 UC Leuven Limburg MOVING MINDS	aantal pagina's: 8	lector: Patrick Fox, Marina Lens, Karen Baerts, Kris Gabriëls opleidingsonderdeel (OPO/OLA): OO-Programmeren code OPO/OLA: MBI53x, MBI53v
datum: 23 augustus 2018 groep: 1TX		naam student: voornaam: studentennummer:
hulpmiddelen: geen	maximumduur: 3u	punten / 20

DEEL 1

Dit is deel 1 van het examen. Je krijgt maximum 1,5 uur om dit uit te werken. Pas als je dit afgegeven hebt, krijg je de vragen voor deel 2.

Ontwerp

Gegeven het verhaal over een nieuw Festivalconcept, teken het UML **klassendiagram**: je moet de klassen, de abstracte klassen, de interfaces, de juiste associaties en pijlen ertussen, de instantievariabelen met hun types en de constanten volledig uitwerken in dit klassen diagram. Gebruik in je oplossing minstens 1 interface en 1 abstracte klasse. Plaats volgende methodes (en eventueel belangrijke methodes extra nodig om deze te implementeren) op de juiste plaats in het klassen diagram. Schrijf de volledige signatuur van deze methodes in UML notatie :

1. Personen kunnen registreren als bezoeker
2. Registreren van een reservatie (met controle of het gevraagde optreden bestaat)
3. Een lijst opvragen van werknemers die gratis bonnetjes moeten krijgen
4. Aanmaken van een werknemer met geldig rijksregisternummer
5. Bereken de totaalprijs voor een gegeven persoon, rekening houdend met eventuele korting
6. Een lijst opvragen van alle optredens, gesorteerd volgens prijs van de artiest

De andere methodes moet je **NIET** uitwerken in dit klassen diagram!

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

Teken hier je UML klassendiagram – Dit mag in potlood indien goed leesbaar

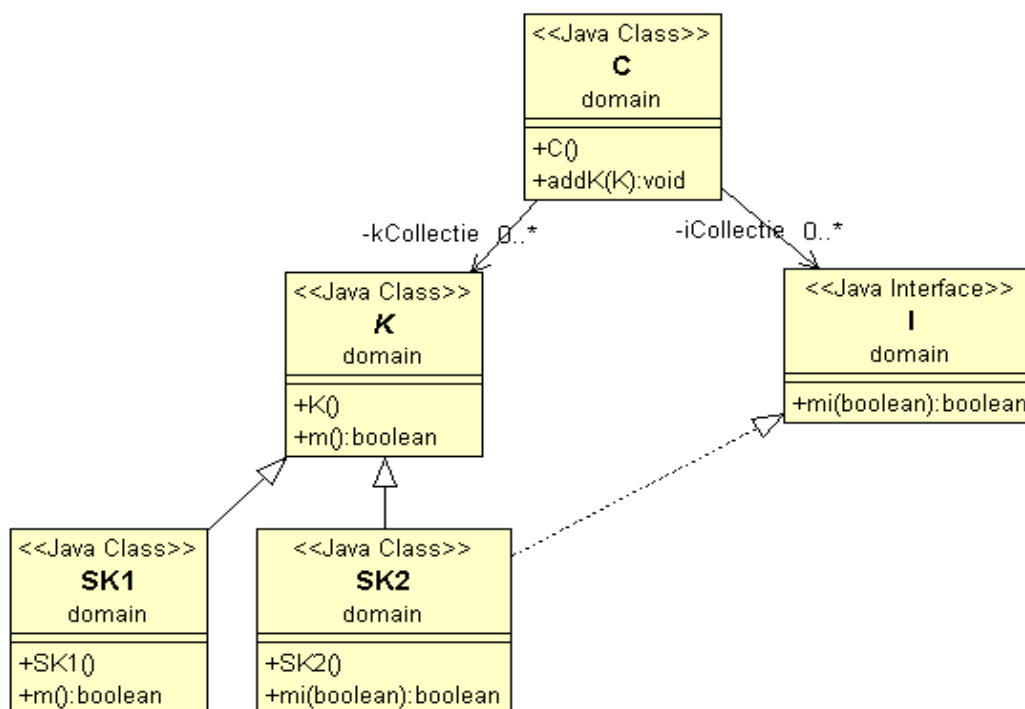
groep:
 naam student:
 voornaam:
 studentennummer:

datum:pg. 3
 lector:

Theorie

VRAAG 1

Gegeven volgend klassendiagram:



Implementatie van m() :

In K : return true;
 In SK1: return false;

Implementatie van mi(boolean p) in SK2:

return p && m()

Geef in de tabel op de volgende bladzijde voor elk van de instructies aan of de code al dan niet compileert. Indien de code niet compileert, vermeld in het kort wat de reden is en wat de eventuele output is.

