

**(X) Voorblad in te leveren door student? Zo ja (X), dan volgende blok invullen.**

Studentgegevens			
<b>Naam student:</b>			
<b>Studentnummer:</b>		<b>Klas:</b>	
<b>Kruis hiernaast naam van de eigen vakdocent aan:</b>	<input type="checkbox"/>	Joost Visser	<input type="checkbox"/>
	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>		Ik weet het niet/ik volgde geen les

**Vermeld op ieder blad je naam, studentnummer en klas.**

**VEEL SUCCES!**

Algemene toetsgegevens			
<b>Academie</b>	AVD	Fase	Hoofdphase
<b>Opleiding</b>	Informatica	Onderwijsvorm	Deeltijd
Naam vak	Algoritmen en Datastructuren	Vakcode (studievolg)	DTIN-ALGDAT-19
Subtitel		Auteur(s)	Joost Visser
Datum	08-06-2021	Review	Patrick ubags
Tijd	19:50 – 21:30 uur		
Aantal pagina's inclusief voorblad	6	Aantal opgaven	6
Bijzonderheden			
Papier: (kruis aan wat van toepassing is)	<input type="checkbox"/> Lijntjes A4	<input checked="" type="checkbox"/> Kladpapier	<input type="checkbox"/> Ruitje A4
	<input type="checkbox"/> Lijntjes A3	<input type="checkbox"/> Transparant	<input type="checkbox"/> Anders nl:
Opgaven inleveren?	<b>Ja</b>		
Rekenmachine	<b>Nee</b>		
Toegestane hulpmiddelen:	<input type="checkbox"/> Geen hulpmiddelen toegestaan		
	<input checked="" type="checkbox"/> Boeken: <i>Algoritmen en Datastructuren</i> , <i>Veerle Fack</i>		
	<input checked="" type="checkbox"/>	Anders nl: <ul style="list-style-type: none"> <li>eigen laptop</li> <li>uitgewerkte practicumopgaven</li> <li>sheets op Brightspace behorende bij de cursus Algoritmen &amp; Datastructuren</li> <li>het Javaproject Zoo van deze toets</li> <li>de website <a href="https://docs.oracle.com">https://docs.oracle.com</a></li> </ul>	

Opmerkingen	<p>Er mag geen gebruik worden gemaakt van internet (google, chat, e-mail, dropbox, etc.), op de volgende bovengenoemde website na.</p> <p>Lever de uitwerkingen in via de Brightspace-inleverlink in de cursus AVD-I-Advanced programming onder Beoordeling leeruitkomsten.</p>
Contactgegevens	<p>Academbureau AVD; kamer HG304</p> <p>Contactpersoon: <b>Joost Visser</b></p>

Elke student wordt geacht de bepalingen m.b.t. het afleggen van de toetsen te kennen.

## Tijd

Dit tentamen bestaat uit een theorie- en een praktijkgedeelte. De geschatte benodigde tijd is maximaal 30 minuten voor het theoriegedeelte en 120 minuten voor het praktijkgedeelte

## Inleveren

Lever de uitwerkingen in via de Brightspace-inleverlink in de cursus AVD-I-Advanced programming onder Beoordeling leeruitkomsten

Stuur **een ZIP**-bestand met daarin:

- een tekstdocument (.txt, .docx of .pdf) met de uitwerking van het theoriegedeelte
- de complete projecten voor het praktijkgedeelte.

## Gegeven source-code

Bij de Blackboard-opdracht staat de Java-code die je nodig hebt voor opdracht 6

## THEORIEGEDEELTE (max. 50 punten)

### Opgave 1 – Complexiteit (max. 10 punten)

#### 1.1

Geef de rekencomplexiteit **én** de geheugencomplexiteit van de volgende **recursieve** methode. Onderbouw je antwoorden. Gebruik de  $O()$ -notatie voor complexiteit.

```
double f (double x) {  
    if (x <= 0)  
        return 1.0;  
    return x * (f(x-1) + f(x-2)); }  
}
```

#### 1.2

Geef de rekencomplexiteit **én** de geheugencomplexiteit van de volgende methode. Onderbouw je antwoorden. Gebruik de  $O()$ -notatie voor complexiteit. Bestudeer de code goed.

```
int f (int n) {  
    int t = 0;  
    for(int i = 0; i < n; i++) {  
        for(int j = 0; j < 10; j++) {  
            t += n;  
        }  
    }  
    return t;  
}
```

### Opgave 2 – Arrays (max. 10 punten)

Stukje Java:

```
int[] anArray = new int[100];  
ArrayList<Integer> aList = new ArrayList<>();
```

#### 2.1

Beschrijf een situatie waarin je het best voor anArray kunt kiezen, en één waarin aList efficiënter is. Onderbouw je keuzes.

