Buenas estimado profesor, compañeros y compañeras,

Adjunto mi aporte en relación a la pregunta generadora #2 sobre estructuras de selección en programación.

Las estructuras de selección son fundamentales en el desarrollo de aplicaciones, ya que nos permiten tomar decisiones basadas en diferentes condiciones. A continuación, presento tres ejemplos prácticos que ilustran el uso de estructuras de selección simple y múltiple en situaciones numéricas.

1. Selección Simple - Cálculo de Descuento en una Compra:

Supongamos que estás desarrollando un programa para una tienda en línea y deseas calcular el descuento aplicable a una compra según el monto total. Aquí, solo necesitas evaluar una condición para determinar si el descuento es aplicable.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "spanish");

double monto\_total = 0.0;

double descuento = 0.0;

cout << "Ingrese el monto total de la compra: ";

cin >> monto\_total;

if (monto\_total > 1000) {

descuento = monto\_total \* 0.1; // 10% de descuento si el monto supera los 1000

} else {

descuento = 0; // No hay descuento

}

cout << "El descuento aplicable es: $" << descuento << endl;

return 0;

}

1. Selección Múltiple - Clasificación de Números según su Rango:

Imagina que estás desarrollando un programa que clasifica los números en diferentes rangos según su valor. Aquí, evaluamos múltiples condiciones para determinar en qué rango se encuentra un número dado.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "spanish");

int numero = 0;

cout << "Ingrese un número entero: ";

cin >> numero;

if (numero < 0) {

cout << "El número está en el rango negativo." << endl;

} else if (numero >= 0 && numero < 10) {

cout << "El número está en el rango de 0 a 9." << endl;

} else if (numero >= 10 && numero < 100) {

cout << "El número está en el rango de 10 a 99." << endl;

} else {

cout << "El número es igual o mayor que 100." << endl;

}

return 0;

}

1. Selección Simple - Determinar si un Número es Par o Impar:

En este ejemplo, utilizamos una estructura de selección simple para determinar si un número ingresado por el usuario es par o impar.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "spanish");

int numero = 0;

cout << "Ingrese un número entero: ";

cin >> numero;

if (numero % 2 == 0) {

cout << "El número es par." << endl;

} else {

cout << "El número es impar." << endl;

}

return 0;

}

Referencia: Deitel, P. J., & Deitel, H. M. (2021). Cómo Programar en C++. Pearson Education - UNED. México.

Dia 3

[SHARON DAYANA ROSALES OBANDO](https://aprende.uned.ac.cr/user/view.php?id=7472&course=193)

Hola estimada compañera, en la explicación sobre las condiciones completas y el ejemplo del ecosistema fueron muy claros y útiles. Sin embargo, permíteme darte una sugerencia para hacer el código más fácil de entender.

Podrías simplificar el código y hacerlo más legible. Además, podrías proporcionar otro ejemplo utilizando valores numéricos para ilustrar mejor cómo funcionan las condiciones completas.

Aquí te dejo una idea del código con algunos cambios:

#include <iostream>

#include <string>

using namespace std;

int main() {

string animal;

int tamano = 0;

int color = 0;

int comportamiento = 0;

cout << "Ingrese el animal que ha observado (1 para ardilla, 2 para oso): ";

cin >> animal;

if (animal == "1") {

cout << "Ingrese el tamaño del animal (1 para pequeño, 2 para mediano, 3 para grande): ";

cin >> tamano;

cout << "Ingrese el color del animal (1 para marrón, 2 para negro): ";

cin >> color;

cout << "Ingrese el comportamiento del animal (1 para juguetón, 2 para solitario): ";

cin >> comportamiento;

if (tamano == 1 && color == 1 && comportamiento == 1) {

cout << "Es una ardilla común." << endl;

} else {

cout << "No coincide con ninguna especie conocida." << endl;

}

} else if (animal == "2") {

cout << "Ingrese el tamaño del animal (1 para pequeño, 2 para mediano, 3 para grande): ";

cin >> tamano;

cout << "Ingrese el color del animal (1 para marrón, 2 para negro): ";

cin >> color;

cout << "Ingrese el comportamiento del animal (1 para juguetón, 2 para solitario): ";

cin >> comportamiento;

if (tamano == 3 && (color == 1 || color == 2) && comportamiento == 2) {

cout << "Es un oso pardo." << endl;

