

 UC Leuven Limburg MOVING MINDS	aantal pagina's.: 6	lector: K. Baerts, P. Fox, M. Lens, B. Philtjens opleidingsonderdeel (OPO/OLA): OO Programmeren code OPO/OLA: MBI53X
datum: 14 juni 2021 groep: 1TI		naam student: voornaam: studentennummer:
hulpmiddelen: eigen laptop zonder wifi	maximumduur: 2 u	punten /

Instructies

Dit deel van het examen wordt uitgevoerd op je eigen laptop. Je mag alle materiaal dat ter beschikking is op je laptop gebruiken. **Je sluit aan het op wifi-netwerk met naam xToledo en paswoord examen001.**

Communicatie met derden tijdens het examen is uitgesloten.

Ga naar x.toledo.ucll.be waar je het examen onder Beschikbare Examens zal vinden. Als je geen toegang hebt, waarschuw je de toezichthouder.

Je dient de juiste bestanden in na het voltooien van deel 2A en deel 2B in de respectievelijke inleverzones van xToledo (inleverzone voor Deel2A en inleverzone voor Deel2B) en drukt op verlaat examen.

Deel 2A: implementatie schaakbord opgave

- Maak een nieuw project in IntelliJ met als naam "jouwNaam_jouwVoornaam_Deel2A"
- Maak een domain package in de src map van je project
- implementeer enkel de klassen en interface uit de gegeven UML; je krijgt al een deel van de klasse Positie cadeau
- maak een klasse DomainException die je gebruikt overal waar zich een Exception voordoet
- Maak een testklasse voor de klasse Positie, waarbij je voor de constructor één illegal case test en één legal case test; plaats deze testklasse op de juiste plaats in je project zoals geleerd tijdens de lessen
- Zip je project op je bureaublad
- Laad het gezippte bestand op in de inleverzone van Deel2A in xToledo

groep:	datum:pg. 2
naam student:	lector:
voornaam:	
studentennummer:	

Deel 2B: opgave op JCF

Instructies

1. Maak in IntelliJ een nieuw Java project met als naam "jouwNaam_JouwVoornaam_deel2B"
2. Maak in de src map van je project een package met de naam domain
3. Maak in de src map van je project een package met de naam ui
4. Kopieer de gegeven klassen Appeke1 en Appeke2 in de ui package
5. Kopieer de gegeven klasse Persoon in de domain package
6. Kopieer het gegeven csv bestand vaccinatieGegevens in je projectmap
7. Vul de ontbrekende code aan (zie //TODO) in de 3 meegegeven klassen
8. Run Appeke1 en Appeke2 in de ui map. Indien je juiste code hebt geschreven krijg je de console output op volgende pagina.
9. Maak een zip file van je src folder van je project met als naam "jouwNaam_JouwVoornaam_deel2B"
10. Laad dit zip bestand op in de antwoordruimte voor deel2B van het OOP examen in xToledo. (Controleer eerst of je zip file alle klassen die je hebt aangevuld bevat!

groep:	datum:pg. 3
naam student:	lector:
voornaam:	
studentennummer:	

Jullie opdracht

Gegeven een bestand met de naam vaccinatieGegevens.csv dat N lijnen (in de praktijk duizenden lijnen!) bevat met op elke lijn volgende info:

- Eerste 3 kolommen datum (dag;maand;jaar) van een geplande vaccinatie
- Kolom 4: emailadres van de te vaccineren persoon
- Kolom 5: eventueel gsmnummer van de te vaccineren persoon

Gegeven de klasse Persoon en startcode voor Appeke1 en Appeke2

Gegeven volgend CSV bestand vaccinatieGegevens

1	5	2021	jan.franssen@hotmail.com	477253691
1	5	2021	an.lemmens@gmail.com	488255366
11	5	2021	jef.leenders@gmail.com	
12	5	2021	jonathan.marien@hotmail.com	789568457
13	5	2021	an.grootaerts@gmail.com	
30	5	2021	jeannine.lemmens@gmail.com	488255366

BELANGRIJK

- Vul in de meegeleverde klassen (Persoon, Appeke1 en Appeke2) de ontbrekende code aan zodat de console outputs op de volgende bladzijde worden bekomen. In Persoon ontbreken eventueel variabelen en methodes. In Appeke1 en Appeke2 moet je op de plaats van //TODO ??? één of meerdere lijnen code toevoegen.
- Focus je bij het aanvullen van code op de typische java collection framework code, niet op Exceptions, ...!
- Je mag in je oplossing **GEEN** gebruik maken van **ArrayList of LinkedList**. Maak gebruik van Set en Map! (zie bijlage op volgende pagina's)

