Examen final programacion orientada a objetos

Composición y estructura del examen

|  |  |
| --- | --- |
| **Componente Teórico** | **Componente práctico** |
| Programación estructurada | Ejercicio práctico programación estructurada |
| Programación orientada a objetos | Ejercicio práctico POO |
| Lenguaje de consulta estructurada | Ejercicio práctico SQL |

**Componente Teórico**: Esta sección del examen se centra en evaluar la comprensión teórica del estudiante en conceptos fundamentales de la programación. Es una parte esencial que sienta las bases para las secciones prácticas subsiguientes.

**Componente Práctico**: En esta sección, el examen pasa a evaluar las habilidades prácticas en varios dominios, subdivididos en las siguientes áreas:

**a. Programación Estructurada:**

Teoría: Aquí, se evaluará la comprensión del estudiante en la programación estructurada, que incluye el manejo de estructuras de control, funciones, y más.

Ejercicio Práctico: En este segmento, se requerirá que el estudiante demuestre su habilidad en la programación estructurada a través de un ejercicio práctico.

**b. Programación Orientada a Objetos (POO):**

Teoría: Esta parte aborda conceptos clave en la programación orientada a objetos, como la herencia, el encapsulamiento, la polimorfia, etc.

Ejercicio Práctico POO: La evaluación práctica en POO requerirá que los estudiantes desarrollen un ejercicio que reflejen su comprensión y habilidad en esta metodología de programación.

**c. Lenguaje de Consulta Estructurada (SQL):**

Teoría: Aquí se evaluará el conocimiento en SQL, incluyendo la creación, manipulación y consulta de bases de datos.

Ejercicio Práctico SQL: Este segmento exigirá que el estudiante demuestre su capacidad para aplicar su conocimiento teórico en SQL a través de un ejercicio práctico.

***NOTA PARA LA ENTREGA DEL CÓDIGO FUENTE SE DEBE ADJUNTAR EL ENLACE AL REPOSITORIO DE GIT Y HACER COMMIT DE CADA PUNTO RESUELTO***

