

**Hogeschool PXL**

Departement IT  
Academiejaar 2017-2018

## EXAMEN

<b>Vak: Java Essentials</b>	<b>(deel 1 : schriftelijk gesloten boek)</b>
Resultaat	/ 29 ➡ / 8
Periode	1 <sup>ste</sup> zit
Datum	29/1/2018
Tijdstip	
Klassen	1TIN A – B – C – D – E – F – G – H – I – J – K – L
Lectoren	N. Custers – R. Krekels – H. Tans – S. Vanderstraeten - I. Vanherwegen – J. Willekens

Studentengegevens	
Naam student	
Voornaam student	
Klas	
Lector	
Lokaal	

Samenstelling bundel		
Onderdelen (*)	<b>Deel 1</b>	Deel 2
Inhoud	<b>theorie</b>	praktijk
Pagina's	9 p	13 p
Puntenverdeling	40%	60%
Digitaal beginbestand		X
Digitale indiening		X
Toegelaten hulpmiddelen:		
* rekenmachine		
* laptop		X
* internet		
* cursusmateriaal		X
Opmerkingen:	Dit deel duurt maximaal tot 9u45. Na afgifte van deel1, krijg je deel2	Studenten mogen gebruik maken van alle materiaal dat op de laptop aanwezig is alsook de cursus: Programming Basics Java Essentials PXL versie en 1 ringmap met oefeningenbundel + oplossingen.

*Elke student(e) is verantwoordelijk voor de correcte samenstelling van zijn/haar bundeltje. Eventuele afwijkingen moeten onmiddellijk aan de toezichthouder gemeld worden.*

Aanvangsuur examen: 8:30u

Einde examen: 12u



## Checklist voor het afleggen van examen

- ☐ GSM's/smartphones liggen uitgeschakeld op tafel
- ☐ Studentenkaart ter beschikking houden
- ☐ Jassen en tassen vooraan in het lokaal
- ☐ GEEN GSM's of smartphones in jassen of tassen
- ☐ Bij laptopexamen strikt de opgelegde regels volgen
- ☐ Examenbundels blijven steeds samengeniet
- ☐ Toegestaan: 1 droog koekje en 1 drankje in hersluitbaar flesje
- ☐ Iedereen zwijgt tijdens het examen
- ☐ Afgeven mag ten vroegste een half uur na aanvang van het examen.
- ☐ Niet vergeten op je examenkopij te vermelden:

**Naam van de lector**



## Vraag 2 (6 punten)

Gegeven volgende informatie bekomen uit de documentatie van de klasse `LocalTime`.

### of

```
public static LocalTime of(int hour,  
                           int minute)
```

Obtains an instance of `LocalTime` from an hour and minute.

This returns a `LocalTime` with the specified hour and minute. The second and nanosecond fields will be set to zero.

#### Parameters:

hour - the hour-of-day to represent, from 0 to 23

minute - the minute-of-hour to represent, from 0 to 59

#### Returns:

the local time, not null

#### Throws:

`DateTimeException` - if the value of any field is out of range

### format

```
public String format(DateTimeFormatter formatter)
```

Formats this time using the specified formatter.

This time will be passed to the formatter to produce a string.

