



Academiejaar 2015-2016 – 1 ^{ste} examenperiode (januari 2016) - 1 ^{ste} examenkans	
Opleiding, afstudeerrichting en jaar: 1 TIN / TILE Naam van het opleidingsonderdeel: OO Programmeren I (Eventueel) dOLOD / deexamen: Campus: Schoonmeersen Lector(en): L. Lewyllie, A. Pierreux, H. Roobrouck, S. Vandermeersch, L. Vuyge	Examendatum: 14/01/2016 Aanvangsuur examen: 9u15 REEKS A
Naam en voornaam student:	
Geboortedatum student:	Studentennummer:
Lector bij wie de student de onderwijsactiviteit volgde:	Lesgroep waarin de student de onderwijsactiviteit volgde:
Behaald resultaat: _____ op _____	

☐

Tijdens het examen mogen **GEEN** hulpmiddelen gebruikt worden:

☒

Tijdens het examen mogen onderstaande hulpmiddelen gebruikt worden:

- Java How To Program (Early objects) – 10e editie (of eerder) - Deitel & Deitel Prentice Hall

Algemene richtlijnen:

- Vul het bovenstaande kader aan.
- Vul op elke bladzijde je naam en voornaam in.
- Controleer of deze examenbundel alle pagina's bevat, zo niet verwittig de docent of de toezichter zodat je een nieuw exemplaar krijgt.
- Schrijf duidelijk, onleesbaar wil zeggen geen punten.
- Elke vorm van mondelinge, schriftelijke of elektronische (bijv. mobiele telefoon, smartphones, enz.) communicatie tijdens examens, tenzij deze uitdrukkelijk is toegelaten, worden beschouwd als "onregelmatigheid" en vallen onder de toepassing van OER art. 60 Examentucht.

Vraag 1	/ 12 p
Vraag 2	/ 15 p
Vraag 3	/ 13 p
Vraag 4	/ 30 p
TOTAAL	/ 70 p

Vraag 1 (12 punten):

Maak het schema voor $n = 4$ en bepaal de uitvoer:

```
public class Examen
{
    public static void main (String[] args)
    {
        System.out.println("4 = " + recursie(4));
    }

    public static int recursie (int n)
    {
        if (n == 0)
            return 3;
        else
            if (n == 1)
                return 4;
            else
                return recursie(n-2) + (n - 1) * recursie(n-1)/2;
    }
}
```

Schema?

$N = 2$

Return $\text{recursie}(0) + 1 * \text{recursie}(1) / 2 = 3 + 1 * 4 / 2 = 5$

$N = 3$

Return $\text{recursie}(1) + 2 * \text{recursie}(2) / 2 = 4 + 2 * 5 / 2 = 9$

$N = 4$

Return $\text{recursie}(2) + 3 * \text{recursie}(3) / 2 = 5 + 3 * 9 / 2 = 18,5$

Vraag 2 (15 punten)

Schrijf een programma dat een array van 20 gehele getallen random opvult met getallen tussen 1 en 10 (= MAXWAARDE). De methode main definieert de constante MAXWAARDE én creëert de array "getallen".

De methode **vulArray** zorgt voor de opvulling van de array "getallen".

De methode **verwerkArray** zal op basis van de constante MAXWAARDE en de gevulde array "getallen" de frequentie van elke waarde in de array "getallen" tussen 1 en MAXWAARDE bepalen en opslaan in een frequentiearray. Deze opgevulde array wordt teruggegeven aan main.

De methode **maakOverzicht** geeft een String terug met onderstaand overzicht. We tonen de inhoud van de frequentiearray in een tabel en we bepalen intussen ook hoeveel getallen even en hoeveel oneven zijn in de array "getallen":

run:

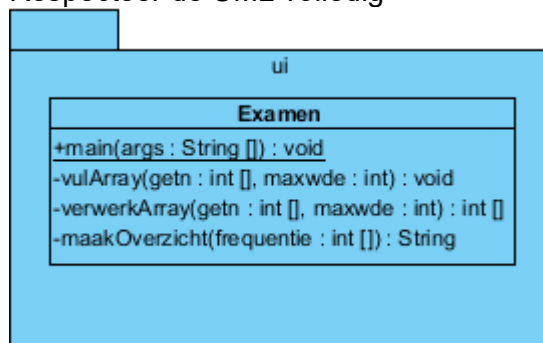
Waarde	Aantal
1	1
2	1
3	0
4	5
5	3
6	1
7	3
8	1
9	2
10	3

Aantal even = 11

Aantal oneven = 9

BUILD SUCCESSFUL (total time: 0 seconds)