Programación Estructurada

**¿Qué estructura permite repetir un bloque de código mientras se cumpla una condición?**

A) If

B) Switch

C) While

D) Goto

**¿Cuál es el resultado de este fragmento de código? int x = 5; int y = x++;**

A) x = 6, y = 6

B) x = 5, y = 5

C) x = 6, y = 5

D) Error de compilación

**¿Qué técnica de programación estructurada permite tomar decisiones múltiples basadas en una variable o expresión?**

A) For loop

B) While loop

C) Goto

D) Switch statement

**En programación estructurada, ¿qué herramienta permite evitar código duplicado?**

A) Herencia

B) Funciones

C) Clases

D) Concatenación

**¿Cuál es la salida del siguiente código en C? int x = 10; if(x > 5) printf("Yes"); else printf("No");**

A) Yes

B) No

C) 10

D) Ninguna de las anteriores

**En programación estructurada, ¿qué concepto no se enfoca en la separación de datos y funciones?**

A) Encapsulamiento

B) Modulación

C) Cohesión

D) Ninguna de las anteriores

**¿Qué ciclo permite ejecutar un bloque de código un número determinado de veces?**

A) Switch

B) If

C) For loop

D) While loop

**¿Qué declaración se utiliza para saltar al próximo ciclo en una iteración?**

A) Continue

B) Break

C) Return

D) Goto

**¿Cuál de las siguientes no es una estructura de control de flujo en programación estructurada?**

A) If-else

B) For loop

C) Class

D) Switch statement

**¿Cómo se llama la técnica que permite ejecutar un bloque de código hasta que una condición sea falsa en C?**

A) Do-while loop

B) For loop

C) Goto

D) Switch statement

**¿Qué operador se usa para comparar si dos valores son iguales en C#?**

A) =

B) :=

C) ==

D) !=

**En programación estructurada, ¿qué se utiliza para agrupar un conjunto de declaraciones bajo una condición única?**

A) Switch statement

B) If statement

C) Loop

D) Function

**¿Qué herramienta permite la ejecución condicional de segmentos de código en la programación estructurada?**

A) Looping

B) Functions

C) Conditional Statements

D) Encapsulation

**¿Cuál es la salida del siguiente fragmento de código en C#?**



A) x = 6, y = 15

B) x = 6, y = 16

C) x = 5, y = 15

D) Error de compilación

**¿Qué valor se asigna a z en el siguiente código en C#?**



A) 5

B) 6

C) 7

D) Ninguna de las anteriores

**¿Cuál es la salida del siguiente fragmento de código en C#?**



A) 1 3 5

B) 1 3

C) 0 2 4

D) Ninguna de las anteriores

**Dado el siguiente código en C#, ¿cuál es la salida?**



A) Yes

B) No

C) Error de compilación

D) Ninguna de las anteriores

**¿Cuál es el resultado de este fragmento de código en C#?**



A) 11

B) 12

C) 13

D) 14

Programación orientada a objetos

**¿Cuál de los siguientes NO es un principio de diseño en POO?**

A) Abstracción

B) Herencia

C) Polimorfismo

D) Normalización

**¿Qué concepto de POO permite ocultar los detalles de implementación de una clase y mostrar solo la funcionalidad al usuario?**

A) Herencia

B) Abstracción

C) Polimorfismo

D) Encapsulamiento

**¿Cómo se llama la técnica que permite que un objeto pueda tomar muchas formas?**

A) Polimorfismo

B) Herencia

C) Encapsulamiento

D) Abstracción

**En C#, ¿cuál es la palabra clave utilizada para heredar las propiedades de una clase?**

A) inherit

B) extends

C) include

D) “:”

**¿Qué tipo de método en POO se llama automáticamente cuando se crea un objeto?**

A) Destructor

B) Constructor

C) Static method

D) Virtual method

**¿Qué principio de POO asegura que los datos internos de un objeto están protegidos de ser accedidos directamente desde fuera?**

A) Herencia

B) Polimorfismo

C) Abstracción

D) Encapsulamiento

**Dado el siguiente código en C#, ¿cuál es la salida?**



A) Animal Sound

B) Woof!

C) Error de compilación

D) Ninguna de las anteriores

**¿Qué resultado produce el siguiente fragmento de código en C#?**



A) 25

B) 10

C) Error de compilación

D) 5

**¿Cuál es la salida del siguiente código en C#?**



A) Class A

B) Class B

C) Error de compilación

D) Ninguna de las anteriores

**Dado el siguiente código en C#, ¿cuál es el valor de result?**



A) 7

B) 10

C) 12

D) 24

**En C#, ¿cuál es la salida del siguiente fragmento de código?**



A) Base

B) Derived

C) Error de compilación

D) Ninguna de las anteriores

Lenguaje de consulta estructurada

**¿Qué tipo de clave en una base de datos se utiliza para establecer una relación con otra tabla?**

A) Clave primaria

B) Clave foránea

C) Clave única

D) Clave compuesta

**¿Qué comando SQL se utiliza para eliminar una tabla de la base de datos?**

A) DELETE

B) DROP

C) REMOVE

D) TRUNCATE

**¿Qué cláusula SQL se usa para filtrar registros y mostrar solo aquellos que cumplen una condición específica?**

A) SELECT

B) WHERE

C) LIKE

D) JOIN

**En una relación de muchos a muchos en una base de datos, ¿qué se utiliza comúnmente para gestionar esta relación?**

A) Clave primaria

B) Clave compuesta

C) Clave foránea

D) Tabla de unión

**¿Cuál es el propósito del comando SQL "ALTER TABLE"?**

A) Eliminar una tabla

B) Modificar una tabla existente

C) Crear una nueva tabla

D) Seleccionar datos de una tabla

**¿Qué cláusula SQL se utiliza para ordenar los resultados de una consulta?**

A) SORT

B) ORDER BY

C) GROUP BY

D) RANK

**¿Qué comando SQL se utiliza para insertar nuevos registros en una tabla?**

A) INSERT INTO

B) ADD

C) UPDATE

D) CREATE

**En el diseño de una base de datos, ¿cómo se llama la regla que garantiza que los datos se encuentren de manera consistente en toda la base de datos?**

A) Integridad referencial

B) Normalización

C) Indexación

D) Herencia

**¿Qué comando SQL modifica los datos existentes en una tabla?**

A) MODIFY

B) CHANGE

C) UPDATE

D) ALTER

**¿Cuál es el comando SQL para crear una nueva base de datos?**

A) CREATE DATABASE

B) NEW DATABASE

C) ADD DATABASE

D) BUILD DATABASE

**¿Qué cláusula se utiliza en SQL para combinar filas de dos o más tablas basándose en una columna relacionada entre ellas?**