#### Parameters:

formatter - the formatter to use, not null

#### Returns:

the formatted time string, not null

#### Throws:

`DateTimeException` - if an error occurs during printing

Vervolledig onderstaande code. Je gebruikt de variabelen uren en minuten om een object van de klasse `LocalTime` aan te maken. Je mag ervan uitgaan dat de gebruiker gehele getallen heeft ingegeven. Je laat de variabele tijd naar dit nieuwe object verwijzen. Indien de creatie van het object goed verloopt, wordt de waarde van de variabele `correcteIngave` true. Indien er iets mis gaat bij de creatie van het object (en er dus een `DateTimeException` wordt opgeworpen), geef je de gebruiker de foutmelding “foutieve ingave” en zorg je ervoor dat de gebruiker een nieuwe ingave kan doen. Uiteindelijk, als er een correcte ingave wordt gedaan, zorg je er nog voor dat je de formatter gebruikt om de waarde van referentie-variabele `tijd` af te drukken op het scherm.

```

public class Vraag2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        boolean correcteIngave = false;
        LocalTime tijd = null;
        // conditie van de while-lus aanvullen!

        while (.....) {

            System.out.println("Geef uren:");
            int uren = input.nextInt();
            System.out.println("Geef minuten:");
            int minuten = input.nextInt();
            // gebruik uren en minuten om een object van de klasse
            // LocalTime aan te maken en toe te kennen aan de variabele tijd
            // als dit mislukt toon je de foutmelding "foutieve ingave" en
            // zorg je ervoor dat de gebruiker een nieuwe ingave kan doen

        }

        // toon hier de tijd op het scherm gebruik maken van formatter

        input.close();
    }
}

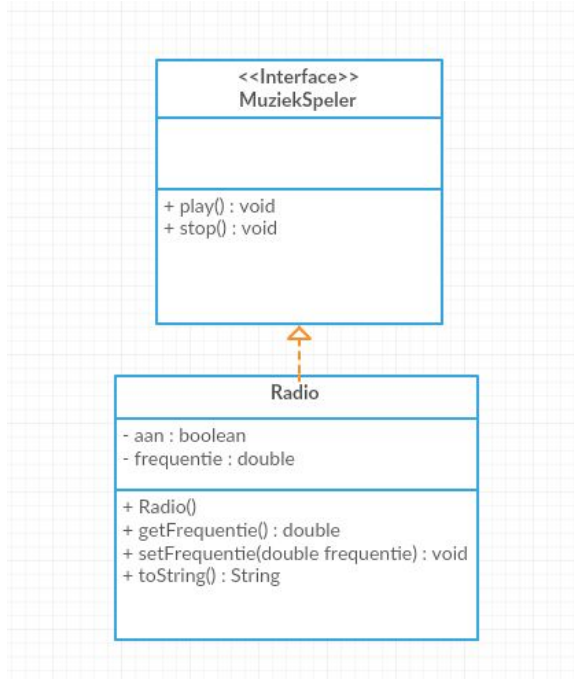
```



### Vraag 3 (6 punten)

Gegeven de interface MuziekSpeler.

```
public interface MuziekSpeler {  
    void play();  
    void stop();  
}
```



Maak nu een klasse `Radio` volgens bovenstaand klassendiagram. Voor frequentie zijn waarden van 88 tot 108 Mhz toegelaten. Indien in een setter een foutieve waarde wordt gegeven als parameter werp je een `IllegalArgumentException` op.

java.lang

#### Class `IllegalArgumentException`

```
java.lang.Object  
  java.lang.Throwable  
    java.lang.Exception  
      java.lang.RuntimeException  
        java.lang.IllegalArgumentException
```

## Constructor Summary

### Constructors

#### Constructor and Description

`IllegalArgumentException()`

Constructs an `IllegalArgumentException` with no detail message.

`IllegalArgumentException(String s)`

Constructs an `IllegalArgumentException` with the specified detail message.

`IllegalArgumentException(String message, Throwable cause)`

Constructs a new exception with the specified detail message and cause.


`IllegalArgumentException(Throwable cause)`

Constructs a new exception with the specified cause and a detail message of `(cause==null ? null : cause.toString())` (which typically contains the class and detail message of cause).

Voorzie in de klasse `Radio` de methode `toString()` die een tekst met de frequentie en de status van de radio (aan of uit) geeft.

De default constructor maakt een `Radio`-object aan met frequentie waarde 97,9 en status uit. Wanneer je voor de radio de methode `play()` aanroept zal de status aan worden, bij de methode `stop()` wordt de status uit.

