

Voorbeeldoplossing Hoofdstuk 9 oefening 2

```
package be.pxl.h9.oef2;

public class Leraar {
    private String naam;
    private String voornaam;
    private int aanstellingsPerc;
    private Vak[] vakken = new Vak[MAX_AANTAL_VAKKEN];
    private int aantalVakken;
    private static int MAX_AANTAL_VAKKEN = 5;

    public Leraar(String naam, String voornaam, int aanstellingsPerc) {
        this.naam = naam;
        this.voornaam = voornaam;
        setAanstelling(aanstellingsPerc);
    }

    public String getNaam() {
        return naam + " " + voornaam;
    }

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

    public void setVoornaam(String voornaam) {
        this.voornaam = voornaam;
    }

    public void voegVakToe(Vak vak) {
        if (aantalVakken < MAX_AANTAL_VAKKEN) {
            vakken[aantalVakken] = vak;
            aantalVakken++;
        } else {
            System.out.println("Deze leraar heeft reeds " +
MAX_AANTAL_VAKKEN + " vakken");
        }
    }

    public Vak getVak(int i) {
        if (i >= aantalVakken) {
            System.out.println("deze leraar heeft niet zoveel vakken");
            return null;
        } else {
            return vakken[i];
        }
    }

    public int getAanstelling() {
        return aanstellingsPerc;
    }

    public void setAanstelling(int aanstelling) {
        if (aanstelling > 100) {
            aanstellingsPerc = 100;
        } else {
            aanstellingsPerc = (int)(Math.round(aanstelling/10.0)*10);
        }
    }

    public int getAantalVakken() {
        return aantalVakken;
    }
}
```

```

    public static String getMAX() {
        return MAX_AANTAL_VAKKEN;
    }
}

```

```

package be.px1.h9.oef2;

public class Vak {
    private String code;
    private String naam;
    private int aantalStudiepunten;
    private static final int MAX_STUDIEPUNTEN = 18;
    // niet in UML-diagram maar verbetert onderhoud van de klasse

    public Vak(String code, String naam, int studiepunten) {
        this.naam = naam;
        setCode(code);
        setAantalStudiepunten(studiepunten);
    }

    public String getJaar() {
        return code.substring(1, 2);
    }

    public String getNaam() {
        return naam;
    }

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

    public String getCode() {
        return code;
    }

    public void setCode(String code) {
        String jaar = code.substring(1, 2);
        if (jaar.compareTo("1") < 0 || jaar.compareTo("3") > 0) {
            jaar = "1";
            this.code = code.charAt(0) + jaar + code.substring(2);
        } else {
            this.code = code;
        }
    }
}

```

```

    public int getAantalStudiepunten() {
        return aantalStudiepunten;
    }

    public void setAantalStudiepunten(int studiepunten) {
        if (studiepunten > MAX_STUDIEPUNTEN) {
            aantalStudiepunten = MAX_STUDIEPUNTEN;
        } else {
            int rest = studiepunten % 3;
            switch (rest) {
                case 0: aantalStudiepunten = studiepunten; break;
                case 1: aantalStudiepunten = studiepunten - 1; break;
                case 2: aantalStudiepunten = studiepunten + 1; break;
            }
        }
    }
}

package be.pxl.h9.oef2;

public class PXLApp {

    public static void main(String[] args) {
        String[] codeTabel = { "41TIN1100", "48TIN1130", "41TIN1140",
            "41TIN1160", "41TIN1240", "41TIN1290",
            "41TIN1300", "42TIN1220", "42TIN1230", "43AON3120",
            "43AON3130" };

        String[] naamTabel = { "Web Essentials", "Java Essentials", ".Net
Essentials", "SQL Essentials", "Desktop OS",
            "Math for IT", "Programming Basics", ".Net Advanced", "Java
Advanced", "Programming Expert",
            "Mobile development" };
        int[] studiepuntenTabel = { 7, 6, 6, 3, 6, 6, 5, 3, 3, 3, 3 };

        Vak[] vakken = new Vak[codeTabel.length];

        for (int i = 0; i < vakken.length; i++) {
            vakken[i] = new Vak(codeTabel[i], naamTabel[i],
                studiepuntenTabel[i]);
        }

        Leraar leraar = new Leraar("Daems", "Greta", 73);
        leraar.voegVakToe(vakken[1]);
        leraar.voegVakToe(vakken[6]);
        leraar.voegVakToe(vakken[9]);
        leraar.voegVakToe(vakken[7]);
        leraar.voegVakToe(vakken[8]);
        leraar.voegVakToe(vakken[0]);
        print(leraar);
    }

    public static void print(Leraar leraar) {
        System.out.println("Leraar " + leraar.getNaam() + " is aangesteld
voor " + leraar.getAanstelling() + "%");
        System.out.println("Volgende vakken behoren tot het takenpakket:");
        for (int jaar = 1; jaar <= 3; jaar++) {

```

```

String jaarString = String.valueOf(jaar);
System.out.println("Vakken van jaar " + jaar + ":");
for (int i = 0; i < leraar.getAantalVakken(); i++) {
    Vak vak = leraar.getVak(i);
    if (vak.getJaar().equals(jaarString)) {
        System.out.printf("\t%-13s%-25s%d\n", vak.getCode(),
vak.getNaam(), vak.getAantalStudiepunten());
    }
}
}
}
}
}

```