Respecteer de UML volledig



én gebruik een enhanced for waar mogelijk!

```
public class Examen
{
    public static void main (String[] args){
        int[] getallen = new int[20];
        final int MAXWAARDE = 10;
        vulArray(getallen, MAXWAARDE);
        System.out.println(maakOverzicht(verwerkArray(getallen, MAXWAARDE)));
    }

    private static void vulArray (int[] getn, int maxwde) {
        Random random = new Random();
        for (int i = 0; i < getn.length; i++) {
            getn[i] = random.nextInt(maxwde);
        }
    }

    private static int[] verwerkArray (int[] getn, int maxwde) {
        int[] frequentieArray = new int[10];
        for (int i = 0; i < getn.length; i++) {
            frequentieArray[getn[i]] += 1;
        }
        return frequentieArray;
    }

    private static String maakOverzicht(int[] frequenties) {
        String overzicht;
        int even = 0;
        int oneven = 0;

        overzicht = String.format("Waarde Aantal %n");
        int i = 0;
        for (int frequentie: frequenties) {
            overzicht += String.format("%d %d%n",i+1,frequentie);
            i++;
            if (i % 2 == 0) {
                even += frequentie;
            } else {
                oneven += frequentie;
            }
        }
        overzicht += String.format("Aantal even %d%n", even);
        overzicht += String.format("Aantal oneven %d%n", oneven);
        return overzicht;
    }
}
```

Vraag 3 (13 punten):

Vul de code in de main-methode verder aan. Gebruik een enhanced for waar mogelijk!
Hieronder zie je de uitvoer van de applicatie!

```
run:
In de 2dim-array komt de waarde true 7 keer voor.
Er zijn 2 rijen met een even aantal kolommen.
Aantal vervangen waarden: 3
[[false, true, false], [true, false], [false, false, true, false], [false, false, true]]
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
package ui;
```

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
```

```
public class WaarValsApplicatie {
```

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

```
        boolean[][] waarden = {{true, true, false}, {true, false},
                                {true, true, true, false}, {false, false, true}};
```

```
        // Hoeveel keer zit de waarde true in deze 2Dim-array? Schrijf dit uit in een
        // zin (zie uitvoer)
```

```
        Int aantalTrue = 0;

        for (int i = 0; i < waarden.length; i++){
            for (int j = 0; j < waarden[i].length; j++){
                if (waarden[i][j] == true) {
                    aantalTrue++;
                }
            }
        }

        System.out.println("In de 2dim-array komt de waarde true %s keer voor",aantalTrue);
```

**//Verzamel de rij-nummers met een even aantal kolommen in een ArrayList.
//Schrijf dit aantal rijen uit. Let op enkelvoud/meervoud in deze zin!**

```
ArrayList<int> aantalEven = new ArrayList<>();

for (int i = 0; i<waarden.length; i++){
    if (waarden[i].length % 2 == 0 ) {
        aantalEven.add(i);
    }
}

System.out.println("Er zijn %s rijen met een even aantal kolommen",aantalEven.size())
```

**// Elke waarde true die op dezelfde rij onmiddellijk gevolgd wordt
// door een true, moet vervangen worden door een false. Houd het aantal
// vervangingen bij. Druk vervolgens dit aantal af én de nieuwe inhoud van de
// tabel – zie uitvoer**

```
int aantalVervangingen = 0;

for (int i = 0; i<waarden.length; i++){
    for (int j = 0; j<waarden[i].length; j++){
        if ( j != 0) {
            boolean vorigeWaarde = waarden[i] [j - 1]
        }
        if (vorigeWaarde == true && vorigeWaarde = waarden[i][j]) {
            waarden[i][j] = false;
            aantalVervangingen++;
        }
    }
}
```

