

Uitwerkingen oefenopgaven hoofdstuk 4

Oefenopgave 1

```
package pargr4_1;

public class Persoon {
    private String naam;
    private String achterNaam;
    private String woonplaats;
    public Persoon() {
    }

    public Persoon(String naam, String achterNaam, String woonplaats) {
        this.naam = naam;
        this.achterNaam = achterNaam;
        this.woonplaats = woonplaats;
    }

    public String getNaam() {
        return naam;
    }

    public void setNaam(String naam) {
        this.naam = naam;
    }

    public String getAchterNaam() {
        return achterNaam;
    }

    public void setAchterNaam(String achterNaam) {
        this.achterNaam = achterNaam;
    }

    public String getWoonplaats() {
        return woonplaats;
    }

    public void setWoonplaats(String woonplaats) {
        this.woonplaats = woonplaats;
    }

    public String toonNAW() {
        return naam + ", " + achterNaam + ", " + woonplaats;
    }

    @Override
```

```

    public String toString() {
        return "Persoon{" + "naam=" + naam + ", achterNaam=" +
achterNaam + ", woonplaats=" + woonplaats + '}';
    }
}

public class TestPersoon {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Persoon p1 = new Persoon("Piet", "Jansen", "Helvoirt");
        Persoon p2 = new Persoon("Piet", "Jansen", "Helvoirt");
        Persoon p3 = new Persoon("Karel", "Heinen", "Utrecht");
        System.out.println(p1.toonNAW());

        boolean gelijk = p1.getNaam() == p2.getNaam();
        System.out.println("gelijk = " + gelijk);
        gelijk = p1.getNaam() == p3.getNaam();
        System.out.println("gelijk = " + gelijk);
        gelijk = p1.getNaam().equals(p2.getNaam());
        System.out.println("gelijk = " + gelijk);
        gelijk = p1.getNaam().equals(p3.getNaam());
        System.out.println("gelijk = " + gelijk);
        String var1 = new String("Piet");
        String var2 = new String("Piet");
        String var3 = new String("Karel");
        gelijk = var1 == var2;
        System.out.println("gelijkv = " + gelijk);
        gelijk = var1 == var3;
        System.out.println("gelijkv = " + gelijk);
        gelijk = var1.equals(var2);
        System.out.println("gelijkv = " + gelijk);
        gelijk = var1.equals(var3);
        System.out.println("gelijkv = " + gelijk);
    }
}

```

Deze methodes zijn:

- a. indexOf (int ch) geeft een indexcijfer terug waar de char als eerste te vinden is.
- b. lastIndexOf (int ch) geeft een indexcijfer terug waar het laatste karakter te vinden is.
- c. indexOf(int ch, int fromIndex) geeft een indexcijfer terug waar de char als eerste te vinden is. De method begint te zoeken vanaf de fromIndex.
- d. lastIndexOf(int ch, int endIndex) geeft een indexcijfer terug waar het laatste karakter te vinden is, maar stopt met zoeken bij de endIndex.
- e. indexOf (String str) geeft een indexcijfer terug waar de substring als eerste begint.
- f. lastIndexOf (String str) geeft een indexcijfer terug, waar het woord voor de laatste keer begint.
- g. indexOf (String str, int fromIndex) geeft een indexcijfer terug, waar de substring als eerste begint. De method begint te zoeken vanaf de fromIndex.
- h. lastIndexOf (String str, int endIndex) geeft een indexcijfer terug, waar het woord voor de laatste keer begint, maar stopt met zoeken bij de endIndex.

Alle acht methodes geven een -1 terug als de char of substring niet voorkomt in de string.

