



Principet e Avancuara të Gjuheve të Programimit

Lamir Shkurti, Can. PhD.
lamir.shkurti@gmail.com

EXCEPTIONS



- Një exception (përjashtim) është një ngjarje e padëshiruar ose e papritur, e cila ndodh gjatë interpretimit të një programi d.m.th. në kohën e ekzekutimit, që prish rrjedhën normale të udhëzimeve të programit.
- Kuptimi i fjalorit: Exception është një gjendje anormale.
- Trajtimi i Exceptions është një mekanizëm për të trajtuar gabimet e ekzekutimit të tilla si `ClassNotFoundException`, `IOException`, `SQLException`, `RemoteException`, etj.

AVANTAZHI I TRAJTIMIT TË EXCEPTIONS

Avantazhi kryesor i trajtimit të exceptions është ruajtja e rrjedhës normale të aplikacionit. Një exception normalisht prish rrjedhën normale të aplikacionit, prandaj ne përdorim trajtimin e exceptions (përgjashtimeve).

Shembull

```
deklarata1;  
deklarata2;  
deklarata3;  
deklarata4;  
deklarata5; // ndodh përgjashtimi  
deklarata6;  
deklarata7;  
deklarata8;  
deklarata9;  
deklarata10;
```

Supozoni se ka 10 deklarata në programin tuaj dhe ndodh një përgjashtim në deklaratën e 5, pjesa tjetër e kodit nuk do të ekzekutohet, d.m.th. deklarata 6 deri në 10 nuk do të ekzekutohet. Nëse kryejmë trajtim të exceptions (përgjashtimeve), pjesa tjetër e deklaratës do të ekzekutohet. Kjo është arsyeja pse ne përdorim trajtimin e përgjashtimeve në Java.

LLOJET E EXCEPTION NË JAVA



Ekzistojnë kryesisht dy lloje të exceptions (përjashtimeve): të kontrolluara dhe të pakontrolluara.

Sipas Oracle, ekzistojnë tre lloje të përjashtimeve:

1. Exception (Përjashtim) i kontrolluar
2. Exception (Përjashtim) i pakontrolluar
3. Error

DALLIMI NË MES PËRJASHTIMEVE TË KONTROLLUARA DHE TË PAKONTROLLUARA

Exception (përrjashtim) i kontrolluar

- Klasat që trashëgojnë drejtpërdrejt klasën Throwable përveç RuntimeException dhe Error njihen si Exception (përrjashtime) të kontrolluara p.sh. IOException, SQLException etj.
- Exceptions (përrjashtimet) e kontrolluara kontrollohen në kohën e përpilimit (compile-time).

Exception (përrjashtim) i pakontrolluar

- Klasat të cilat trashëgojnë RuntimeException njihen si exception (përrjashtime) të pakontrolluara p.sh. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etj.
- Exceptions (përrjashtimet) e pakontrolluara nuk kontrollohen në kohën e përpilimit, por ato kontrollohen në kohën e intepretimit (runtime).

Error

- Error-i është i pariparueshëm p.sh. OutOfMemoryError, Virtual MachineError, AssertionError etj.

FJALË KYÇE TË EXCEPTION NË JAVA



try - Fjala kyçe "**try**" përdoret për të specifikuar një bllok ku duhet të vendosim kodin e exception (përjashtimit). Blloku try duhet të ndiqet nga catch ose finally. Do të thotë, nuk mund të përdorim vetëm try.

catch - Blloku "**catch**" përdoret për të trajtuar exception (përjashtimin). Duhet të paraprihet nga blloku i try që do të thotë se nuk mund të përdorim bllokun catch vetëm. Mund të pasohet nga blloku finally.

finally - Blloku "**finally**" përdoret për të ekzekutuar kodin e rëndësishëm të programit. Ekzekutohet nëse trajtohet një exception (përjashtim) ose jo.

throw - Fjala kyçe "**throw**" përdoret për të hedhur një exception (përjashtim).

throws - Fjala kyçe "**throws**" përdoret për të deklaruar exceptions (përjashtime). Nuk e hedh një exception (përjashtim), por specifikon që mund të ndodhë një exception (përjashtim) në metodë.

