has / does

Exception Methods

○ object

Exception Object
- name ▭
- desc ▭
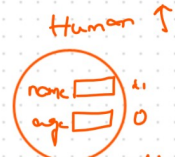- StackTrace ▭
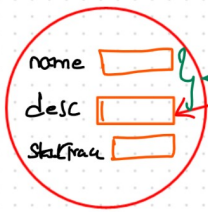
getMessage ( )

✓ toString( )

printStackTrace ( )   All ←

try {   class
        ↗
} catch (Exception e) {
    { S.o.P (e)   e.toString()

    e.getMessage( )

Human ↑

none ▭ ''
age ▭ 0

Human h = new Human ( )

S.o.P(h); ——

h.toString()

Chr man @ address

---

getMessage: this method returns String type of data which contains the description of the exception

It is inherited from throwable class

toString( ): this method returns String data which contains the name as well as description of exception

It is inherited from Object class & overiden in throwable class & again it is herited from throwable class

ⓒ Object → toString()
        ↓
ⓒ throwable → toString()

Exception ⓒ ←

printStackTrace: this method prints the complete description of the exception object such as name, description & StackTrace

Car ↑
↑ relat
Audi C

Audi q
Err c = new Audi ( )
= new Audi ( )

?? ce

parent ← child type
type   ↓ returns
    loose coupling

Explicitly Creating the Exception Object
— throw

__Notes__   (refer program : ThrowingExceptionExample.)

→ "If we want to create a exception explicitly, we can create it by creating the object of that particular Exception class

→ After creating the exception object, we need to throw the exception to the JVM

→ It can be done with the help of __"throw keyword"__