A) MERGE

B) COMBINE

C) JOIN

D) LINK

**¿Qué tipo de clave asegura que cada valor en la columna sea único?**

A) Clave primaria

B) Clave única

C) Clave compuesta

D) Clave foránea

**En SQL, si queremos seleccionar todos los campos de una tabla, ¿qué símbolo se utiliza?**

A) &

B) %

C) \*

D) #

Práctica programación orientada a objetos

**Título: Sistema de Gestión de una Librería Pequeña**

**Enunciado del problema:**

La pequeña librería "ReadersCove" necesita un sistema de gestión para llevar el control de su inventario y ventas. Debes desarrollar un programa orientado a objetos que utilice los conceptos de encapsulamiento y abstracción. A continuación, se detallan los objetos involucrados y los cálculos específicos a implementar:

**Objetos involucrados:**

Libro: Representa cada uno de los libros disponibles en "ReadersCove". Cada libro tiene un título, ISBN, autor, precio unitario y cantidad en stock. Debe ser posible agregar nuevos libros al inventario, actualizar sus detalles y consultar la información de un libro específico.

Venta: Esta clase representa una venta realizada en "ReadersCove". Debe contener información sobre los libros comprados, la cantidad de cada libro y el monto total de la venta. Además, se debe mantener un registro de todas las ventas realizadas en la librería.

Cálculos específicos a implementar:

Cálculo del precio total de la venta: El precio total de la venta debe calcularse sumando el precio de cada libro en la venta.

Actualización de inventario: Después de cada venta, el sistema debe actualizar el inventario disminuyendo la cantidad de libros vendidos del stock disponible.

Tu tarea es diseñar e implementar el sistema de gestión de "ReadersCove" utilizando los conceptos de la programación orientada a objetos mencionados anteriormente. Este sistema más simple ayudará a la librería a mantener un control adecuado de su inventario y ventas, sin requerir detalles adicionales como descuentos o el manejo de empleados y clientes.

Práctica SQL

**Problema Base de datos**

La empresa ElectroTech se dedica a la venta y mantenimiento de equipos electrónicos. Te han contratado para diseñar y crear la base de datos para su sistema de gestión de clientes y productos.

A continuación, se describen los elementos que debe contener la base de datos:

**Tabla "Clientes":**

Campos: idCliente (clave primaria), nombre, dirección, teléfono, correo electrónico.

**Tabla "Categorías":**

Campos: idCategoria (clave primaria), nombreCategoria.

**Tabla "Productos":**

Campos: idProducto (clave primaria), nombreProducto, descripción, precioUnitario, idCategoria (clave foránea a la tabla Categorías).

**Tabla "Ventas":**

Campos: idVenta (clave primaria), fechaVenta, idCliente (clave foránea a la tabla Clientes).

**Tabla "DetallesVentas":**

Campos: idDetalleVenta (clave primaria), idVenta (clave foránea a la tabla Ventas), idProducto (clave foránea a la tabla Productos), cantidad, subtotal.

Consideraciones adicionales:

Cada cliente puede realizar varias compras (registros en la tabla Ventas), y cada venta puede tener varios detalles de venta (registros en la tabla DetallesVentas) que corresponden a los productos adquiridos en esa venta.

La tabla Categorías almacena las diferentes categorías de productos disponibles en la tienda.

La tabla Productos contiene información sobre los productos disponibles para la venta y utiliza una clave foránea para asociar cada producto a una categoría específica.

La tabla Ventas registra la información básica de cada venta, como la fecha y el cliente que realizó la compra.

La tabla DetallesVentas almacena los detalles específicos de cada venta, incluyendo el producto vendido, la cantidad comprada y el subtotal calculado a partir del precio unitario del producto y la cantidad.

Tu tarea es diseñar el esquema de la base de datos mencionada, incluyendo las tablas y las relaciones entre ellas. Además, deberás definir adecuadamente las claves primarias y foráneas, así como especificar los tipos de datos adecuados para cada campo. ¡Éxito en tu examen de base de datos en SQL Server!

**GITHUB:**

***NOTA PARA LA ENTREGA DEL CÓDIGO FUENTE SE DEBE ADJUNTAR EL ENLACE AL REPOSITORIO DE GIT Y HACER COMMIT DE CADA PUNTO RESUELTO***