****

**TEAM MEMBERS:**

Caetano Flores - Pamela Chipe - Anthony Morales - Jeffrey Manobanda - Leonel Tipán - Ruben Benavides

**EXCEPTIONS**

**Definitions:**

* **Errors:** This type of errors occurs mostly due to the environment in which the code is being executed, and they cannot be handled during runtime, resulting in the program always terminating. Some types of these errors are:
* **AWTError**: This error occurs when an exception related to the Abstract Window Toolkit (AWT), which is a graphical library in Java, occurs.

o **ThreadDeath**: This error occurs when a thread abruptly stops due to a call to the stop() method on a Thread object. This error is generally not caught or directly handled.

o **VirtualMachineError**: This error occurs when an exception related to the Java Virtual Machine (JVM) occurs. Examples of this type of error are StackOverflowError and OutOfMemoryError.

**Exceptions:** Exceptions are the result of abnormal conditions during program runtime and can be of different types. Some examples are:

* **RuntimeException**: This exception indicates errors that occur during program runtime and are usually caused by logical errors in the code. Examples of this type of exception are NullPointerException and IndexOutOfBoundsException.
  + Arithmetic exception
  + NullPointerException
  + IndexOutOfBoudsException
  + NumberFormatException
* **IOException**: This exception occurs when an error occurs during input or output operations, such as reading or writing files.
* **ClassNotFoundException**: This exception occurs when attempting to load a class using Class.forName() and the specified class cannot be found at runtime.
* **Custom Exceptions:** Custom exceptions are classes created to represent specific error conditions within an application. These classes extend the Exception class or one of its subclasses and are used to capture and handle specific exceptional situations in the code.

**Exception Handling:**

Exceptions, unlike errors, can be handled during runtime. For this, there are several methods:

* **Try-Catch**: The try-catch block is used to surround a block of code where an exception is expected to occur. If an exception occurs within the try block, it is caught and handled in the corresponding catch block. This prevents the exception from interrupting the program's runtime.
  + **The finally block:** it is an optional block used to specify a block of code that will always execute, whether an exception occurs or not. It is used to perform cleanup or resource release actions, regardless of whether an exception occurred or not.
* **Throw**: The throw keyword is used to manually throw an exception at a specific point in the code. It allows the programmer to generate a custom exception or throw an existing exception to indicate an exceptional condition.
* **Other methods:** In addition to try-catch and throw, there are other methods for handling exceptions, such as using nested try-catch blocks, using the throws keyword to propagate exceptions to higher-level methods, and using try-with-resources blocks to automatically handle resource release.

Thanks to these methods, all exceptions can be handled as long as they are implemented correctly, allowing us to control the flow of runtime and handle exceptional situations properly.

**OBJECTS: Reading and writing**

**Definition:**

They are defined by their properties and behavior. Properties are the characteristics of an object, such as its size, color, and shape called attributes on programmation. Behavior or methods are what an object can do, such as move, speak, think, or like calculate an addition. They consist on three things:

* Name: This is a variable name that represents the object.
* Member data: The data that describes the object
* Member methods: Behavior that describes the object.

**Reading and writing**

**Serialization:** It is the conversion of an object to a series of bytes , so that the object can be easily saved to persistent storage or streamed even across other platforms or networks.

**Read and write:**

The ObjectOutputStream is used to serialize it and write to a file

To write to an object's property, you use the dot notation and the assignment operator (=). For example, to set the name property of myObject to the value "John Doe", you would use the following code

modelCar.name = "Bugatti Chiron"

**BIBLIOGRAPHY:**

***Excepciones en Java - try/catch*. (n.d.). https://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?codigo=161**

**Pankaj. (2022). Java ClassNotFoundException - java.lang.ClassNotFoundException. *DigitalOcean*. https://www.digitalocean.com/community/tutorials/java-classnotfoundexception-java-lang-classnotfoundexception**

**Vasquez Gomez, A. P. (n.d.). *tema-5-poo-excepciones*. Studocu. https://www.studocu.com/es-mx/document/instituto-tecnologico-de-toluca/programacion-orientada-a-objetos/tema-5-poo-excepciones/25403848**