**Exceptii**

Exceptie = eveniment ce se produce in timpul executiei unui program si ce perturba functionarea corecta a acestuia.

Exemple de exceptii - in cazul unui program ce citeste date din fisier:

* daca fisierul nu exista
* daca nu avem suficiente permisiuni pentru a-l citi

**Aruncarea exceptiilor**

* Exceptiile sunt obiecte => pentru a instantia o exceptie se foloseste cuvantul cheie **new**
* Pentru a arunca o exceptie => se foloseste cuvantul cheie **throw**

Cateva exemple de exceptii:

* Exception: părintele majorităţii claselor excepţie din Java
* IndexOutOfBoundsException: cand accesam un array/colectie folosind un index ce depaseste marimea alocata a acestuia
* NullPointerException: când se accesează un obiect neinstanţiat (null).
* NoSuchElementException: când se apelează next pe un Iterator care nu mai conţine un element următor.

În momentul în care se instanţiază un obiect-excepţie, în acesta se reţine întregul lanţ de apeluri de funcţii prin care s-a ajuns la instrucţiunea curentă. Această succesiune se numeşte **stack trace** şi se poate afişa prin apelul **e.printStackTrace()**, unde e este obiectul excepţie.

**Tratarea exceptiilor**

Exceptiile pot fi tratate in doua moduri:

* adaugand **throws** in antetul functiei
* folosind un bloc try-catch

**Try - catch - finally**

* putem avea mai multe blocuri catch inlantuite (!ordinea exceptiilor trebuie sa fie de la cea mai specifica la cea mai putin specifica)
* codul din catch se executa in cazul in care se arunca exceptia respectiva
* finally - poate fi adaugat dupa catch; codul din acest bloc se executa **intotdeauna**, si daca a fost aruncata exceptia, si daca nu.

**Clasificarea exceptiilor**

Nu toate exceptiile trebuie prinse.

* Checked exceptions: trebuiesc prinse si tratate
* Unchecked exceptions: nu trebuie

**Checked exceptions**

Aceste exceptii ar trebui prinse, tratate, iar apoi executia programului ar trebui sa fie reluata Exemplu: daca se citeste numele unui fisier de la tastatura, aplicatia urmand a-l deschide si parsa, in cazul in care acest fisier nu exista, ar trebui sa aruncam o eroare; aceasta poate fi tratata ulterior, astfel incat utilizatorul sa-si dea seama ca a intervenit o eroare (ex: sa ii afisam un mesaj de eroare)

**Errors**

* definesc situaţii excepţionale declanşate de factori externi aplicaţiei, ce nu pot fi de obicei anticipate sau tratate Exemplu: alocarea unui obiect foarte mare => OutOfMemoryError.

**Runtime Exceptions**

* situatii excepţionale, declanşate de factori interni aplicaţiei. Aplicaţia nu poate anticipa şi nu îşi poate reveni dacă acestea sunt aruncate.
* Aceste exceptii de obicei semnaleaza prezenta unui bug in cod, deci in loc sa le prindem, ar trebui sa gasim sursa problemei si sa o tratam.
* exemplu: NullPointerException

**Good practices:**

* nu folositi return intr-un bloc finally
* nu folositi throws intr-un bloc finally
* nu ignorati exceptiile (eg: doar sa le printam si sa continuam executia)