Appeke1:

We wensen snel voor een persoon op basis van het emailadres de datum te kunnen terugvinden waarop deze persoon zal gevaccineerd worden. We typen in de console het emailadres van een persoon in. Als dit emailadres gevonden wordt toont de app de vaccinatiedatum van de bijhorende persoon (email adressen zijn uniek!), zo niet wordt het bericht "geen vaccinatiedatum gevonden" getoond. We kunnen vaccinatiedatums blijven opvragen totdat we voor het email adres een * intypen

groep:	datum:pg. 4
naam student:	lector:
voornaam:	
studentennummer:	

Console output Appeke1:

```

Typ emailadres (* = stoppen)
an.lemmens@gmail.com
Vaccinatiedatum is 2021-05-01
Typ emailadres (* = stoppen)
an.bams@outlook.com
geen vaccinatiedatum gevonden
Typ emailadres (* = stoppen)
*

```

Appeke2:

Appeke2 maakt van de het meegegeven CSV bestand met vaccinatiegegevens 2 overzichten:

Overzicht1:

Geeft een lijst van alle personen uit het vaccinatiebestand met gsm nummer (indien aanwezig, sommige gsm nummers behoren bij meerdere personen!), het emailadres en de bijhorende vaccinatiedatum. Deze lijst is gesorteerd op oplopend gsm nummer en bij gelijk gsmnummer oplopend op email adres. De personen zonder gsmnummer staan achteraan in de lijst in willekeurige volgorde.

Overzicht 2:

Geeft een lijst met vaccinatiedatums in oplopende volgorde met per vaccinatiedatum het aantal personen dat op die datum zal gevaccineerd worden.

Console output Appeke2:

```

477253691  jan.franssen@hotmail.com    2021-05-01
488255366  an.lemmens@gmail.com                 2021-05-01
488255366  jeannine.lemmens@gmail.com          2021-05-30
789568457  jonathan.marien@hotmail.com         2021-05-12
??        jef.leenders@gmail.com             2021-05-11
??        an.grootaerts@gmail.com            2021-05-13

```

```

2021-05-01: 2
2021-05-11: 1
2021-05-12: 1
2021-05-13: 1
2021-05-30: 1

```

groep:	datum:pg. 5
naam student:	lector:
voornaam:	
studentennummer:	

Documentatie Map interface

Modifier and Type	Method and Description
void	clear() Removes all of the mappings from this map (optional operation).
boolean	containsKey(Object key) Returns true if this map contains a mapping for the specified key.
boolean	containsValue(Object value) Returns true if this map maps one or more keys to the specified value.
Set<Map.Entry<K,V>>	entrySet() Returns a Set view of the mappings contained in this map.
boolean	equals(Object o) Compares the specified object with this map for equality.
V	get(Object key) Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key.
int	hashCode() Returns the hash code value for this map.
boolean	isEmpty() Returns true if this map contains no key-value mappings.
Set<K>	keySet() Returns a Set view of the keys contained in this map.
V	put(K key, V value) Associates the specified value with the specified key in this map (optional operation).
void	putAll(Map<? extends K,? extends V> m) Copies all of the mappings from the specified map to this map (optional operation).
V	remove(Object key) Removes the mapping for a key from this map if it is present (optional operation).
int	size() Returns the number of key-value mappings in this map.
Collection<V>	values() Returns a Collection view of the values contained in this map.

groep:	datum:pg. 6
naam student:	lector:
voornaam:	
studentennummer:	

Documentatie Set interface

Modifier and Type	Method and Description
boolean	add(E e) Adds the specified element to this set if it is not already present (optional operation).
boolean	addAll(Collection<? extends E> c) Adds all of the elements in the specified collection to this set if they're not already present (optional operation).
void	clear() Removes all of the elements from this set (optional operation).
boolean	contains(Object o) Returns true if this set contains the specified element.
boolean	containsAll(Collection<?> c) Returns true if this set contains all of the elements of the specified collection.
boolean	equals(Object o) Compares the specified object with this set for equality.
int	hashCode() Returns the hash code value for this set.
boolean	isEmpty() Returns true if this set contains no elements.
Iterator<E>	iterator() Returns an iterator over the elements in this set.
boolean	remove(Object o) Removes the specified element from this set if it is present (optional operation).
boolean	removeAll(Collection<?> c) Removes from this set all of its elements that are contained in the specified collection (optional operation).
boolean	retainAll(Collection<?> c) Retains only the elements in this set that are contained in the specified collection (optional operation).
int	size() Returns the number of elements in this set (its cardinality).
Object[]	toArray() Returns an array containing all of the elements in this set.