Exceptions

DE HOGESCHOOL
MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

INHOUD

- Wat zijn exceptions?
- Exception handling
- Exception types
- Custom exceptions
- Exception throwing
- Unit tests voor exceptions
- Oefeningen

Wat zijn exceptions?

- Gebeurtenis tijdens uitvoeren van code
- Verstoort normale *flow*

- Exception wordt *gegooid* (throw)
- ... en kan opgevangen worden (*catch*)

- Indien niet opgevangen: programma stopt met uitvoeren

Wat zijn exceptions?

Code:

```
public static void main(String[] args) {

    int[] lijstje = { 1, 1, 2, 3, 5, 8};

    System.out.println(lijstje[13]);

}
```

Wat zijn exceptions?

Code:

```
public static void main(String[] args) {

    int[] lijstje = { 1, 1, 2, 3, 5, 8};

    System.out.println(lijstje[13]);

}
```

throws exception

Wat zijn exceptions?

Intern:

```
…   (failure in code)

throw new ArrayIndexOutOfBoundsException(…);

…
```

Wat zijn exceptions?

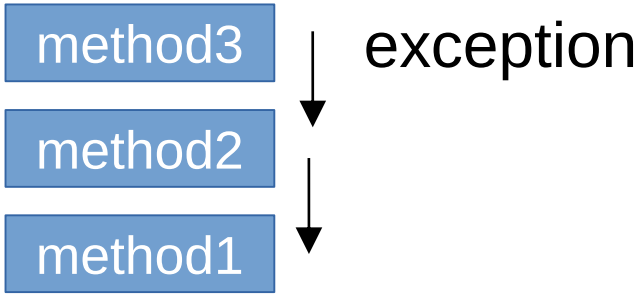
Console toont stack trace & info:

```
Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException:
     Index 13 out of bounds for length 6
   at ExceptionsDemo.main(ExceptionsDemo.java:11)
```

→ programma gestopt

Exception handling

- Opvangen van exception
- Call stack wordt doorlopen



Exception handling

Syntax

```
try {
   // Code die “mogelijk” een exception genereert
}
catch (Throwable exceptionObject) {
   // Code om de exception af te handelen
   // Lijkt op een methode met 1 parameter
   // Exception-object wordt doorgegeven aan catch-blok
   // Exception-object moet afgeleid zijn van de class
   // “Throwable”
}
```

Exception handling

```
public static void main(String[] args) {

    int[] lijstje = { 1, 1, 2, 3, 5, 8};

    System.out.println(lijstje[13]);

}
```

Genereert exception

Exception handling

Exceptions opvangen:

```
public static void main(String[] args) {

    int[] lijstje = {1, 1, 2, 3, 5, 8};
    try {
        System.out.println(lijstje[13]);
    } catch(ArrayIndexOutOfBoundsException exception) {
        System.out.println("Foute index!");
    }
    // Programma wordt verder uitgevoerd
}
```

Exception handling

```
…
catch(NullPointerException exception) {
   System.out.print(exception.getMessage());
   exception.printStackTrace();
}
```

getMessage(): omschrijving van de exception

printStackTrace(): waar vond de exception plaats?

Meerdere exceptions

- Meerdere mogelijke exceptions in try-blok
- Afzonderlijk catch-blok voor elk type

```
try {
  // Code die “mogelijk” een exception genereert
}
catch (ThrowableClass1 exceptionObject) {
       // afhandeling van exceptions van klasse ThrowableClass1
}
catch (ThrowableClass2 exceptionObject) {
       // afhandeling van exceptions van klasse ThrowableClass2
}
catch (ThrowableClass3 exceptionObject) {
       // afhandeling van exceptions van klasse ThrowableClass3
}
```

Finally

Wordt **altijd** uitgevoerd, met of zonder exception

Bestanden afsluiten

Netwerkconnectie sluiten

```
try {
        …
}
catch (Exception ex){
        …
}
finally {
        …
}
```

Oefening

Gegeven de code in *ArithmeticDemo.java*

1. Zoek wat er fout zou kunnen gaan bij uitvoeren van deze code
2. Zoek de methode Integer.parseInt() in de JavaDoc
   - Welke exception kan deze methode genereren?
3. Wat kan er verkeerd gaan bij de deling?
   - Kan je dit ook opvangen?
4. Plaats het keyboard.close() event in het *finally*-blok

parseInt

```
public static int parseInt(String s)
                    throws NumberFormatException
```

Parses the string argument as a signed decimal integer. The characters in the string must all be decimal digits, except that the first character may be an ASCII minus sign `'-'` (`'\u002D'`) to indicate a negative value or an ASCII plus sign `'+'` (`'\u002B'`) to indicate a positive value. The resulting integer value is returned, exactly as if the argument and the radix 10 were given as arguments to the `parseInt(java.lang.String, int)` method.