```
System.out.println("Aantal vervangen waardes: %s",aantalVervangingen);
```

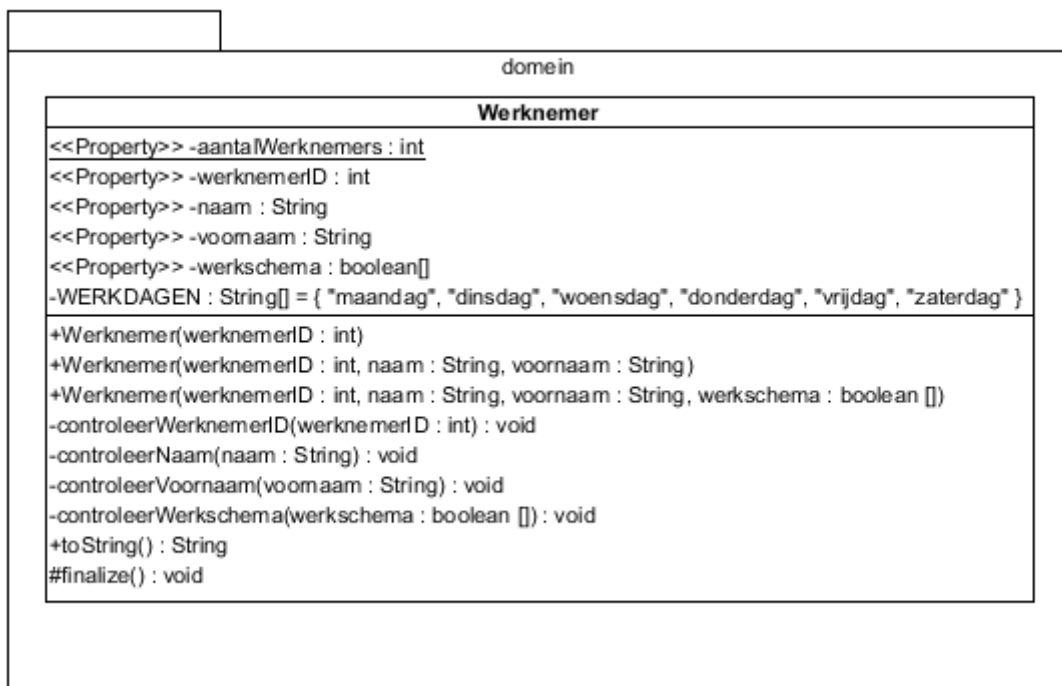
```
//Nieuwe array word al afgeprint hieronder
```

```
System.out.println(Arrays.deepToString(waarden));
}
```

Vraag 4 (30 punten).

Vervolledig onderstaande domeinklasse en applicatieklasse (gebruik een enhanced for waar mogelijk!):

- a) In de domeinklasse `Werknemer` wordt van elke werknemer zijn `werknemerID`, naam, voornaam en werkschema bijgehouden. De dagen waarop gewerkt wordt in het bedrijf, worden bijgehouden in het attribuut `WERKDAGEN`. De dagen waarop de werknemer werkt, worden bijgehouden in het attribuut `werkschema` (`false`=werkt niet en `true`=werkt). We houden ook het totaal aantal werknemers bij in het attribuut `'aantalWerknemers'`. Zorg ervoor dat dit steeds het juiste aantal bevat!



Voorzie in de domeinklasse 3 constructoren :

- Een constructor met één parameter : het `werknemerID`. Naam en voornaam zijn “onbekend” en voor het werkschema staan alle dagen op `false`.
- Een constructor met drie parameters: `werknemerID`, naam en voornaam. Voor het werkschema staan alle dagen op `false`.
- Een constructor met 4 parameters : `werknemerID`, naam, voornaam en werkschema.
- Schrijf één constructor volledig uit, en roep deze aan in de andere constructors.
- Bij de creatie van `Werknemer`-objecten en de toekenning van waarden aan de attributen in de setters gelden de volgende controles:
 - `werknemerID` is een getal tussen 0 en 3000 (grenzen niet inbegrepen)
 - naam en voornaam zijn niet null en ook niet gelijk aan de lege string
 - werkschema moet naar een array refereren én het aantal elementen in werkschema moet 6 zijn

```
public class Werknemer
{
    private int werknemerID;
    private String naam, voornaam;
    private boolean[] werkschema;

    private final String[] WERKDAGEN =
        {"maandag", "dinsdag", "woensdag", "donderdag", "vrijdag", "zaterdag"};
```

```
//extra attribuut
```

```
private static int aantalWerknemers;
```

```
// 3 constructoren
```

```
public Werknemer(int werknemerID, String naam, String voornaam, boolean[] werkschema) {
    controleerWerknemerID(werknemerID);
    this.werknemerID = werknemerID;
    controleerVoornaam(voornaam);
    this.voornaam = voornaam;
    controleerNaam(naam);
    this.naam = naam;
    controleerWerkschema(werkschema);
    this.werkschema = werkschema;
    aantalWerknemers++;
}
```

```
Public Werknemer(int werknemerID) {
    this(werknemerID, "onbekend", "onbekend", [false, false, false, false, false, false]);
}
```

```
Public Werknemer(int werknemerID, String naam, String voornaam)
    this(werknemerID, naam, voornaam, [false, false, false, false, false, false]);
}
```


/*Vervolledig onderstaande controlemethodes. Zie voorwaarden hierboven.
* Indien de meegegeven waarden niet voldoen, dan wordt een IllegalArgumentException met
* passende foutboodschap gegoooid.
*/

private void controleerWerknemerID(int werknemerID)