**Klasse `Radio`:**

	<b>Hogeschool PXL</b> Departement IT Academiejaar 2016-2017  EXAMEN
---	---

Vak: Java Essentials	(deel 2 : praktijk deel: laptop)
Resultaat	/ 12
Periode	1 <sup>ste</sup> zit
Datum	29/1/2018
Tijdstip	
Klassen	1TIN A – B – C – D – E – F – G – H – I – J – K – L
Lectoren	N. Custers – R. Krekels – H. Tans – S. Vanderstraeten - I. Vanherwegen – J. Willekens

Studentengegevens	
Naam student	
Voornaam student	
Klas	
Lector	
Lokaal	

Samenstelling bundel		
Onderdelen (*)	Deel 1	<b>Deel 2</b>
Inhoud	theorie	<b>praktijk</b>
Pagina's	9 p	13 p
Puntenverdeling	40%	60%
Digitaal beginbestand		X
Digitale indiening		X
Toegelaten hulpmiddelen:		
* rekenmachine		
* laptop		X
* internet		
* cursusmateriaal		X
Opmerkingen:	Dit deel duurt maximaal tot 9:45u. Na afgifte van deel1, krijg je deel2	Studenten mogen gebruik maken van alle materiaal dat op de laptop aanwezig is alsook de cursus Programming Basics Java Essentials PXL versie en 1 ringmap met oefeningenbundel + oplossingen.

*Elke student(e) is verantwoordelijk voor de correcte samenstelling van zijn/haar bundeltje.  
Eventuele afwijkingen moeten onmiddellijk aan de toezichthouder gesignaleerd worden.*

MD5Hash:

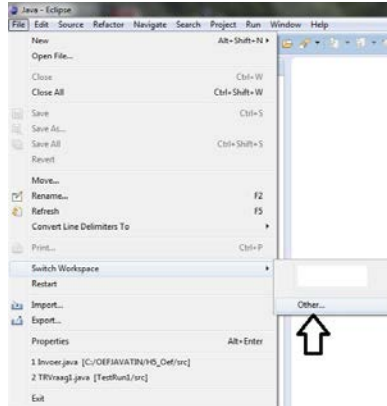
Aanvangsuur examen: 8:30u      Einde examen: 12u



## STAPPEN UIT TE VOEREN VOOR JE AAN HET LAPTOPGEDEELTE BEGINT

**!!! Voordat je begint zorg je ervoor dat je laptop in AIRPLANE mode is gezet !!!**

Als voorbereiding voor het examen dien je eerst een nieuwe, lege workspace aan te maken. Je doet dit door in het menu te kiezen voor *Switch Workspace* en vervolgens *Other*.



In de Workspace Launcher geef je als Workspace `C:\<Naam>_<Voornaam>` in.

Bv: voor de student met naam Jan Peeters wordt de workspace gegeven door `C:\Peeters_Jan`.

Eclipse zal nu opnieuw opstarten in de nieuwe workspace. Je Eclipse venster zou nu leeg moeten zijn, m.a.w. het Package Explorer venster toont geen mappen of files.

Maak een java project met als naam `<familienaam_voornaam_JE_zit1>`.

Bv: een student met naam Jan Peeters maakt een project met naam *peeters\_jan\_JE\_zit1*.

## STAPPEN UIT TE VOEREN WANNEER JE KLAAR BENT MET HET EXAMEN

Wanneer je klaar bent met het oplossen van de vragen, sluit je Eclipse volledig af. **VERGEET NIET EERST JE BESTANDEN TE SAVEN!!!**

Ga vervolgens via je Windows Explorer naar de map `C:\<Naam>_<Voornaam>` die je voor dit examen hebt aangemaakt.

Je moet nu deze map zippen. Het makkelijkst kun je dit doen door te rechtsklikken op de folder, vervolgens te kiezen voor 'Kopiëren naar' en tot slot 'Gecomprimeerde (gezipte) map'. Plaats de zip-file op je bureaublad.

Je zip-file moet als naam `<Naam>_<Voornaam>.zip` hebben (bijvoorbeeld `Peeters_Jan.zip`).

