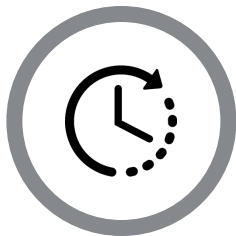


Lenguajes y Paradigmas de Programación

Sugerencias en el desarrollo de la clase



RESPECTO POR EL TIEMPO
DE INICIO Y TÉRMINO



ALCEN LA MANO
PARA HABLAR



TRABAJEN SOLO
CON LO DE LA CLASE



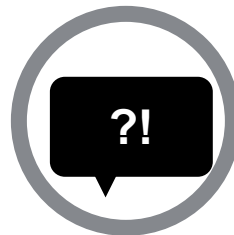
EL FEEDBACK ES
BIENVENIDO



PARTICIPEN EN
LA CLASE



USE CONSCIENTEMENTE



EXPRESEN SUS
CONSULTAS E IDEAS
LIBREMENTE



DEJA TU PUESTO
LIMPIO

Paradigma Orientado a Objetos

Agenda

- Lectura/Escritura de Variables
- Colecciones
- Excepciones
- Ejercicio 1 – Captura de Errores – Java
- Ejercicio 2 – Captura de Errores – Python
- Ejercicio 3 – Lanzamiento y Captura de Errores - Java
- Ejercicio 4 – Lanzamiento y Captura de Errores – Python
- Ejercicio 5 – Pirámide - Python
- ¿Que hemos visto?

Lectura/Escritura de Variables

¿Recuerdan que la idea es dejar las variables de clases como protected?

Entonces, ¿Cómo accedo al valor o lo modifico si es necesario?

Getter en Java

Si la clase tiene una variable llamada nombre de tipo String, protegida, y deseo obtener su valor de c

Tipo de Dato

Método comienza con get

```
public String getNombre() {  
    return this.nombre;  
}
```

Nombre de la variable

Setter en Java

Si la clase tiene una variable llamada nombre de tipo String, protegida, y deseo modificar su valor de esta manera:

Método comienza con set

Tipo de Dato

```
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}
```

Nombre de la variable

Python

Lo anterior es usado mayoritariamente en el mundo Java, pero en Python no, dado que las variables siguen siendo públicas → Solución Python: “replicar” su uso directo

@property y @.setter

```
class Persona(object):  
    def __init__(self, edad):  
        self.__edad = edad
```

```
@property  
def edad(self):  
    return self.__edad
```

