Respuesta correcta B) El tiempo de inserción es menor en HashSet y mayor en TreeSet.
9. Respuesta correcta C) LIST
10. Verdadera
11. Respuesta correcta D) Inner classes.

Aprendiendo a Programar en Java

Sonia Jaramillo

2015

Estructuras de Datos en Java

Mark Allen Weis

2013

……..

1. No se puede utilizar por sí sola porque es parte del paquete.