### Opgave 3 - Hash (max. 10 punten)

#### 3.1

Wat is de returnwaarde van de default Hash-methode van de klasse Object in Java?

#### 3.2

Waarom is deze niet goed bruikbaar?

#### 3.3

Wat is de rekencomplexiteit van de methode() containsKey van de klasse HashMap? Verklaar je antwoord.

### Opgave 4 - Generics (max. 10 punten)

Bekijk de volgende declaraties:

A) `public Shape handleList(List<Shape> shapes);`

B) `public <T extends Shape> T handleList(List<T> shapes);`

C) `public Shape handleList(List<? extends Shape> shapes);`

`public class Circle extends Shape { ... }`

#### 4.1

Met welke van de drie kun je een lijst van het type List<Circle> verwerken? (meerdere antwoorden mogelijk). Leg per klasse uit waarom wel of waarom niet.

#### 4.2

Leg het verschil uit tussen de declaraties B en C.

### Opgave 5 - Bomen (max. 10 punten)

#### 5.1

Beschrijf in woorden hoe je de diepte van een boom kunt bepalen. Schrijf eventueel voorbeeldcode ter verduidelijking.

## PRAKTIJKGEDEELTE

### Opdracht 6 – Zoo (max. 50 punten)

Begin met het project 'Zoo'. Je vindt de code bij de inleverlink op Blackboard.

Hierin worden dieren en voedertijden (AnimalFeeding) gekoppeld aan een dierenverzorger (ZooKeeper).

Er zijn twee verschillende overzichten in de klasse ZooKeeper:

- 1) printAnimals() laat per regel een diersoort zien, gevolgd door de voedertijd
- 2) printSchedule() laat per regel een voedertijd zien, gevolgd door een diersoort

Doordat AnimalFeedings in ZooKeeper in een ArrayList staan, zijn de twee overzichten nu gesorteerd op volgorde van invoer. Die volgorde moet anders!

Voor de volgende opgaven geldt:

- Pas de klasse ZooKeeper aan waar nodig.
- *De klassen Zoo en AnimalFeeding moeten ongewijzigd blijven!*
- Het is niet de bedoeling dat je sorteermethodes maakt of aanroept. De keuze voor de juiste containerklassen moet voor impliciete sortering zorgen. - **Kijk naar de gewenste uitvoer op de volgende bladzijde!**

#### 6.1 (max. 20 punten)

De methode printAnimals() van ZooKeeper moet de regels op alfabetische volgorde van dierennaam gaan weergeven.

- Kies daarom een andere containerklasse voor feedings, zodat het overzicht printAnimals() in de juiste volgorde komt te staan.

#### 6.2 (max. 20 punten)

De methode printSchedule() van ZooKeeper moet de regels op volgorde van tijd gaan tonen.

- Je kunt hiervoor een **extra** container feedingsByTime toevoegen, zodat het overzicht printSchedule() de regels op de juiste volgorde weergeeft.
- In addAnimalFeeding moet je elk item dan ook aan feedingsByTime toevoegen.

#### 6.3 (max. 10 punten)

Zorg ervoor dat het WEL mogelijk is om een groep dieren meerdere malen per dag te voederen.  
Zorg ervoor dat het NIET mogelijk is om op dezelfde tijd meer dan één groep dieren te voederen.

### ***Gewenste uitvoer van opgave 6***

===== Eerste  
schema

=====

Dieren gevoerd door John  
IJsberen - 07:50  
Leeuwen - 07:30  
Panda's - 08:35

Werkschema voor John  
07:30 - Leeuwen  
07:50 - IJsberen 08:35  
- Panda's

===== Uitgebreid  
schema

=====

LET OP: Om 08:35 zouden de Panda's al gevoerd worden

Dieren gevoerd door John  
IJsberen - 07:50  
IJsberen - 12:50  
Kolibries - 08:30  
Konijnen - 08:35  
Leeuwen - 07:30  
Panda's - 08:35  
Zeehonden - 07:10

Werkschema voor John  
07:10 - Zeehonden  
07:30 - Leeuwen  
07:50 - IJsberen  
08:30 - Kolibries  
08:35 - Konijnen  
12:50 - IJsberen