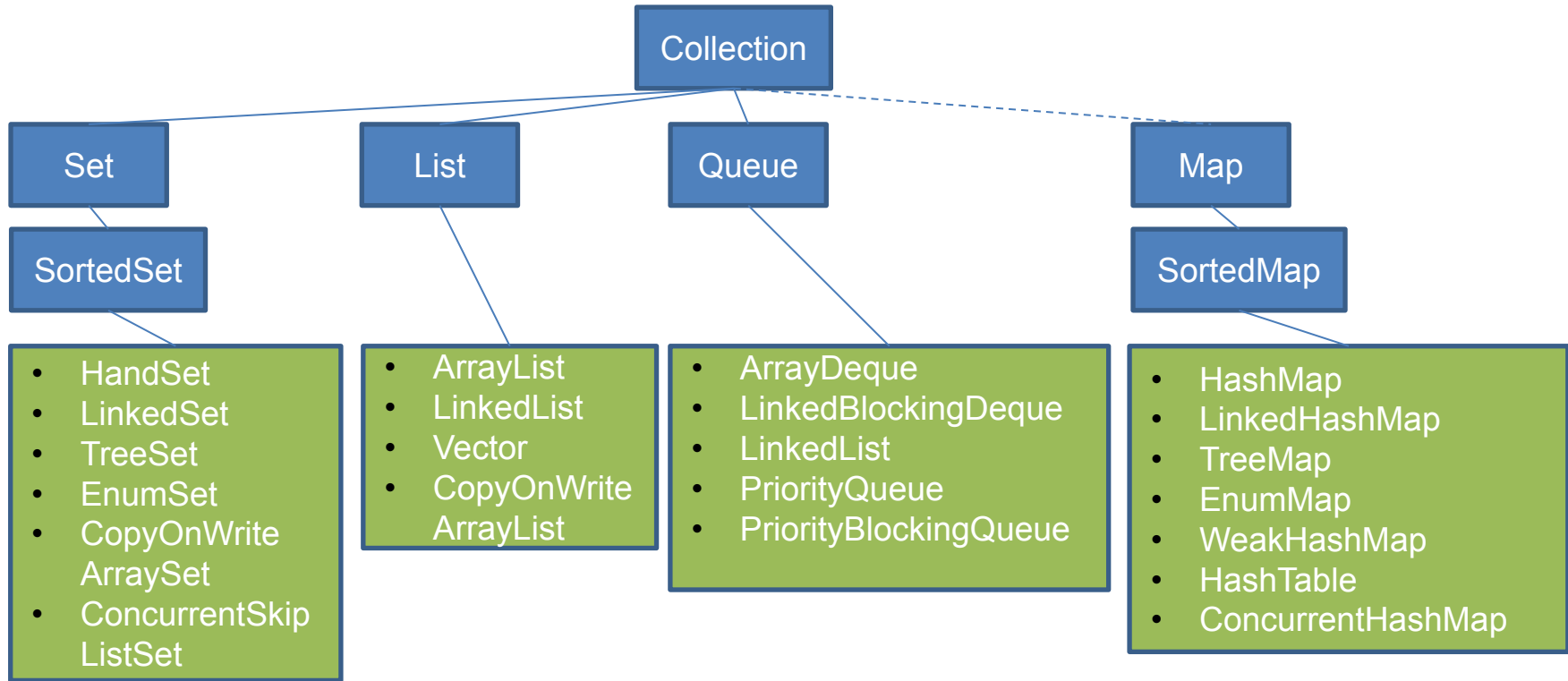
```
@edad.setter  
def edad(self, edad):  
    if edad < 0:  
        raise Exception('edad no válida')  
    if edad > 120:  
        raise ValueError('edad no válida')  
    self.__edad = edad  
  
if __name__ == '__main__':  
    print(Persona(23).edad)
```

¡El orden es importante!

Colecciones

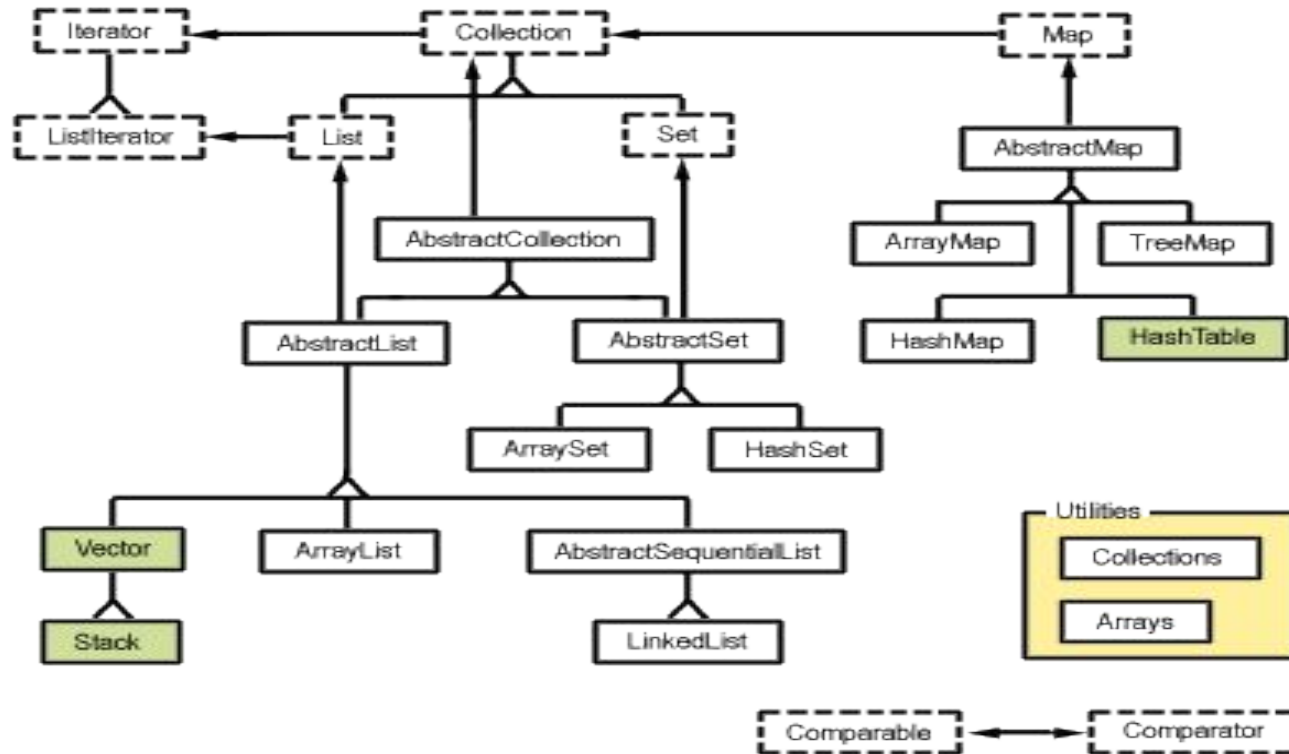
- Almacén de datos dinámico
- Pueden cambiar de tamaño (insertar y eliminar elementos)
- Se pueden ordenar
- Se puede acceder a elementos específicos o subconjunto de la colección
- Se instancian con su interfase (estructura de datos) y haciendo referencia a la clase que la implementa