Parameters:

`s` - a `String` containing the `int` representation to be parsed

Returns:

the integer value represented by the argument in decimal.

Throws:

`NumberFormatException` - if the string does not contain a parsable integer.

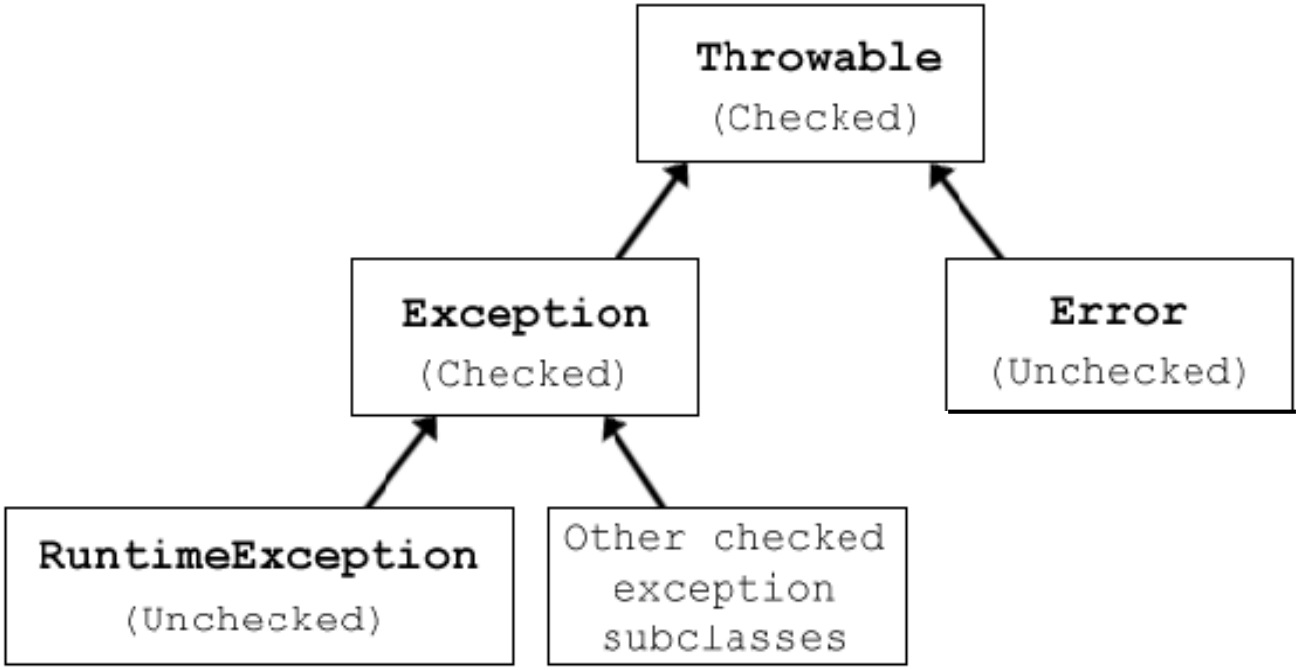

```
5
0
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at ArithmeticDemo.main(ArithmeticDemo.java:8)

Process finished with exit code 1
```

Exception types

- Unchecked exception
  - Runtime exception
  - Mogen opgevangen worden
  - bv. NullPointerException, IndexOutOfBoundsException
- Checked exception
  - Compile time exception
  - Grote kans op fout
  - Moéten opgevangen worden
  - bv. IOException, …

Exception types



NullPointerException, ArrayIndexOutOfBoundsException = ?

Exception handling

Exception handler die alle exceptions opvangt:

```
try {
   …
}
catch (Exception ex) {
      …
}
```

Bad practice Geen info over welke fout is opgetreden

Custom exceptions

- Startcode: *CustomExceptionsDemo.java*

```java
public class Person {
    private String name;

    public Person(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String toString() {
        return this.name;
    }
}
```

```java
public class CustomExceptionsDemo {

    public static String[] names = {"Bart", "Nele", "Sam", ""};

    public static void main(String[] args) {
        ArrayList<Person> list = initialize();
    }

    public static ArrayList<Person> initialize() {
        ArrayList<Person> personList = new ArrayList<~>();

        for(String name : names) {
            Person person = createPerson(name);
            personList.add(person);
        }
        return personList;
    }

    public static Person createPerson(String name) {
        return new Person(name);
    }

}
```