In de volgende code staat van alle methodes een voorbeeld.

```

public class IndexenVragen {
    public static void main(String[] args) {
        String voorbeeld = "Deze mooie taal is vernoemd naar de Java
koffie";
        System.out.println(voorbeeld);
        int met_1 = voorbeeld.indexOf('J');
        System.out.println("De J begint op index " + met_1);
        int met_2 = voorbeeld.lastIndexOf('a');
        System.out.println("De laatste a staat op index " + met_2);
        int met_3 = voorbeeld.indexOf('a', 14);
        System.out.println("De eerste a na positie 14 staat op index " +
met_3);
        int met_4 = voorbeeld.lastIndexOf('a', 15);
        System.out.println("De laatste a voor positie 15 staat op index " +
met_4);
        int met_5 = voorbeeld.indexOf("taal");
        System.out.println("Het woord taal begint op index " + met_5);
        int met_6 = voorbeeld.lastIndexOf("taal");
        System.out.println("Het woord taal begint voor de laatste keer op
index " + met_6);
        int met_7 = voorbeeld.indexOf("vernoemd", 8);
        System.out.println("Het woord vernoemd komt na pos 8 als eerste voor
op index " + met_7);
        int met_8 = voorbeeld.lastIndexOf("Java", 4);
        System.out.println("Als we de laatste index van Java zoeken, maar
stoppen op index 4 " + "dan krijgen we " + met_8);
    }
}

```

Oefenopgave 2

```
package pargr4_2;
```

```

public class StringUtil {

    /**
     * Deze methode verdubbelt de single quotes die in een string,
     * die worden gebruikt in een SQL-uitdrukking, voorkomen.
     */
    public static String fixSqlFieldValue(String value) {
        if (value == null) {
            return null;
        }
        int length = value.length();
        StringBuilder fixedValue = new StringBuilder((int) (length *
1.1));
        for (int i = 0; i < length; i++) {
            char c = value.charAt(i);
            if (c == '\\') {
                fixedValue.append("\\\\");
            } else {
                fixedValue.append(c);
            }
        }
        return fixedValue.toString();
    }
}

```

```

    /**
    * Deze methode vervangt speciale tekens zodat ze op de
    * juiste wijze in de browser zullen getoond worden.
    * De HTML-tag wordt in een StringBuffer geplaatst en elk
    * van de volgende tekens wordt als volgt vervangen: '<'
    * met "&lt;" '>' met "&gt;" '&' met "&amp;" '"' met
    * "&quot;" ' ' met "&nbsp;"
    */
    public static String encodeHtmlTag(String tag) {
        if (tag == null) {
            return null;
        }
        int length = tag.length();
        StringBuilder encodedTag = new StringBuilder(2 * length);
        for (int i = 0; i < length; i++) {
            char c = tag.charAt(i);
            if (c == '<') {
                encodedTag.append("&lt;");
            } else if (c == '>') {
                encodedTag.append("&gt;");
            } else if (c == '&') {
                encodedTag.append("&amp;");
            } else if (c == '"') {
                encodedTag.append("&quot;");
            } else if (c == ' ') {
                encodedTag.append("&nbsp;");
            } else {
                encodedTag.append(c);
            }
        }
        return encodedTag.toString();
    }
}

package pagr4_2;

public class TestStringUtil {

    public static void main(String[] args) {
        System.out.println("<html> 'is' erg 'leuk' ");
        System.out.println(StringUtil.encodeHtmlTag("<html> 'is' erg
'leuk' "));
        System.out.println("< > & 'abcd' ");
        System.out.println(StringUtil.fixSqlFieldValue("< > & 'abcd'
"));
    }
}

```

Oefenopgave 3a

```

package pagr4_3;

public class Inwoner {
    private int leeftijd;
    private byte geslacht;

    public Inwoner(int leeftijd, byte geslacht) {
        this.leeftijd = leeftijd;
    }
}