Upload je bestand m.b.v. de handleiding FileZilla.

## Vraag 1: (25 punten)

1001 0110	5%
--------------	----

{ CC }	2% max 4%
--------	--------------

✓ SPEC	2% max 4%
--------	--------------

Maak voor dit project gebruik van de meegeleverde klasse **Data** en de klasse **WinkelApp**. Maak volgende klassen aan, lees aandachtig de opgave! Er mogen geen bijkomende variabelen aangemaakt worden. Vermijd DUBBELE CODE!

### Klasse Product

Maak een abstracte klasse Product die de volgende variabelen bevat:

- productNummer: int
- merk: string
- naam: string
- volume: int
- prijs: double (dit is de prijs inclusief btw)

Voorzie:

- Een **constructor** die alle eigenschappen initialiseert.
- De nodige **getters en setters**
- Een methode **“getProductCode()”** die een String teruggeeft.

De code bestaat uit:

- o De eerste 3 letters van het merk;
- o De eerste 3 letters van de naam;
- o De volume waarde.

Bovendien wordt alles in hoofdletters gezet en worden eventuele spaties door een underscore vervangen.

Je mag ervan uitgaan dat merk en naam steeds meer dan 3 karakters bevatten.

Voorbeelden:

*Merk: Chanel Naam: Coco Mademoiselle Volume: 100ml => CHACOC100*

*Merk: DKNY Naam: Be Delicious Woman Volume: 100ml => DKNBE\_100*

- Override de **toString()** methode: Kijk voor een goede implementatie van deze methode even naar de printscreen op pagina 5. Maak gebruik van de **FORMAT** methode!

- Override de **equals()** methode zodat 2 producten gelijk zijn als hun merk, naam en volume gelijk zijn.

**De enum DeoType** met de constanten VAPO en STICK.

### **De klassen AfterShave, Deodorant, en Parfum**

Implementeer de klassen:

- Laat de klassen AfterShave, Deodorant en Parfum **overerven** van de abstracte klasse Product.
- AfterShave en Deodorant hebben een instance-variabele 'soort' van het type DeoType.
- In de klasse Parfum hebben we geen specifieke variabelen nodig.
- Schrijf de nodige constructors in de klasse AfterShave, Deodorant en Parfum.
- Voorzie, indien nodig, een overriding van de toString() methode van de superklasse.

### **De Interface Berekenbaar**

Maak een interface Berekenbaar met daarin:

- Een constante BTW met waarde 21
- de methode **"totalePrijs()"** met resultaat-type double, dit geeft de totale prijs inclusief btw.
- Een standaard (of default) methode **"totalePrijsExclBtw()"** die gebruik maakt van de methode **"totalePrijs()"** en de constante BTW om de prijs exclusief btw te tonen.

### **De klasse Bestelling**

Implementeer de klasse Bestelling. Test elke implementatie stap voor stap aan de hand van de meegeleverde testklasse WinkelApp waarvoor de testdata gedefinieerd is in de klasse Data.

In de klasse Data zijn alle productnummers 0. Wanneer een product aan de bestelling wordt toegevoegd, wordt aan het productNummer een waarde toegekend.

Voorzie de volgende members in de klasse bestelling:

- bestelling: array waar 100 producten in kunnen
- Een klasse variabele productNummer : int met startwaarde 1000
- Voorzie de nodige constructors
- Schrijf de methode **"voegProductToe()"** die een Product-object als parameter heeft om aan de bestelling toe te voegen. Zorg er voor dat het productnummer met 1000 start en met één verhoogd wordt voor elk volgende product dat wordt toegevoegd. Je mag

een product niet toevoegen indien er reeds een identiek product aan de bestelling werd toegevoegd (identiek betekent dat het merk, de naam en het volume gelijk zijn). Zorg voor een foutmelding indien een product niet toegevoegd kan worden.