```
if (werknemerID < 0 || werknemerID > 3000) {  
    throw new IllegalArgumentException();  
}
```

private void controleerNaam(String naam)

```
if (naam == null || naam.isEmpty()) {  
    throw new IllegalArgumentException();  
}
```

private void controleerVoornaam(String voornaam)

```
if (voornaam == null || voornaam.isEmpty()) {  
    throw new IllegalArgumentException();  
}
```

private void controleerWerkschema(boolean[] werkschema)

```
if (!werkschema instanceof boolean[] || werkschema.length != 6)  
    throw new IllegalArgumentException();  
}
```

Vul hieronder je naam en voornaam in:

/* Vervolledig de volgende setter.*/

```
public void setWerknemerID (int werknemerID)
{
    controleerWerknemerID(werknemerID);
    this.werknemerID = werknemerID;
}
```

//Overige set- en getmethodes.

/* Vervolledig de methode toString(). Als resultaat geeft deze de naam, voornaam, werknemerID en de dagen weer waarop de werknemer werkt ahv de tabel WERKDAGEN; plaats een tab tussen elke werkdag (zie ook voorbeeld uitvoer WerknemerApplicatie).



OF

```
    Janssens Jan met ID : 104
    werkt NIET!
*/
```

/*Tot slot willen we ook het aantal reeds gecreëerde Werknemer-objecten gaan bijhouden. Definieer daartoe een extra attribuut **op pagina 8**. Zorg ook voor de nodige extra code zodanig dat het attribuut op de juiste manier wordt aangepast (Bij creatie / bij verwijdering van een Werknemer-object). Tevens voorzie je een get-methode om de waarde van het attribuut op te vragen.*/

```
// Extra code
protected void finalize() {
    aantalWerknemers--;
}

public static int getAantalWerknemers() {
    return aantalWerknemers;
}
```

```
}/einde domeinklasse Werknemer
```

- b) In de applicatieklasse `WerknemerApplicatie` wordt gestart met het tonen van het aantal werknemers.
 Daarna worden 5 `Werknemer`-objecten aangemaakt. Daarbij wordt gebruik gemaakt van de ids, namen en voornamen die gegeven zijn. De 5 `Werknemer`-objecten worden bijgehouden in een array.
 Enkel het werkschema voor de eerste werknemer is gekend en wordt dus ook toegekend aan het `Werknemer`-object.
 Na creatie van de `Werknemer`-objecten wordt terug het aantal werknemers getoond.

Voorbeeld uitvoer:

```
Aantal werknemers : 0

Peters Piet met ID : 102
werkt op: dinsdag        woensdag        donderdag        zaterdag

Janssens Jan met ID : 104
werkt NIET!

Pieters Peter met ID : 562
werkt NIET!

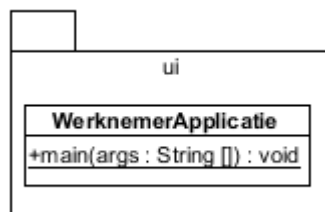
De Zeure Tine met ID : 256
werkt NIET!

Boni Alain met ID : 789
werkt NIET!

Aantal werknemers : 5
BUILD SUCCESSFUL (total time: 0 seconds)
|
```

Deze uitvoer is gebaseerd op de volgende gegevens:

```
int[] ids = {102, 104, 562, 256, 789 };
String[] namen = {"Peters", "Janssens", "Pieters", "De Zeure", "Boni"};
String[] voornamen = {"Piet", "Jan", "Peter", "Tine", "Alain"};
boolean[] werkschemaPeters = {false, true, true, true, false, true};
```



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

```
// Tonen van het aantal werknemers
```

```
    System.out.println("Aantal werknemers : %s",getAantalWerknemers());
```

```
    int[] ids ={102, 104, 562, 256, 789 };
    String[] namen={"Peters","Janssens","Pieters","De Zeure","Boni"};
    String[] voornamen={"Piet","Jan","Peter","Tine","Alain"};
```

```
    boolean[] werkschemaPeters={false,true,true,true,false,true};
```

```
// Creatie Werknemer-objecten
```

```
Werknemer[] werknemers = new Werknemer[5];
werknemers[0] = new Werknemer(ids[0],namen[0],voornamen[0],werkschemaPeters);
for (int i = 1; namen.length ; i++) {
    werknemers[i] = new Werknemer(ids[i],namen[i],voornamen[i])
}
```

```
// Afdrukken van de Werknemer-gegevens
```

```
for (Werknemer werknemer : werknemers) {
    System.out.printf("%s%n%n",werknemer.toString());
}
```

```
// Tonen van het aantal werknemers
```

```
    System.out.println("Aantal werknemers : %s",getAantalWerknemers());
```

```
    }
}
```