```

```

        this.geslacht = geslacht;
    }

    public int getLeeftijd() {
        return leeftijd;
    }

    public byte getGeslacht() {
        return geslacht;
    }
}

package pargr4_3;

public class InwonerGegevens {

    Inwoner[] inwoners;

    public InwonerGegevens() {
        inwoners = new Inwoner[1000];
        int a;
        byte b;
        for (int i = 0; i < 1000; i++) {
            a = (int) (100 * Math.random());
            b = (byte) (2 * Math.random());
            inwoners[i] = new Inwoner(a, b);
        }
    }

    public int getJeugd() {
        int jeugd = 0;
        for (Inwoner w : inwoners) {
            if (w.getLeeftijd() <= 18) {
                jeugd++;
            }
        }
        return jeugd;
    }

    public int getVolwassen() {
        int volwassen = 0;
        for (Inwoner w : inwoners) {
            if (w.getLeeftijd() > 18 && w.getLeeftijd() <= 64) {
                volwassen++;
            }
        }
        return volwassen;
    }

    public int getBejaardeMannen() {
        int bejaardM = 0;
        for (Inwoner w : inwoners) {
            if (w.getLeeftijd() >= 65 && w.getGeslacht() == 0) {
                bejaardM++;
            }
        }
        return bejaardM;
    }
}

```

```

    }
}

package pargr4_3;

public class TestCBS {

    public static void main(String[] args) {
        InwonerGegevens inw = new InwonerGegevens();
        System.out.println("aantal jeugd: " + inw.getJeugd());
        System.out.println("aantal volwassen: " + inw.getVolwassen());
        System.out.println("aantal bejaarde mannen: " +
inw.getBejaardeMannen());
    }
}

```

Oefenopgave 3b

```

package pargr4_3;
public class TweeDem {
    private int [][] multiArray;

    public TweeDem() {
        multiArray = new int[2][2];
    }

    public TweeDem(int r, int k ) {
        multiArray = new int[r][k];
    }

    void vulRandom() {
        for (int i=0; i<multiArray.length; i++) {
            for (int j=0; j<multiArray[i].length; j++) {
                multiArray[i][j] = (int)(100*Math.random());
            }
        }
    }

    int kleinste() {
        int kl = 100;
        for (int i=0; i<multiArray.length; i++) {
            for (int j=0; j<multiArray[i].length; j++) {
                if (kl > multiArray[i][j]) kl = multiArray[i][j];
            }
        }
        return kl;
    }

    int grootste() {
        int gr = 0;
        for (int i=0; i<multiArray.length; i++) {
            for (int j=0; j<multiArray[i].length; j++) {
                if (gr < multiArray[i][j]) gr = multiArray[i][j];
            }
        }
        return gr;
    }
}

```

```

String indexen(int getal) {
    for (int i=0; i<multiArray.length; i++) {
        for (int j=0; j<multiArray[i].length; j++) {
            if (getal == multiArray[i][j]){
                return i + " " + j;
            }
        }
    }
    return "-1 -1";
}

}

package pargr4_3;

public class TestArray {

    public static void main(String[] args) {
        TweeDem td = new TweeDem(5,6);
        td.vulRandom();
        System.out.println("kleinste = " + td.kleinste());
        System.out.println("grootste = " + td.grootste());
        System.out.println("index = " + td.indexen(38));
    }

}

```

Oefenopgave 4

```

import java.util.ArrayList;
import java.util.ListIterator;

ArrayList<Persoon> personen = new ArrayList<Persoon>();
personen.add(p1);
personen.add(p2);
personen.add(p3);
personen.set(2, p4);
for (Persoon element : personen) { //zie 4.4.3
    System.out.println(element.toonNAW());
}
ListIterator<Persoon> iterator = personen.listIterator();
while (iterator.hasNext()) {
    System.out.println(iterator.next().toonNAW());
}

```

Oefenopgave 5

```
@Override
public boolean equals(Object obj) {
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Persoon other = (Persoon) obj;
    if (!Objects.equals(this.naam, other.naam)) {
        return false;
    }
    if (!Objects.equals(this.achterNaam, other.achterNaam)) {
        return false;
    }
    if (!Objects.equals(this.woonplaats, other.woonplaats)) {
        return false;
    }
    return true;
}
```