Custom exception

- Eigen exception definiëren
- Bij ongeldige naam in *createPerson()*
- Lege String als waarde: InvalidNameException

Custom exceptions

- Afleiden van Exception of RuntimeException
  - Checked vs Unchecked

```
public class InvalidNameException extends Exception {
    public InvalidNameException(String msg) {
        super(msg);
    }

}
```

Throw custom exception

```
public Person createPerson(String name) {
    if(name.isBlank()) {
        throw new InvalidNameException(name + "is not a name");
    }

    return new Person(name);
}
```

Throw exception

```
public Person createPerson(String name) {
    if(name.isBlank()) {
        throw new InvalidNameException(name + "is not a name");
    }

    return new Person(name);
}
```

Compilation error:
Unhandled exception

Checked exceptions

- Methode **moet**:
  - Exception opvangen

```
try {
    ...
} catch(InvalidNameException
                    exception) {
    ...

}
```

OF

  - Exception ‘omhoog’ werpen

```
public Person createPerson(String name)
        throws InvalidNameException {
    ...
}
```

Checked exceptions

```
public static Person createPerson(String name) {
    if(name.isBlank()) {
        throw new InvalidNameException(name + " is not a correct name");
    }

    return new Person(name);
}
```

- Add exception to method signature
- Surround with try/catch
- Remove braces from 'if' statement

```
public static Person createPerson(String name) throws InvalidNameException {
    if(name.isBlank()) {
        throw new InvalidNameException(name + " is not a correct name");
    }

    return new Person(name);
}
```

Checked exceptions

```
public static Person createPerson(String name) {
    if(name.isBlank()) {
        throw new InvalidNameException(name + " is not a correct name");
    }

    return new Person(name);
}
```

- Add exception to method signature
- Surround with try/catch
- Remove braces from 'if' statement

```
public static Person createPerson(String name) throws InvalidNameException {
    if(name.isBlank()) {
        throw new InvalidNameException(name + " is not a correct name");
    }

    return new Person(name);
}
```

Checked exceptions

```
public ArrayList<Person> initialize() {
    ArrayList<Person> personList = new ArrayList<Person>();
    String[] names = {"Bart", "Nele", "Sam", ""};
    for(String name : names) {
        Person person = createPerson(name);
        personList.add(person);
    }
    return personList;

}
```

Compilation error:
Unhandled exception

Checked exceptions

```
public static ArrayList<Person> initialize() {
    ArrayList<Person> personList = new ArrayList<Person>();
    String[] names = {"Bart", "Nele", "Sam", ""};
    for(String name : names) {
        try {
            Person person = createPerson(name);
            personList.add(person);
        } catch (InvalidNameException exception) {
            System.out.println(exception.getMessage());
        }
    }
    return personList;

}
```

Checked exceptions

```
public static ArrayList<Person> initialize() {
    ArrayList<Person> personList = new ArrayList<Person>();
    String[] names = {"Bart", "Nele", "Sam", ""};
    for(String name : names) {
        try {
            Person person = createPerson(name);
            personList.add(person);
        } catch (InvalidNameException exception) {
            System.out.println(exception.getMessage());
        }
    }
    return personList;

}
```

Scope!

Oefening

Gegeven de code in *CustomExceptionDemo.java*

1. Maak de klasse InvalidNameException aan
2. Zorg dat deze op het juiste moment gegooid wordt
3. Gooi de exception verder
4. Vang hem op in *initialize()*
5. Print de bijhorende message en stack trace wanneer de exception optreedt

Unit testing exceptions

- Testen of exception geworpen wordt
- Opnemen in automatische tests

- Condities voor exception triggeren

Unit testing exceptions

```
@Test
void testExpectedException() {

  Assertions.assertThrows(NumberFormatException.class, () -> {
    Integer.parseInt("One");
  });

}
```

Unit testing exceptions

```
@Test
void testInvalidNameException() {

  Assertions.assertThrows(InvalidNameException.class, () -> {
      Person p = CustomExceptionsDemo.createPerson("");
  });

}
```

Test Results	32 ms
PersonTest	32 ms
testInvalidNameException()	32 ms

```
"C:\Program Files\Java\jdk-11.0.4\bin\java.exe" ...

Process finished with exit code 0
```

Leerstof

- Handboek: Hoofdstuk 1
- Oefening: VriendenApp – zie Blackboard

- Extra: PluralSight - Exception handling