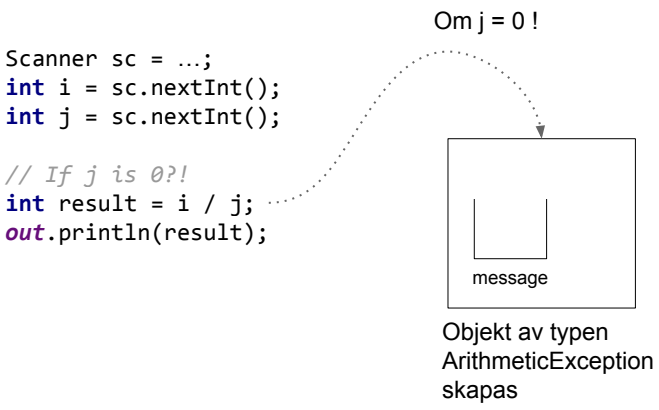
ex4exceptions

Joachim von Hacht

Undantag



2

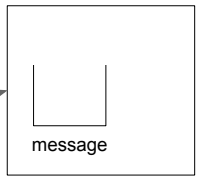
Programmet har under körning hamnat i en omöjlig situation (division med 0)

- Ett undantag (exception) uppstår
- I samband med undantaget:
  - Skapas automatisk ett objekt som bl.a. innehåller information om undantaget
  - Kommer programmet att vandra genom anropsstacken (metod för metod) för att söka efter en s.k undantagshanterare (en try/catch-sats mer strax).
    - Programflödet blir annorlunda, vi kan få ett s.k. icke-lokalt hopp, programmet hoppar rakt igenom flera metoder.
  - Om ingen undantags hanterare hittas i någon metod avbryts (kraschar) programmet
    - En felutskrift skrivs ut.

Fånga undantag

```
int i = ...;
int j = ...;
int result;

try {
    result = i / j; // If 0..
} catch (AritmeticException e ){
    // .. do this
    out.println("You divided by zero!");
}
out.println(result);
```



3

Att fånga ett undantag innebär att programmet inte avbryts (kraschar)

- Man fångar ett undantag genom att lägga ett anrop som kan kasta ett undantag i try-delen av en **try-catch-sats**
- Om inget undantag uppstår körs bara satserna i try-blocket, catch-blocket hoppas över.
- Om ett undantag uppstår, någonstans i try-delen, kommer satserna i blocket efter catch att utföras
    - Kvarvarande satser (efter undantaget) i try-blocket körs inte
    - Tanken är att man, i catch-delen, skall kunna åtgärda felet
    - Efter detta räknas undantaget som fångat och programmet fortsätter med första sats efter catch-blocket
- I catch-grenen initieras automatisk en parameter med en referens till undantagsobjektet som skapades
    - Typen på undantagsobjektet måste stämma med parametertypen
        - Annars sker ingen “fångning”

Kasta Undantag

```
public void add( String s ){
  if( s == null){
    // Create exception object and throw
    throw new IllegalArgumentException("nulls not allowed");
  }
  arr[i] = s;
  i++;
}


try {
    add( str );    // Call method
} catch (IllegalArgumentException e ){
    out.println( e.getMessage() ); // nulls not allowed
}
```

4

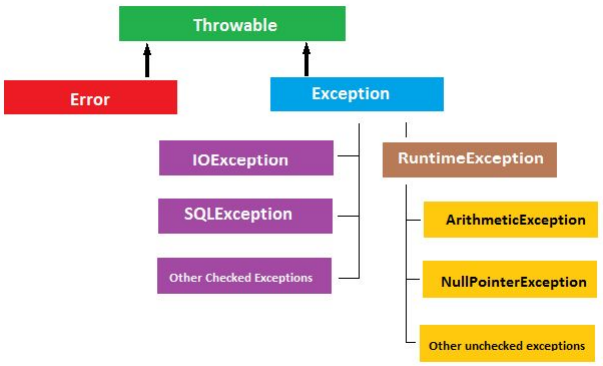
Man kan programmatiskt (i koden kasta) undantag.

- Antag t.ex. att vi har en metod som sparar referenser i en array och vi inte tillåter null-referenser i arrayen …
- … om någon annan programmerare gör fel och skickar in en referens, … vad göra???
- Jo, vi kan låta metoden kasta ett undantag om parametern är null … på så sätt upptäcks felet direkt (i stället för att null-värdet sparas och felet dyker upp långt senare).

Ett eget undantag kastas m.h.a. throw + att man skapar ett objekt av någon undantagsklass, mer strax.

- Meddelandet, argumentet till konstruktorn, kan avläsas med metoden e.getMessage() i catch-grenen
- Finns även e.printStackTrace(), skriver ut hela anropskedjan (stacken)

Undantagsklasser



5

Undantagsobjekten är instanser av olika undantagsklasser

- … klasserna är ordnade i en arvshierarki (d.v.s. det finns ett super/subtyp förhållande)

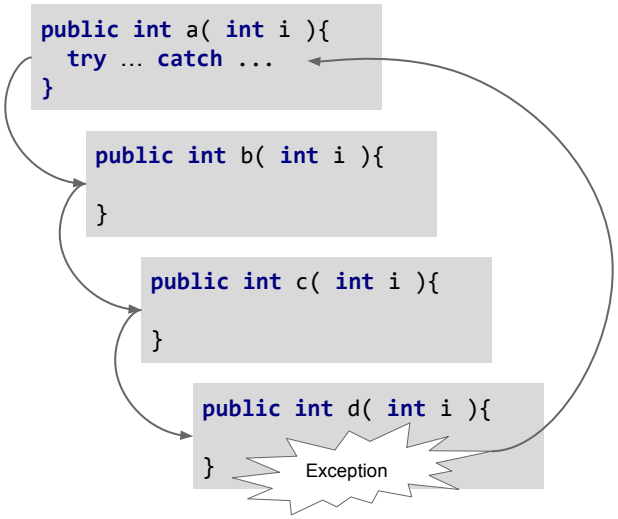
Undantagsklasserna direkt under Exception (de lila) representerar undantag som måste fångas (checked exceptions)

- D.v.s. situationer som kan kasta ett undantagsobjekt av dessa typer måste stå i en try..catch-sats
  - Kompilatorn kontrollerar, vi får ett kompileringsfel om vi inte fångar!
- Många färdiga Java-metoder kan kasta dessa typer av undantag, mer strax

Undantagsklasserna under Runtime är okontrollerade undantag (unchecked exception)

- Vi är inte tvungna att fånga dem.
- Normalt fångar man inte unchecked exceptions …
  - … man vill att undantaget skall krascha programmet (under utvecklingen) eftersom det finns ett programmeringsfel (vi har gjort fel)

Programflöde vid Undantag



6

Ett undantag kan uppstå långt ner i en anropskejda

- I bilden: Metod a anropar b, som anropar c, som anropar d … ett undantag uppstår.
- Undantagshantering kommer att vandra genom anropen (anropsstacken) till dess den hittar en try..catch som kan fånga undantaget.
  - Vi får ett **icke-lokalt hopp** …
  - … programmet hoppar och fortsätter i en annan metod än den anropande.. (kanske väldigt långt bort, i någon helt annan klass…)
  - Finns ingen try-catch avbryts som sagt programmet.