SKENARËT E ZAKONSHËM TË EXCEPTION (PËRJASHTIMEVE) NGA JAVA

- **ArithmeticException**
 - `int a = 50/0; //ArithmeticException`
- **NullPointerException**
 - `String s = null;`
 - `System.out.println(s.length()); //NullPointerException`
- **NumberFormatException**
 - `String s = "abc";`
 - `int i= Integer.parseInt(s); //NumberFormatException`
- **ArrayIndexOutOfBoundsException**
 - `int a[] = new int[5];`
 - `a[10]=50; //ArrayIndexOutOfBoundsException`

BLLOKU TRY-CATCH



- Bllokutry-catch përdoret për të mbyllur kodin që mund të sjellë një exception (përrjashtim). Duhet të përdoret brenda metodës.
- Nëse ndodh një exception (përrjashtim) në deklaratën e veçantë të bllokut try, pjesa tjetër e kodit të bllokut nuk do të ekzekutohet.
- Blloku try duhet të ndiqet nga catch ose nga blloku finally.

SINTAKSA



```
try{  
    //code that may throw an exception  
}  
catch(Exception_class_Name ref){  
}
```

```
try{  
    //code that may throw an exception  
}  
finally{  
}
```

SHEMBULL I TRAJTIMIT TË EXCEPTION



SHEMBULL

```
public class JavaExceptionExample2 {  
    public static void main(String args[]) {  
        //error  
        int data=100/0;  
  
        //Pjesa tjeter e kodit  
        System.out.println("rest of the code...");  
    }  
}
```

SHEMBULL I TRAJTIMIT TË EXCEPTION

```
public class JavaExceptionExample {  
    public static void main(String args[]) {  
        try{  
            //exception  
            int data=100/0;  
        }  
        catch (ArithmeticException e) {  
            System.out.println(e);  
        }  
        //Pjesa tjeter e kodit  
        System.out.println("Vazhdimi i kodit..");  
    }  
}
```

SHEMBULL I TRAJTIMIT TË EXCEPTION

```
public class JavaExceptionExample3{
    public static void main(String args[]) {
        try{
            //exception
            int data=100/0;

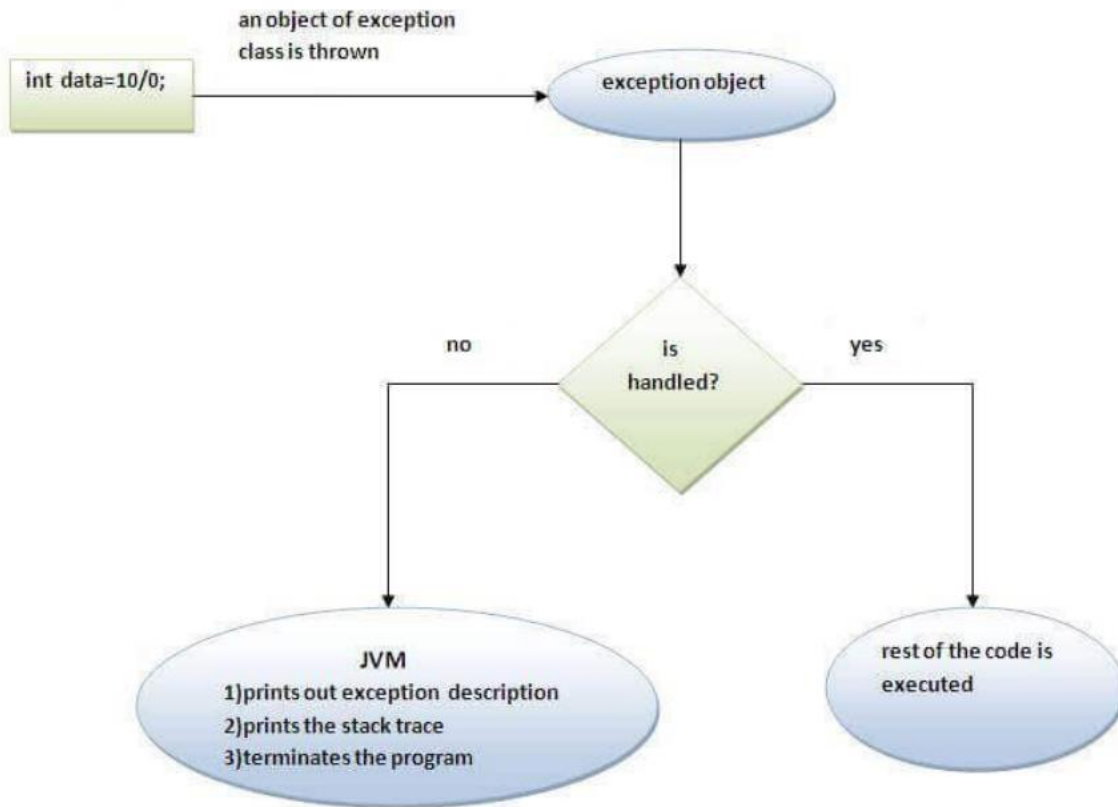
            //Pjesa tjeter e kodit
            System.out.println("Vazhdimi i kodit ...");
        }
        catch (ArithmeticException e) {
            System.out.println(e);
        }
    }
}
```