Colecciones Java



Fuente: <https://www.youtube.com/watch?v=bTu-fz1JmWQ>

Colecciones Python



Fuente: <http://dis.um.es/~bmoros/Tutorial/parte4/cap4-7.html>

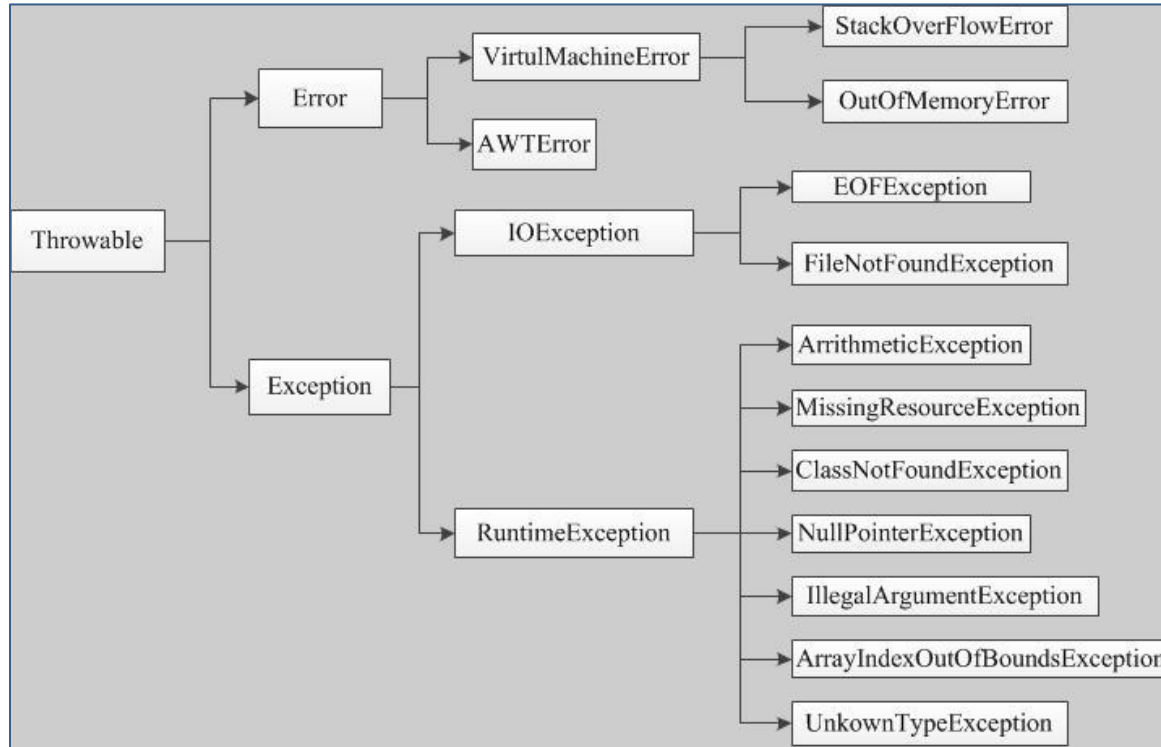
Ejemplos de Implementaciones

- Ejemplo List - Python:
<https://replit.com/@GustavoJara/clase0901PythonList#main.py>
- Ejemplo Slicing - Python:
<https://replit.com/@GustavoJara/clase0902Slicing#main.py>
- Ejemplo List - Java:
<https://replit.com/@GustavoJara/clase0903JavaList#Main.java>
- Uso de List e Iterator para lectura de archivo - Java:
<https://replit.com/@GustavoJara/clase0801lecturacsv#Main.java>

