

UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS

Materia: Computación tolerante a fallas

Clave: I7036

Sección: D06



NRC: 179961

Actividad 2

Código estudiante: 221350567

Alumno: Rauf Alfonso Hamden Estrada

Carrera: Ingeniería en Computación

Fecha: 27/08/2023

Docente: Michel Emanuel Lopez Franco

2023B

Introducción

La programación es una disciplina que implica la creación y el mantenimiento de software, y a medida que los programas se vuelven más complejos, la gestión de errores y la validación de datos se vuelven esenciales para garantizar que haya menos fallos en las aplicaciones o softwares. Se utilizan diversas herramientas y técnicas para abordar estos desafíos.

Los errores en la programación son muy comunes y por lo cual solemos recurrir a herramientas para poder detectarlos. Por ello se utilizan estructuras como el "try...catch" en Java, que permite capturar y manejar excepciones de manera controlada, evitando que el programa se detenga abruptamente.

Herramientas para el Manejo de Errores

Bibliotecas de Validación de Datos

Puedes utilizar bibliotecas como validators en Python para validar datos de entrada y detectar errores. Esto es útil en funciones que toman datos de usuario, es una herramienta útil para realizar una variedad de validaciones de datos en Python

Ejemplo 1 (validación de correo):

```
from validators import email

email_address = "pesopluma@hotmail.com"

if email(email_address):
    print("La direccion de correo electronico es valida.")
else:
    print("La direccion de correo electronico no es valida.")
```

Ejemplo 2(validation de rango de numero):

```
from validators import between

number = 42

if between(number, min=0, max=100):
    print("El número está dentro del rango válido.")
else:
    print("El número está fuera del rango válido.")
```

Try catch en java

El bloque try...catch en Java es una estructura fundamental para manejar excepciones, que son situaciones excepcionales o errores que pueden ocurrir durante la ejecución de un programa. El propósito del try...catch es permitirte escribir código que pueda lanzar excepciones y proporcionar un mecanismo para manejar esas excepciones de manera controlada, evitando que el programa se detenga de forma abrupta.

Try en inglés es el verbo intentar, así que todo el código que vaya dentro de esta sentencia será el código sobre el que se intentará capturar el error si se produce y una vez capturado hacer algo con él.

Un ejemplo sería:

```
try {  
    System.out.println("bloque de código donde pudiera saltar un error es este");  
}
```

Cuando se produce un error o excepción en el código que se encuentra dentro de un bloque try, pasamos directamente a ejecutar el conjunto de sentencias que tengamos en el bloque catch.

```
catch (Exception e) {  
    System.out.println("bloque de código donde se trata el problema");  
}
```

Validación de errores

Se refiere al proceso de verificar si los datos ingresados o procesados en un programa cumplen con ciertos criterios o restricciones antes de ser utilizados o procesados adicionalmente. El objetivo principal de la validación de datos es garantizar la integridad y la corrección de los datos, así como evitar errores y comportamientos inesperados en el programa.

Ejemplo de validación de números en c++:

```
#include <iostream>  
#include <cstdlib>  
using namespace std;  
  
bool validar(char *num) {  
    // Verifica si el caracter esta en el rango de caracteres numericos en la tabla ASCII  
    if ((char)*num >= 48 && (char)*num <= 57) {  
        system("cls"); // Limpia la pantalla de la consola  
        return true; // El caracter es un numero valido  
    }  
}
```

```

    } else {
        cout << "\nError. Ingresa un valor valido: "; // Mensaje de error
        return false; // El caracter que se ingreso
    }
}

char valorChar[100], *ptrChar;
bool Validar_Numeros(int *dato);
int *ptr = NULL;
int main(int argc, char **argv) {
    cout << "Escribe un numero=" << endl;
    do {
        cin >> valorChar;
        cin.ignore(); // Limpia el bufer del teclado
        ptrChar = &valorChar[0]; // Asigna la direccion del primer caracter al puntero
    } while (!validar(ptrChar));

    return 0;
}

```

Conclusiones

El manejo de errores en programación es esencial para garantizar aplicaciones de alta calidad y confiabilidad. Las herramientas para la detección de errores como el try catch es uno de los mas utilizados y más eficaz, sin embargo , con las bibliotecas de python es aun mas sencillo ya que Python es un lenguaje que tiene muchas librerías que nos podrían ayudar a detectar errores y la validación de errores es algo que yo suelo utilizar para evitar que el programa cierre cuando el usuario ponga un carácter incorrecto. La elección de la herramienta adecuada depende de las necesidades específicas del proyecto y del lenguaje de programación utilizado.

Enlaces

TylerMSFT. (2023, 3 abril). Procedimientos recomendados de C++ moderno para el control de errores y excepciones. Microsoft Learn. <https://learn.microsoft.com/es-es/cpp/cpp/errors-and-exception-handling-modern-cpp?view=msvc-170>

Kantor, I. (s. f.). Manejo de errores, «Try. . .catch». <https://es.javascript.info/try-catch>

Errores y excepciones. (s. f.). Python documentation. <https://docs.python.org/es/3/tutorial/errors.html>

Corredera, P. Á. (2022, 27 diciembre). 5 herramientas para verificar tu código en Python. CIBERNINJAS. <https://ciberninjas.com/python-5-herramientas-limpiar-codigo/>