University of Applied Sciences
HOCHSCHULE EMDEN·LEER

John Ousterhout:A Philosophy of Software Design

Error of Existence

Adarsh Pal

3. Mai 2023

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Introduction

It is a software design emphasizes the importance of designing systems in a way that minimizes the likelihood of errors occurring in the first place. This approach focuses on preventing errors from happening rather than simply detecting and handling them after they occur.

- The goal is to create software systems that are more reliable, robust, and easier to maintain.
- Errors in software can have significant negative impacts, such as crashes, data loss, or security vulnerabilities.
- In most programming languages,it is coded in try-catch blocks.(Exception Handling)

Here is the example in Python

```
try:
num1 = int(input(Ënter a number: "))
num2 = int(input(Ënter another number: "))
result = num1 / num2
print(The result is:", result)
except ValueError:
print(Ìnvalid input, please enter a number.")
except ZeroDivisionError:
print(Cannot divide by zero.")
except:
print(Än unknown error occurred.")
```

- The code inside the try block accepts two input values from the user and calculates the result by dividing the first number with the second number.
- If there is a ValueError or ZeroDivisionError, the program jumps to the corresponding except block to handle the error. If an unknown error occurs,the final except block will catch it.
- If the user enters a non-numeric value, the program will jump to the ValueError except block and print the message Invalid input.
- Similarly, if the user enters a 0 for the second number, the program will jump to the ZeroDivisionError except block and print Cannot divide by zero."

Exception Add Complexity

Many programming language include a formal exception mechanism that allows exceptions to be thrown by the lower level code and caught by enclosing code.However,when the new mechanism is added it must me integrated with the existence mechanism or else it will change the flow of program and add more complexity to it.

Several different ways a particular piece of code encounter exceptions:

- A caller may provide bad argument or configuration information.
- The code may detect bugs, internal inconsistencies or situations is not prepared to handle.
- In distributed system, network packets may be lost or delayed, servers may not respond in timely fashion.

Too Many Exceptions

- Most Programmer are taught that it's important to detected and reports error; they often interpret this to mean the more errors detected, the better."
- Excessive use of exception can be a problematic, it make code hard to understand. If it is used too many then, it become too difficult to predict its behavior and identify the root cause of error.
- Exception should be used only in case when it is necessary such as dealing with unpredictable error that can be easily handle in normal control flow of program.
- To overcome from this situation is very easy just reduce the number of places were the exception have to be handled.

```java
try {
    // some code
} catch (IOException e) {
    // handle IOException
} catch (SQLException e) {
    // handle SQLException
} catch (NumberFormatException e) {
    // handle NumberFormatException
} catch (NullPointerException e) {
    // handle NullPointerException
} catch (ArrayIndexOutOfBoundsException e) {
    // handle ArrayIndexOutOfBoundsException
} catch (ClassNotFoundException e) {
    // handle ClassNotFoundException
} catch (Exception e) {
    // handle all other exceptions
```

Abbildung:

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Mask Exceptions

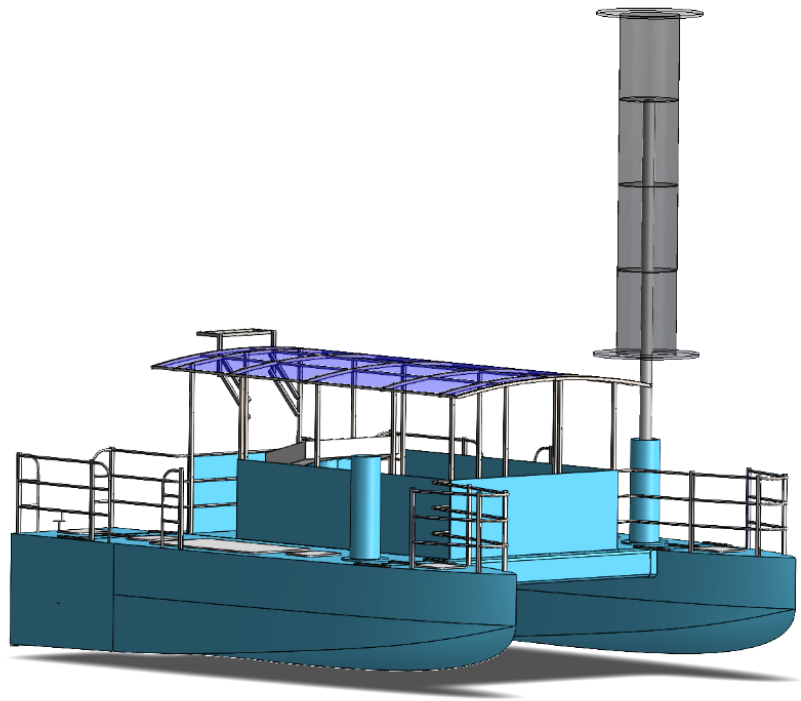
University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Aggregation Exceptions

- Exception aggregation is the process of collecting multiple related exceptions and consolidating them into a single exception object. This can be useful in situations where multiple errors occur within a system or application, and it is desirable to present the user with a single error message or log a single exception for easier troubleshooting.
- It can also be useful in distributed systems, where errors may occur in different components of the system and need to be consolidated into a single error message or exception for easier management and debugging.
- It should not be used to mask or hide underlying errors. Rather, it should be used as a tool for better presentation and management of errors and exceptions.

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Water Taxi I



University of Applied Sciences
HOCHSCHULE EMDEN·LEER

3D-Printer I

3D-Printer I

Der Datenstruktur `MyStructure` aus der Datei `MyStructure.py` ist sehr interessant.

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Code

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models

MODEL = models.Sequential()
```

Code

```
# Author: Ardit Sulce, Automate Everything with Python,
# Course URL: https://www.udemy.com/course/automate-ever

import tabula

table = tabula.read_pdf('weather.pdf', pages=1)

print(type(table[0]))

table[0].to_csv('output.csv', index=None)
```

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Vielen Dank

für Ihre Aufmerksamkeit

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Quellen

Quellen I

Nachfolgend werden die Quellen der Bilder angegeben, die für diese Präsentation in ihrer ursprünglichen Form oder modifiziert verwendet worden sind.