} else {

cout << "No coincide con ninguna especie conocida." << endl;

}

} else {

cout << "No se reconoce el animal ingresado." << endl;

}

return 0;

}

En esta nueva versión, se utiliza valores numéricos para representar el tamaño, el color y el comportamiento del animal, lo que hace que el código sea más claro y fácil de entender. Espero que te sea útil la sugerencia

Buenas estimado profesor, compañeros y compañeras,

Quería compartir con ustedes algunos ejemplos prácticos sobre el uso de operadores lógicos en programación. Como sabemos, los operadores lógicos son fundamentales en el desarrollo de aplicaciones, ya que nos permiten combinar condiciones y tomar decisiones más complejas basadas en múltiples criterios.

Según lo que he aprendido, estos operadores son como "bloques de construcción" que nos ayudan a formar condiciones simples y complejas en nuestros programas. Con ellos, podemos controlar el flujo de nuestro código de manera más eficiente y precisa.

Espero que encuentren útiles estos ejemplos y que les ayuden a comprender mejor el uso de los operadores lógicos en la programación.

1. **Control de Acceso a una Zona Restringida:** estamos controlando el acceso a una zona restringida y queremos permitir la entrada solo a las personas que no están en una lista de no permitidos y tienen más de 18 años.

#include <iostream>

#include <string>

using namespace std;

int main() {

setlocale(LC\_ALL, "spanish");

string nombre;

int edad = 0;

bool no\_en\_lista\_no\_permitidos = false;

bool mayor\_de\_edad = false;

cout << "Ingrese su nombre: ";

cin >> nombre;

cout << "Ingrese su edad: ";

cin >> edad;

cout << "¿Está en la lista de no permitidos? (1 para Sí, 0 para No): ";

cin >> no\_en\_lista\_no\_permitidos;

mayor\_de\_edad = (edad >= 18);

if (!no\_en\_lista\_no\_permitidos && mayor\_de\_edad) {

cout << "Bienvenido/a, " << nombre << ". Puede ingresar a la zona restringida." << endl;

} else {

cout << "Lo siento, " << nombre << ". No tiene permitido ingresar a la zona restringida." << endl;

}

return 0;

}

1. **Descuento Especial para Clientes Nuevos:** queremos ofrecer un descuento especial a los clientes que no han realizado al menos una compra anteriormente o tienen un ingreso mensual menor o igual a $1000.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "spanish");

bool cliente\_nuevo = false;

double ingreso\_mensual = 0.0;

bool no\_tiene\_ingreso\_suficiente = false;

cout << "¿Es usted un cliente nuevo? (1 para Sí, 0 para No): ";

cin >> cliente\_nuevo;

cout << "Ingrese su ingreso mensual: $";

cin >> ingreso\_mensual;

no\_tiene\_ingreso\_suficiente = !(ingreso\_mensual > 1000);

if (cliente\_nuevo || no\_tiene\_ingreso\_suficiente) {

cout << "¡Felicidades! Usted tiene derecho a un descuento especial." << endl;

} else {

cout << "Gracias por su interés, pero no cumple con los requisitos para el descuento." << endl;

}

return 0;

}

1. **Aprobación de Préstamo Hipotecario:** queremos determinar si un cliente es elegible para un préstamo hipotecario basándonos en su edad, capacidad de pago y si ha tenido algún incumplimiento financiero previo.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "spanish");

int edad = 0;

double ingreso\_mensual = 0.0;

bool tiene\_incumplimientos = false;

bool no\_es\_elegible = false;

cout << "Ingrese su edad: ";

cin >> edad;

cout << "Ingrese su ingreso mensual: $";

cin >> ingreso\_mensual;

cout << "¿Ha tenido algún incumplimiento financiero previo? (1 para Sí, 0 para No): ";

cin >> tiene\_incumplimientos;

no\_es\_elegible = !(edad >= 25 && ingreso\_mensual > 2000 && !tiene\_incumplimientos);

if (no\_es\_elegible) {

cout << "Lo sentimos, no cumple con los requisitos para un préstamo hipotecario." << endl;

} else {

cout << "Felicidades, usted es elegible para un préstamo hipotecario." << endl;

}

return 0;

}

*Referencia: Deitel, P. J., & Deitel, H. M. (2021). Cómo Programar en C++. Pearson Education - UNED. México.*