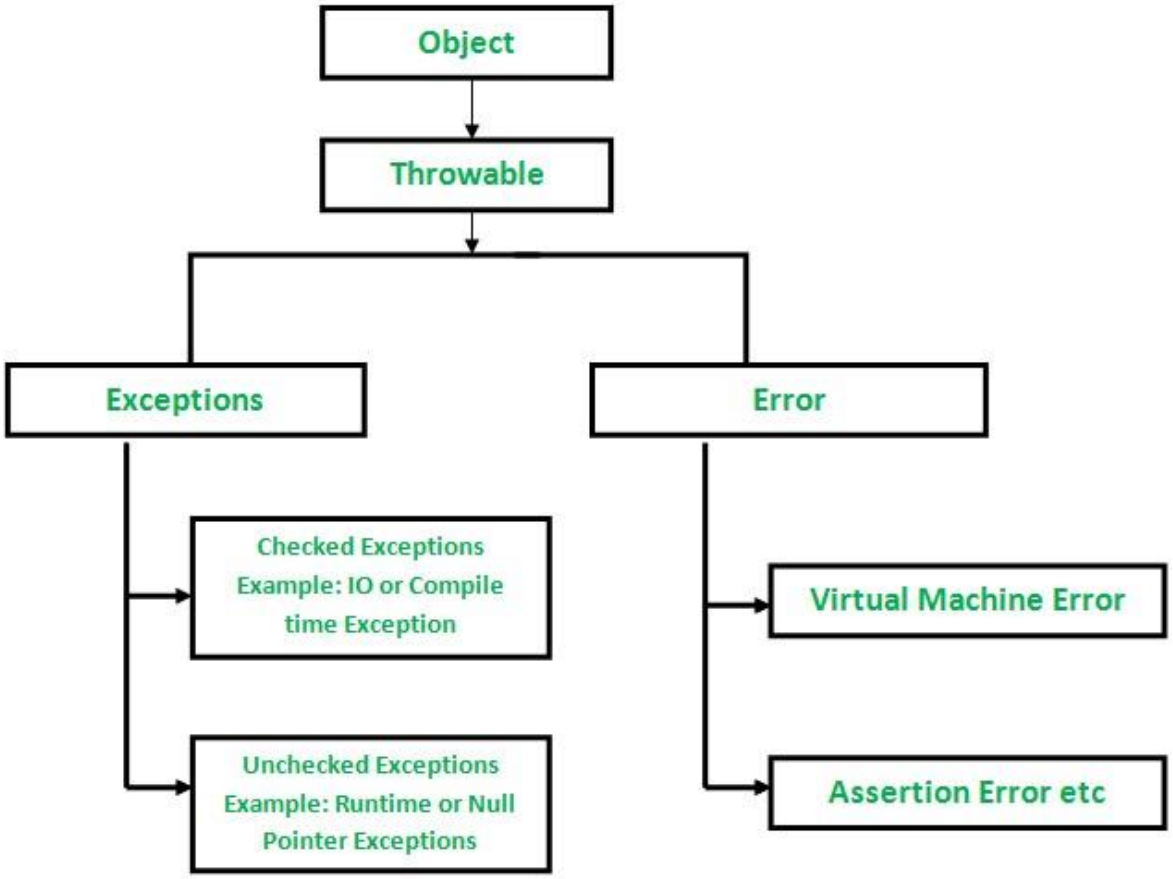
https://github.com/alina-puscasu/pao_lab_2022

Programare avansata pe obiecte – laborator 6

Alina Puscasu *alina.puscasu@endava.com*

Exceptii

- Eveniment care se produce in timpul executiei unui program si perturba buna functionare a acestuia
- Toate exceptiile sunt copii ai clasei **Throwable** (la randul ei copil al Object)
- Ierarhia arata conform imaginii de mai jos:



- Cele doua subclase principale ale lui Throwable sunt **Exception** si **Error**

    - **Error**: nu pot fi anticipate si in general nu sunt tratate explicit intr-o aplicatie (ex: StackOverflowError, OutOfMemoryError, etc)
        - ✦ More on OutOfMemory here
    - **Exception**: se ocupa de exceptii care pot fi rezolvate si tratate
        - ✦ **Checked**: tratate la compilare (ex: IOException, SQLException,

FileNotFoundException, etc); dupa prinderea si tratarea lor programul isi reia buna functionare

- ✦ **Unchecked**: ne-tratate la compilare, apar in timpul executiei programului declansate de factori interni ai aplicatiei, de obicei ca urmare a unui bug, ceea ce necesita gasirea si tratarea sursei (ex: NullPointerException, ArrayIndexOutofBoundsException, etc)

Cum putem arunca exceptii?

- Exceptiile sunt obiecte => pentru a instantia o exceptie se foloseste cuvantul cheie **new**
  - in momentul in care se instantiaza o exceptie in ea se retine intregul lant de apeluri prin care s-a ajuns la instructiunea curenta; aceasta succesiune se numeste **stack trace** si se poate afisa prin apelul **e.printStackTrace()** unde e este obiectul exceptie
- Pentru a arunca o exceptie => se foloseste cuvantul cheie **throw**

Cum tratam exceptiile?

- Cuvantul cheie **throws** in semnatura functiei
- Blocurile **try-catch-finally**

Try-catch-finally

- In blocul *try* punem codul potential generator de exceptii
- Putem avea fie **try cu finally** fie **try cu catch**
- Blocul ***catch*** nu poate exista fara blocul try
  - Putem avea mai multe blocuri catch, insa este important ca exceptiile sa fie de la specifice la generale
  - Putem avea si mai multe exceptii in acelasi bloc, incepand cu Java 7, folosind "**|**"
- Putem avea un singur bloc *finally* care se executa mereu indiferent daca apar sau nu exceptii.

Cum ne cream propria exceptie?

- O noua clasa care sa extinda clasa Exception

Tips and tricks

- Cand marcam o metoda cu cuvantul cheie throws, impacteaza procesul de suprascriere a acelei metode. O subclasa poate arunca un numar mai mic de exceptii decat clasa parinte, dar nu mai multe!

Fluxuri de intrare-iesire

- Vezi pachet fisiere din codul de la laborator

https://github.com/alina-puscasu/pao_lab_2022

Exercitii

1. Sa se citeasca de la tastatura un user si o parola. Se va compara ce se citeste cu inregistrarile existente in fisierul parole.txt. Daca userul si parola se regasesc, se afiseaza mesajul “acces permis”. Daca regasim userul dar parola nu e cea corecta se va semnala o exceptie cu mesajul “parola gresita” si se va trata prin reintroducerea parolei, dar nu mai mult de 3 ori. Daca se atinge acest prag, se va semnala alta eroare cu “cont blocat” si se va incheia programul.

2. Sa se scrie un program care citeste de la tastura perechi de numere in care primul trebuie sa fie mai mic decat al doilea. Daca nu se indeplineste conditia, sa se semnaleze exceptia si sa se trateze prin cererea altei perechi de numere. Toate perechile care indeplinesc conditia se vor scrie intr-un fisier.