Excepciones

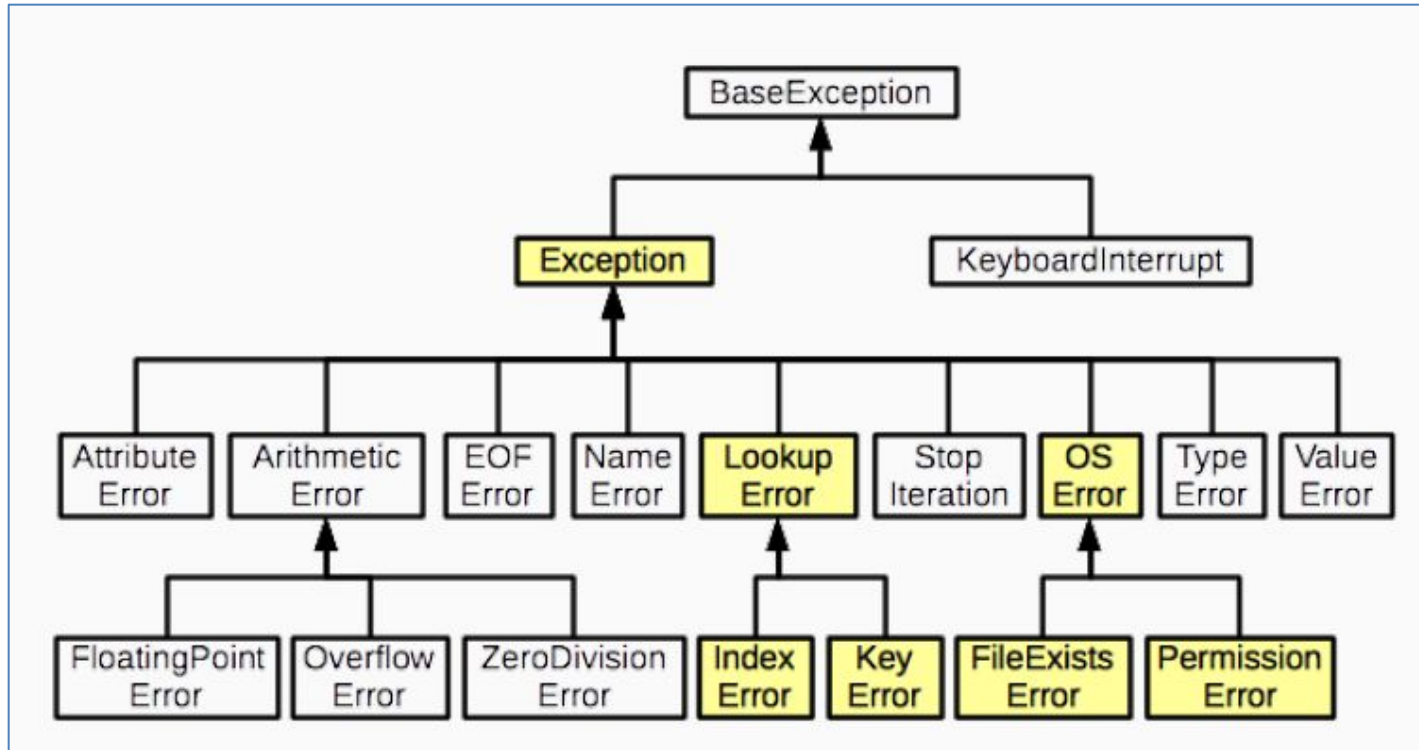
- Una excepción se produce cuando algo falla durante la ejecución de un programa. Ejemplos:
 - Hacer referencia a un objeto que no existe
 - Hacer referencia a un elemento de un arreglo más allá de su dimensión
 - Usar divisor cero
- Las excepciones se pueden generar. Habitualmente para validar datos
- Las excepciones se pueden capturar para que el programa no sufra una caída. Ejemplos:
 - Si un archivo no existe, puedo usar otro archivo de backup
 - Si se ingresa una edad negativa, puedo advertir al usuario
- Las excepciones están organizadas en jerarquías

