

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, 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

        .....

        .....

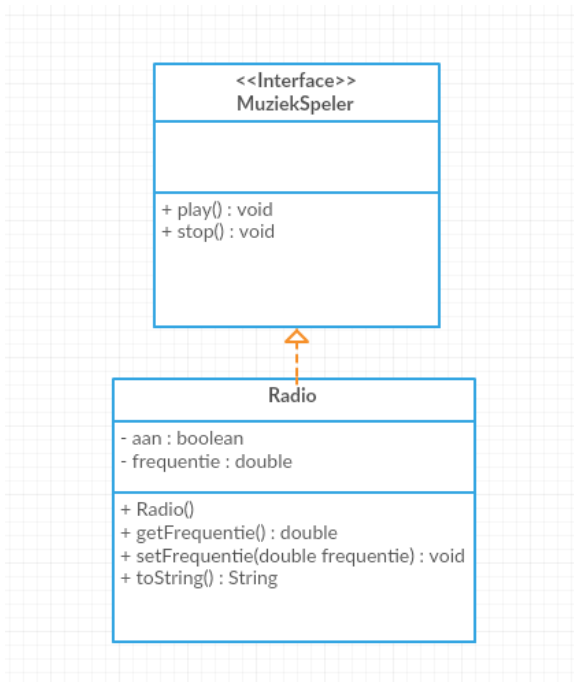
        input.close();
    }
}

```

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.

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:

Schrijf nu een programma waarbij je gebruik maakt van polymorfisme met interfaces.
Print de gegevens van de radio af (door gebruik te maken van de methode toString()).
Zet de radio aan.
Print opnieuw de gegevens van de radio af.
Verander de frequentie naar 107.9.
Print de gegevens van de radio opnieuw af.

Je bekomt dan volgende output:

Radio: uit

Radio: aan
Frequentie: 97.9

Radio: aan
Frequentie: 107.9

```
public class MuziekApp {  
    public static void main(String[] args) {  
        MuziekSpeler radio = new Radio();  
        // ontbrekende code hier:  
        // print de gegevens van radio  
  
        .....  
        // zet de radio aan  
  
        .....  
        // print de gegevens van de radio  
  
        .....  
        // pas de frequentie van de radio aan naar 107.9  
  
        .....  
        // print de gegevens van de radio  
  
        .....  
    }  
}
```

Voor je aan het examen begint haal je via FileZilla enkele bestanden binnen voor beide vragen:

- Vraag 1: klassen Data en WinkelApp
- Vraag 2: klasse TandartsApp

In je oplossingen moeten deze bestanden, zonder aanpassing, gebruikt worden.

Plaats bovenaan in elke file je naam in commentaar.

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

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

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

Vraag 1: (25 punten)

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 afdruckt. 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 afdruckt.
- 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.

```

Problems @ Javadoc Declaration Search Console
<terminated> WinkelApp [Java Application] C:\Program Files (x86)\Java\jre1.8.0_131\bin\javaw.exe (26 jun. 2017 10:51:08)
Oplossing van Jouw 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

```