Je mag indien nodig de ruimte onder de tabel gebruiken om je antwoord te noteren. Zet dan zeker in de betreffende kolom een duidelijke verwijzing hiernaar.

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

CODE	COMPILEERT	REDEN	OUTPUT
SK1 k = new K(); System.out.println(k.m());	JA / NEEN JA / NEEN		
K k = new SK2(); System.out.println(k.mi(true));	JA / NEEN JA / NEEN		
I k = new SK1(); System.out.println(k.mi(true));	JA / NEEN JA / NEEN		
I k = new SK2(); System.out.println(k.mi(false));	JA / NEEN JA / NEEN		
I k = new SK2(); System.out.println(k.m());	JA / NEEN JA / NEEN		

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

VRAAG 2

Een bibliotheek wenst voor elk lid snel informatie kunnen terugvinden over de boeken die momenteel in het bezit zijn van een lid.

We beschikken over de klasse Lid (instantievariabelen: rijksregisternummer, naam, voornaam, adres, emailadres) en over de klasse Boek (instantievariabelen: isbn_nummer, exemplaarnummer, titel, auteur). Het isbn_nummer bepaalt over welk boek het gaat (*bijvoorbeeld het boek met isbn_nummer 9789024415663 is het boek met als titel "Aan de slag met Java en JavaFX" van de auteur "Gerjan Laan"*). Van eenzelfde boek kunnen verschillende exemplaren in het bezit zijn van de bibliotheek. Bijvoorbeeld wanneer er van het boek met isbn_nummer 9789024415663 4 exemplaren in huis zijn worden deze apart genummerd van 1 tot en met 4.

1. Geef de declaratie en initialisatie van de meest gepaste datastructuur uit het Java Collections Framework om deze informatie bij te houden.

2. Geef zowel voor de klasse Lid als de klasse Boek de implicaties op de javacode om deze datastructuur zo performant mogelijk te laten zijn.

a. Gevolgen voor klasse Lid:

b. Gevolgen voor de klasse Boek:

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

3. Bovendien wensen we op een eenvoudige manier een collectie te kunnen genereren bestaande uit alle leden die momenteel minstens 1 boek in hun bezit hebben; deze collectie moet gesorteerd zijn op de gemeente waar het lid woont.
Hoe zou je uit de datastructuur uit puntje 1 deze informatie op een performante manier kunnen genereren?

groep:	datum:pg. 7
naam student:	lector:
voornaam:	
studentennummer:	

VRAAG 3

Gegeven een klasse `Wiel` (implements `Cloneable`) en een klasse `Fiets` (implements `Cloneable`). De klasse `Wiel` heeft 2 integer instantievariabelen (druk en afmeting), getters en setters voor deze variabelen, een constructor (`public Wiel (int druk, int afmeting)`) en een `public Wiel clone()` methode.

De klasse `Fiets` heeft ook 2 instantievariabelen (`Wiel voorwiel` en `Wiel achterwiel`), getters en setters voor deze variabelen, een constructor (`Wiel voorwiel, Wiel achterwiel`) en een `clone()` methode.

1. Veronderstel dat de `clone` methode in de klasse `Fiets` als volgt is geïmplementeerd:

```
public Fiets clone() throws CloneNotSupportedException{
    return (Fiets)super.clone();
}
```

Wat is dan de console output van volgend stukje code en leg uit waarom:

```
Wiel voorwiel = new Wiel(4,28);
Wiel achterwiel = new Wiel(5,28);
Fiets origineel = new Fiets(voorwiel,achterwiel);
Fiets kopie = origineel.clone();
kopie.getAchterwiel().setDruk(4);
System.out.println(origineel.getAchterwiel().getDruk());
System.out.println(kopie.getAchterwiel().getDruk());
```

Consoleoutput + uitleg waarom

groep:	datum:pg. 8
naam student:	lector:
voornaam:	
studentennummer:	

2. Hoe moet je de clone methode in de Fiets klasse gaan aanpassen zodat bovenstaand stukje code de verwachte output zou opleveren?

```
public Fiets clone() throws CloneNotSupportedException{
```