SHEMBULL I TRAJTIMIT TË EXCEPTION

```
public class JavaExceptionExample4{
    public static void main(String args[]) {
        try{
            //exception
            int data=100/0;

            //Pjesa tjeter e kodit
            System.out.println("Vazhdimi i kodit ...");
        }
        catch (ArithmeticException e) {
            System.out.println("Nuk mund te pjesetohet me
            zero");
        }
    }
}
```

BLLOKU TRY-CATCH



TRY-CATCH-FINALLY



Blloku try

- Nëse një përjashtim ngritet/ndodhë në bllokun try, atëherë blloku përkatës catch përcaktohet të trajtoj përjashtimin
- Nëse nuk ekziston blloku përkatës catch, përjashtimi duhet të përcillet tek metoda thirrëse.

Blloku catch (një ose më shumë)

- Fillon me catch dhe emrin e llojit të exception (përjashtimit) që duhet trajtuar
- Përjashtimet në Java ndërtojnë një hierarki të trashëgimisë (inheritance); blloku catch mund të trajtojë po ashtu të gjitha përjashtimet e specializuara (IOException)

Blloku finally (opsional)

- Gjithmonë ekzekutohet, ka apo nuk ka raste të veçanta / përjashtime
- Detyrat tipike: kthimi i resurseve, mbyllja e dokumenteve etj.

PËRCJELLJA E PËREJASHTIMEVE (THROWS)



Deri më tani: Metodat i kanë trajtuar vet përjashtimet (try-catch)

E re: Nëse metoda nuk mund të trajtoj exceptions (përjashtimet, mund ta përcjell exception (përjashtimin) tek metodat thirrëse

Fjala e rezervuar throws

- Informatë për programerin dhe për kompajlerin se çfarë përjashtimi mund të paraqitet në metodë

Deklarimi i fjalës throws në pjesën e definimit të metodës:

```
public Object metoda() throws Exception {  
  
}
```


SHEMBULL PËRCJELLJA E PËREJASHTIMEVE (THROWS)

```
public class TestThrow1{
    static void validate(int age){
        if(age<18)
            throw new ArithmeticException("not valid");
        else
            System.out.println("welcome to vote");
    }
    public static void main(String args[]){
        validate(13);
        System.out.println("rest of the code...");
    }
}
```

Rezultati:

Exception in thread main java.lang.Arithmetic Exception:not valid

EXCEPTION I PERSONALIZUAR



- Nëse jeni duke krijuar exception tuaj, njihet si exception (përjashtim) i personalizuar ose exception (përjashtim) i përcaktuar nga përdoruesi.
- Me ndihmën e exception (përjashtimit) të personalizuar, ju mund të keni exception (përjashtimin) dhe mesazhin tuaj.

EXCEPTION I PERSONALIZUAR

```
public class InvalidAgeException extends Exception{
    public InvalidAgeException (String s) {
        super(s);
    }
}

public class TestCustomException7{
    public static void validate (int age) throws InvalidAgeException{
        if (age<18)
            throw new InvalidAgeException ("not valid");
        else
            System.out.println("welcome to vote");
    }
    public static void main(String args[]) {
        try{
            validate (13);
        }
        catch (Exception m) {
            System.out.println("Exception occurred: "+m);
        }
        System.out.println("rest of the code...");
    }
}
```

Principet e Avancuara të Gjuhëve të Programimit