- Schrijf de methode “**verwijderProduct()**” die een productNummer als parameter heeft en het Product-object met het overeenkomstig productNummer uit de bestelling verwijdert. Geef een gepaste foutmelding als er geen Product-object wordt gevonden. Wanneer je het Product-object hebt verwijderd geef je ook een gepaste melding (zie uitvoer van het programma).
- Schrijf een methode “**toonLijst()**” die alle producten uit de bestelling afdrukt. Je ziet de gewenste uitvoer in het test programma.
- Schrijf een methode “**toonPerMerk()**” die een parameter merk als string verwacht waarmee je alleen het product van een bepaald merk uit de bestelling afdrukt.
- Schrijf een methode “**toonParfums()**” om alle parfums uit de lijst bestelling te tonen.
- Schrijf een methode “**getAantalProducten()**” die het totaal aantal producten in de bestelling teruggeeft.
- Implementeer de interface Berekenbaar. Zorg ervoor dat de methode “**totalePrijs()**” de prijs van alle producten uit de winkelmand met elkaar optelt.

```

<terminated> WinkelApp [Java Application] C:\Program Files (x86)\Java\jre1.8.0_131\bin\javaw.exe (26 jun. 2017 10:51:08)
Oplossing van Jouv naam Student PXL
Product met productnummer 1003 werd verwijderd.
Het product is reeds toegevoegd.
1000 Merk: Dolce & Gabbana      Naam: Light Blue      Volume: 100ml Prijs: 66,72 Code: DOLLIG100
1001 Merk: BVLGARI             Naam: BLV             Volume: 75ml Prijs: 61,52 Code: BVLBLV75
1002 Merk: DKNY                Naam: Be Delicious Women Volume: 100ml Prijs: 33,65 Code: DKNBE_100 STICK
1004 Merk: Yves Saint Laurent Naam: Jazz            Volume: 50ml Prijs: 39,84 Code: YVEJAZ50  VAPO
1005 Merk: Givency             Naam: Absolutely Irresistible Volume: 75ml Prijs: 75,42 Code: GIVABS75
1006 Merk: Yves Saint Laurent Naam: Jazz            Volume: 100ml Prijs: 57,76 Code: YVEJAZ100 VAPO
1007 Merk: Ted Lapidus         Naam: Pour Elle       Volume: 50ml Prijs: 44,48 Code: TEDPOU50
1008 Merk: Giorgio Armani      Naam: Code Donna      Volume: 30ml Prijs: 39,84 Code: GIOCOD30
1009 Merk: Giorgio Armani      Naam: Code Donna      Volume: 75ml Prijs: 76,00 Code: GIOCOD75
1010 Merk: Cacharel           Naam: Anaïs           Volume: 50ml Prijs: 24,50 Code: CACANA50  VAPO

Van het merk Giorgio Armani:
1008 Merk: Giorgio Armani      Naam: Code Donna      Volume: 30ml Prijs: 39,84 Code: GIOCOD30
1009 Merk: Giorgio Armani      Naam: Code Donna      Volume: 75ml Prijs: 76,00 Code: GIOCOD75

Alle Parfums:
1000 Merk: Dolce & Gabbana      Naam: Light Blue      Volume: 100ml Prijs: 66,72 Code: DOLLIG100
1001 Merk: BVLGARI             Naam: BLV             Volume: 75ml Prijs: 61,52 Code: BVLBLV75
1005 Merk: Givency             Naam: Absolutely Irresistible Volume: 75ml Prijs: 75,42 Code: GIVABS75
1007 Merk: Ted Lapidus         Naam: Pour Elle       Volume: 50ml Prijs: 44,48 Code: TEDPOU50
1008 Merk: Giorgio Armani      Naam: Code Donna      Volume: 30ml Prijs: 39,84 Code: GIOCOD30
1009 Merk: Giorgio Armani      Naam: Code Donna      Volume: 75ml Prijs: 76,00 Code: GIOCOD75

Totale prijs: €519,73
Totale prijs excl. btw: €429,53
Aantal producten in bestelling: 10

```