Jerarquía de Excepciones en Java



Fuente: <https://programmerclick.com/article/67581901058/>

Jerarquía de Excepciones en Python



Ver anatomía de excepciones en: <https://edube.org/learn/.../la-anatom-iacute-a-de-las-excepciones>

Ejercicio 1 – Captura Excepciones - Java

- Captura de error aritmético (división por cero)
- Captura de error por índice fuera de rango
- Captura de error por intento de apertura de archivo que inexistente

Ejercicio 2 – Captura Excepciones - Python

- Captura de error por conversión de datos
- Captura de error aritmético (división por cero)
- Captura de error inesperado

Ejercicio 3

Lanzamiento y Captura Excepciones - Java

- Lanzar excepción por validación de datos
- Capturar la excepción para advertir al usuario
- Tomar como base:

<https://replit.com/@GustavoJara/clase0703paqvehiculocamioneta#Main.java>

- Modificar constructor para que use setters
- Modificar setters para que lance excepción limitando largos de Strings
- Solicitar ingreso de marca, modelo y velocidad máxima
- Instanciar clase Vehículo capturando errores

Ejercicio 4

Lanzamiento y Captura Excepciones - Python

- Lanzar excepción por validación de datos
- Capturar la excepción para advertir al usuario
- Tomar como base:
<https://replit.com/@GustavoJara/clase070402visvehiculocamioneta#main.py>
- Usar mangling + decoradores @property y @setter para todas las propiedades
- Modificar constructor para que use setters
- Modificar setters para que lance excepción limitando largos de Strings
- Solicitar ingreso de marca, modelo y velocidad máxima
- Instanciar clase Vehículo capturando errores

Ejercicio 5 - Pirámide - Python

- Crear la clase **Pirámide**
- Crear las propiedades **bloque** (string) y **pisos** (int) usando mangling + decoradores @property y @setter
- Lanzar excepción (Raise) desde el método setter si bloque es de largo distinto a 1
- Lanzar excepción (Raise) desde el método setter si pisos es menor o igual que cero o mayor que 20
- Utilizar setter desde el constructor de la clase para activar las validaciones de los métodos setter
- Crear el método **dibujar pirámide**
- Instanciar la clase, realizando la captura de las excepciones (Try..Except) y activar el método para dibujar la pirámide, si esta no se puede dibujar por error en las validaciones se debe advertir al usuario

Ejemplo: Para la pirámide de bloque tipo # y de 4 pisos, al activar el método debe imprimir:

```
#  
##  
###  
####
```

Que hemos visto ?

- Colecciones, sus jerarquías y usos. Se dejan fuentes para ampliar conocimientos
- Ejemplos de aplicación de List en Java y Python
- Ejemplo de slicing en Python
- Excepciones, sus jerarquías y usos. Se dejan fuentes para ampliar conocimientos
- Ejercicios de lanzamiento y captura de excepciones en en Java y Python

¿Preguntas?