PRINCIPLE OF LEAST ASTONISHMENT

@ferosekhanj

WHAT IS "PRINCIPLE OF LEAST ASTONISHMENT"?

"A component of a system should behave in a way that most users will expect it to behave. The behavior should not astonish or surprise users."

WHY THIS PRINCIPLE IS IMPORTANT?

- People have something called natural reflex.
- An action that is performed without conscious thought as a response to a stimulus.
- And we would like to go with that flow.

PUSH

LETS LOOK AT A REAL WORLD EXAMPLE

You see this door and want to open it. You try to pull the door using the handle. It doesn't open. There is a moment of surprise, then you read the sign and then push the door. Because pulling the door is your natural reflex and is driven from your spinal cord(muscle memory). Reading the sign and then acting based on that is done by your rational brain. When natural reflex fail there is a surprise.

PUSH
HERE
TO
OPEN

APPLYING THE PRINCIPLE HERE...

Don't provide a handle if one has to push open a door. And provide a handle only if the door is to be pulled. Your reflex works and there is no surprise.

WHY THIS PRINCIPLE APPLIES TO SOFTWARE DESIGN?

- Like reflex developer(or people in common) have something called intuition.

- The ability to understand something instinctively, without the need for conscious reasoning.

- This will speed up the developer.

LETS LOOK AT SOME SOFTWARE EXAMPLE

- JavaScript
- C#
- Java

FOLLOW UNIVERSAL CONCEPTS

Javascript

var d = new Date(2021,1,31)

Output:

Wed Mar 02 2021 00:00:00 GMT+0530 (India Standard Time)

- What is happening here?
  - JavaScript date has month represented as 0-11
  - If the date value is wrong then it automatically adjusts date on its own

ENSURE COMPLETE IMPLEMENTATIONS

```csharp
C#
IList<int> l = new int[]{1,2,3};
l.Add(1);


Output:

Throws NotSupportedException
```
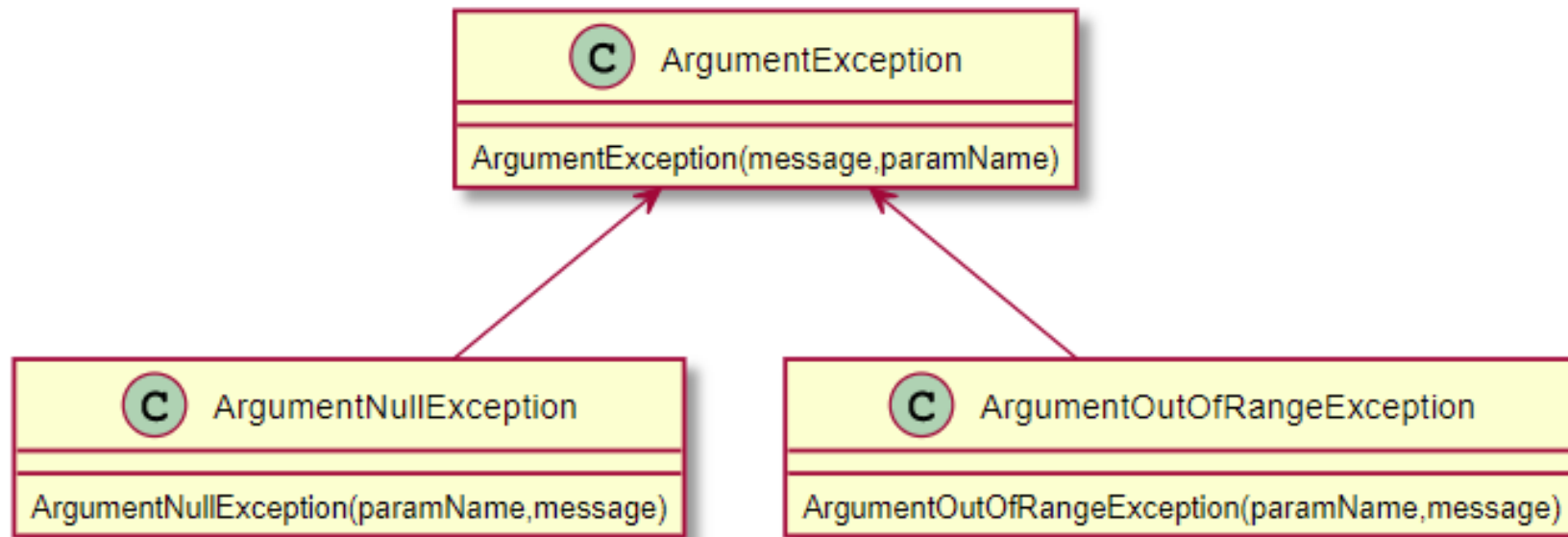
- What is happening here?
  - Array is a fixed collection in .net
  - Someone decided to implement the IList<> interface
  - When you try to use the object as IList<> the implementor denies the contract

CONSISTENT PARAMETER ORDER

The order of parameters in base class and derived class is not consistent here.

CONSISTENT PARAMETER ORDER...

BE CAREFUL WITH NAMES

```csharp
C#
var now = DateTime.Now;
Console.WriteLine(now);
now.AddDays(1);
Console.WriteLine(now);

Output:

27-07-2021 14:52:23
27-07-2021 14:52:23
```

- DateTime is a struct in .net and immutable
- The function AddDays will return a new DateTime.
- But from the name one would expect the original object to be modified.

BE CAREFUL WITH NAMES

```
C#
var utc = DateTime.UtcNow;

var timeSpan = DateTime.Now - utc;

Console.WriteLine(timeSpan.Seconds);

Output:

0
```

- Timespan has Seconds property that returns the seconds part of full timespan
- TotalSeconds returns the timespan in seconds.

BE CAREFUL WITH NAMES

```java
Java
Thread thread = new Thread(){
    public void run(){
      System.out.println(Thread.currentThread().getId());
    }
}
System.out.println(Thread.currentThread().getId());
thread.run();
Output:
1
1
```

- Thread has a method run()
- If you create an anonymous thread overriding run()
- Users often call run() from the current thread.
- But actually Thread in java has a method start() to create a new thread.

PERSPECTIVE IS IMPORTANT

The perspective from which you design the API names and the user's perspective might differ. Before implementing your code, try your own code from the user's perspective.

THANKS

"Design should not dominate things. Not dominate people. It should help people." —Dieter Rams