

# Exceptions

→ try must be followed by catch or finally

A Exception in ... main java.lang.NullPointerException: Cannot invoke "String.length()" because "clock" is null  
B at Exception.exception(Exception.java:9)  
C at Exception.main(Exception.java:3)

A → exception thrown tells which exception was thrown

B → tells that exception occurred inside exception method of exception class

C → tells that main method of exception class called the exception method

To read stack trace:

- 1) bottom & go up
- 2) first method called is at bottom(main), & the method where the exception occurred is at the top(exception)

→ 1st line in stack trace = origin of exception -  
class name, method name,  
file name, & line number  
where exception object  
was first thrown

→ handle exceptions:

1) avoid them

2) handle them

Checked

→ compile time

→ handle it (or)

→ specify exception  
using throws  
keyword

→ IO exception,  
class Not Found

→

add a throws clause to  
method signature

```
public void readfile() throws IO Exception {  
    FileReader file = new FileReader("hello.txt");  
    {  
    }  
}
```

Unchecked

→ runtime

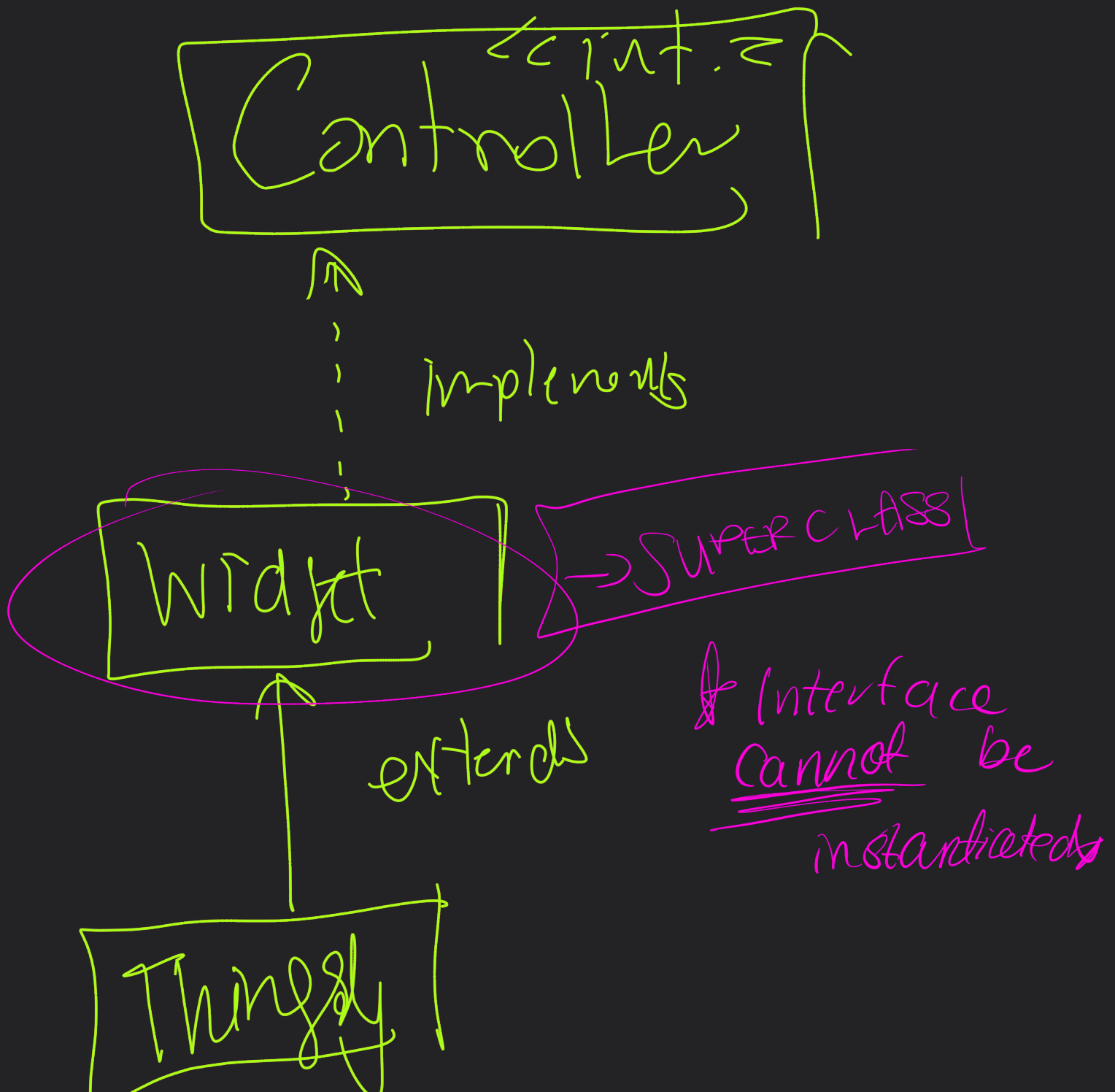
→ Arithmetic, NullP,

Array Index Out...

→

\* means that any  
method that calls  
readfile is responsible  
for handling the  
IO Exception \*

→ explicitly throw exceptions:  
throw new IllegalArgumentException(...);



```
public class Square extends Polygon {  
    private int side;  
    public Square(int side) {  
        super(true);  
        this.side = side  
    }  
}
```

