

A painting of a Dutch landscape, likely a windmill scene. The scene is dominated by a large, green, grassy hill in the foreground. In the background, several windmills are visible, some with their sails partially open. The sky is a mix of light and dark tones, suggesting a cloudy day. The overall style is characteristic of 17th-century Dutch landscape painting.

J15 Exceptions

Java Exceptions
Catch or Specify
Java syntax

Exceptions

Exceptions are a **control flow construct for error management**.

Some similarity to event handling (lecture topic **X02**)

- Both disrupt the normal flow of execution, transferring to event handler or exception handler
- However: exceptions are exceptional situations (events are expected)
 - A file is not found or is inaccessible
 - An array is accessed incorrectly (out of bounds)
 - Division by zero
 - A null pointer is dereferenced, etc.

Java Exceptions

Exceptions are *thrown* either:

- Implicitly (via a program error) or
- Explicitly (by executing the `throw` statement).

Exceptions are *caught* with a `catch` block.

Exceptions are propagated from callee to caller (call stack is *unwound*) until a matching handler is found.

Kinds of Java Exception and Compile-time Check

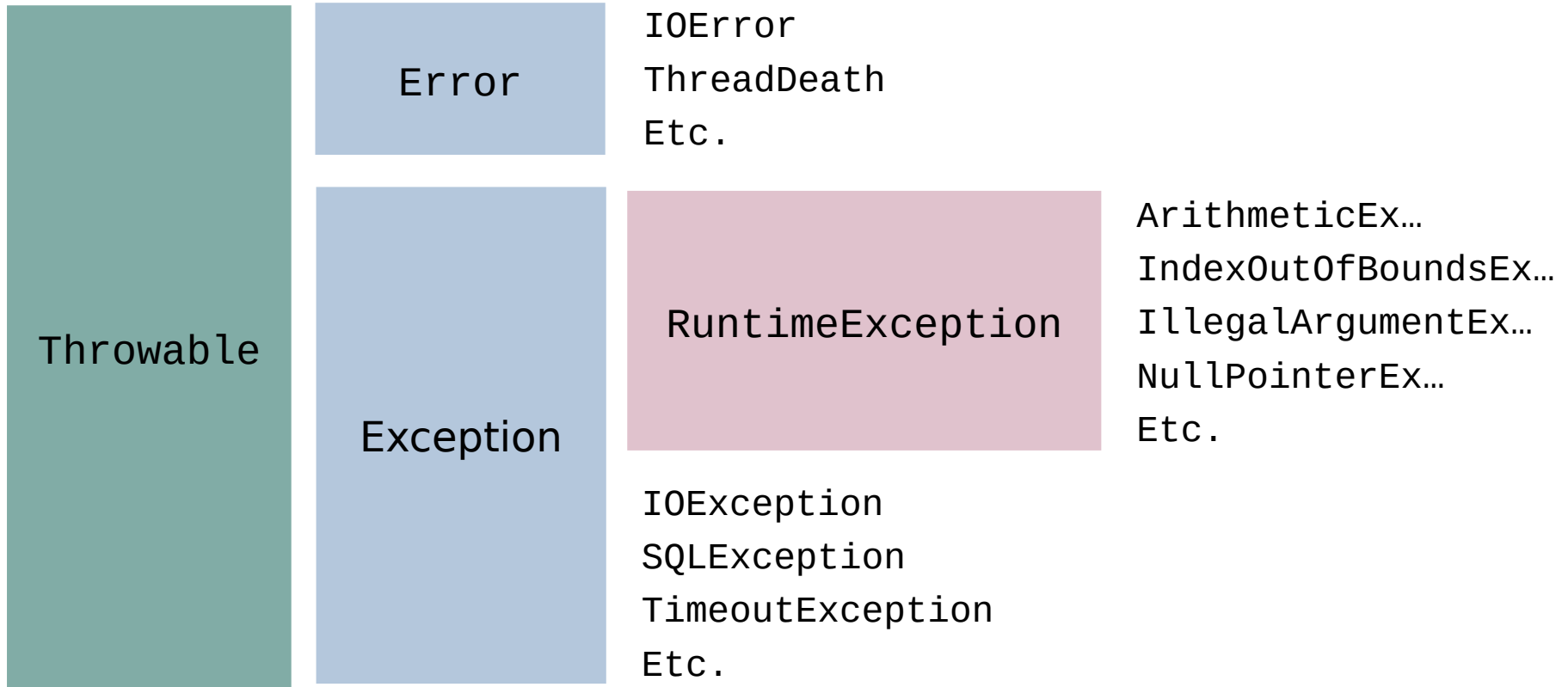
- **error** (Error and its subclasses),
 - serious problems that a reasonable application probably shouldn't attempt to catch
- **runtime exception** (RuntimeException and its subclasses),
 - exceptional situation that often cannot be anticipated or recovered from (e.g., program bugs, logic error, API misuse): probably should fix the bug rather than catch
- **checked exception** (everything else)
 - can be thrown during normal operation and can be reasonably anticipated and handled

unchecked
exceptions

Code that may throw a checked exception must comply with the **catch or specify** requirement, i.e. must be enclosed by either:

- a **try** statement with a suitable handler, or
- a method that declares that it **throws** the exception

Java Exception Type Class Hierarchy



Java try/catch Block Syntax

```
try {  
    // do something that may generate an exception  
} catch (ArithmeticException e1) { // first catch  
    // this is an arithmetic exception handler  
    // handle the error and/or throw an exception  
} catch (Exception e2) { // may have many catch blocks  
    // this an generic exception handler  
    // handle the error and/or throw an exception  
} finally {  
    // this code is guaranteed to run  
    // if you need to clean up, put the code here  
}
```