Exception Handling

Throwable class

- Exception
  - IOException
  - SQLException
  - ClassNotFoundException
  - RuntimeException (unchecked)
    - ArithmeticException - ngoại lệ số học — 50/0
    - NullPointerException — String s=null; call: s.length()
    - NumberFormatException — String s="abc"; call: Integer.parseInt(s)
    - IndexOutOfBoundsException
      - ArrayIndexOutOfBoundsException — int a[]=new int[5]; call: a[10]=5
      - StringIndexOutOfBoundsException
- Error (unchecked)
  - StackOverflowError
  - VirtualMachineError
  - OutOfMemoryError

Exceptions Index

- Try-Catch
  - Luồng xử lý nội bộ — 10 div 0 — Một đối tượng thuộc lớp ngoại lệ được ném ra — Nếu
    - Được xử lý - handled — Phần còn lại của code sẽ được thực thi
    - Không được xử lý, JVM cung cấp trình xử lý ngoại lệ mặc định, làm 3 việc sau
      - In ra mô tả lỗi
      - In ra stack trace
      - Kết thúc chương trình
  - Được sử dụng bên trong phương thức
  - Không đưa code vào trong try block nếu nó không phát sinh ngoại lệ
  - Đi kèm catch block — Có thể sử dụng nhiều khối catch
  - Đi kèm finally block
- Multi-catch
  - Mỗi một catch block chứa những trình xử lý ngoại lệ khác nhau
  - Tại một thời điểm chỉ 1 ngoại lệ xảy ra và 1 catch block được thực thi
  - catch block phải được sắp xếp từ "most specific đến most general"
- Nested Try Block
  - Là khối được sử dụng để thực thi các lệnh quan trọng
    - Đóng kết nối
    - Stream ...
    - closing a file
- Finally Block
  - Luôn luôn được thực hiện dù Exception có được handle hay không
  - Follow khối try hoặc catch
  - Có một khối lệnh Finally duy nhất
  - Finally block sẽ không được thực thi khi
    - chương trình exits (gọi System.exit())
    - Fetal error
- Throw Exception
  - throw được sử dụng để ném ra 1 ngoại lệ tường minh
  - Ném ra ngoại lệ checked hoặc unchecked
  - Được sử dụng chủ yếu để ném ngoại lệ tùy chỉnh - custom
  - Ví dụ: — throw new IOException("sorry device error")
- Sự lan truyền ngoại lệ — Theo nguyên lý stack
- throws keyword
  - Được sử dụng để khai báo một ngoại lệ
  - Cung cấp mã nguồn xử lý ngoại lệ, duy trì luồng bình thường
  - Có thể khai báo nhiều ngoại lệ public void method()throws IOException,SQLException
  - 2 trường hợp sử dụng
    - TH1: xử lý ngoại lệ - handle exception — Đoạn mã sẽ thực hiện tốt cho dù ngoại lệ có xảy ra hoặc không
    - TH2: Khai báo ngoại lệ
      - Nếu ngoại lệ không xảy ra, code vẫn thực thi tốt
      - Nếu ngoại lệ xảy ra, một ngoại lệ sẽ được ném ở thời gian chạy
- final, finally và finalize
  - final được sử dụng để apply những giới hạn cho lớp, phương thức, biến
    - Với lớp final không thể được kế thừa
    - Phương thức final không thể bị ghi đè
    - Biến final không thể bị thay đổi
  - finally được sử dụng để đặt mã nguồn quan trọng cần được thực hiện
  - finalize được sử dụng để clean up/dọn dẹp tiến trình
- Custom Exception