Beginselen van objectgericht programmeren

2008 januari examen

- Maak voor de examenopdracht een UML klassendiagramma. Werk elke klasse uit en maak voor elke klasse een Testklasse (zonder gebruik te maken van JUnit)

Omschrijving examenopdracht G&T ziet een gedroomde kans om mee in de GSM wereld te stappen en ‘low–cost’ operator te worden in België. Zo een beetje als Aldi-talk maar dan enkel voor SMS berichten. Als goede bedrijfsleiders willen we onze klanten serieus opvolgen en het project bijsturen indien nodig. Aan jullie de eer om het geheel te informatiseren. De uit te werken klassen: De klasse Operator bevat:

- een naam (bv. “G&T communication”)
- een ArrayList <Abonnee> : een lijst met al de Abonnee-objecten in onze firma.

De klasse Abonnee bevat:

- een gsm-nummer (String)
- een naam (String)
- een tarief (Tarief)
- een tabel favorieten van 3 favoriete gsm-nummers. (In de toekomst kan het aantal favoriete GSM nrs wijzigen)
- een lijst SMS_en, d.i. een ArrayList <SMS> of tabel van SMS objecten.
- 2 constructoren: eentje met en één zonder Tarief object.

Als er geen tarief wordt meegegeven dan wordt er default een Tarief object geïnitialiseerd met prijs 5 (cent) en geen gratis sms’en.

- setTarief (tarief:Tarief)
- isFavoriet (gsmNr:String) : boolean. Deze methode controleert of een bep. Gsm-nummer is ingesteld als favoriet van dit Abonnee object. Je geeft een boolean terug.
- addSMS (sms:SMS): boolean. Deze methode voegt een SMS object toe, na eerst een check of dit nummer niet reeds in de lijst SMS-en staat. Geeft een boolean terug of dit al dan niet gelukt is.
- voegFavorietToe (i:int,nr:String) Deze methode voegt een gsm nr. (String) toe aan je tabel OP DE PLAATS die je meegeeft met de integer parameter.

Let op: i moet >= 1 en <= 3 en enkel als de plaats vrij is kan je toevoegen!!!
• veranderFavoriet(i:int, nr:String): boolean Deze methode verandert een favoriet op plaats i (i>=1 en i<=3). Enkel als er al een GSM nr. staat kan je veranderen!!! Deze methode geeft een boolean terug.

- berekenSMSFact(maand:int, jaar:int):double Deze methode geeft je de prijs (double) terug van de opgegeven maand en opgegeven jaar. Bv. 12.56 je moet je eurocenten dus herleiden naar euro's.
- toString():String Deze methode geeft je een String notatie van de Abonnee klasse. Je dient mee te geven: je instantievariabelen en een overzicht van de favoriten. Je gebruikt hiervoor de toString() van de desbetreffende klasses.

Return voorbeeld: Abonnee : 0477/406003
Bob Base
4 cent/SMS aantal gratis: 50
Favorieten : 0477/123456 0476/456789 0474/345678

Opmerking. Werk in de klasse Abonnee een methode uit die een aantal SMS berichten random genereert: public void randomSMS (int aantal, int jaar, int vanMaand, int totMaand). Deze methode genereert zowel SMS berichten naar favorieten als naar een dummy Gsm nr (0499/999999)

De klasse SMS bevat:

- naarGSMNr: String
- datum: Datum
- uur: Tijd
- equals (sms:SMS): boolean Deze methode controleert wanneer 2 SMS objecten gelijk zijn. SMS objecten zijn gelijk als de datum en het uur gelijk zijn. De methode geeft een boolean terug.
- toString():String Deze methode geeft een String notatie van je klasse SMS.

Return voorbeeld: 0477/123045 02-10-2008 12:03

De klasse Tarief bevat:

- prijs:int
- aantalGratis.:int
- constructor Tarief (prijs:int, aantalGratis:int)
- de nodige getters en setters
  • equals (tarief:Tarief): boolean. Deze methode gaat na wanneer 2 Tarief objecten gelijk zijn. Tarief objecten zijn gelijk als prijs en aantal gratis gelijk is. De methode geeft een boolean terug.
- toString():String: String notatie van de Tarief klasse.

Return voorbeeld 4 cent/SMS aantal gratis: 50

Nog even wat achtergrond info over tarief:

- per maand zijn de eerste aantalGratis SMS'en gratis.

- De prijs voor 1 bijkomend sms is prijs eurocent.
- De prijs voor 1 SMS naar één van je favorieten is 50% van de gewone prijs afgekapt naar onder dus 5 eurocent wordt 2 eurocent of 7 eurocent wordt 3 eurocent.

2012 januari examen

GEGEVEN

Maak een Stageinformatiesysteem waarin wordt opgeslagen welke stages er vrij zijn en waar studenten kunnen toegewezen worden aan een stageplaats. (Evenals verwijderd worden...) Je krijgt een blad waarop de verschillende klassen staan die je moet maken/vervolledigen(Klassendiagram).Dit zijn:

- Klasse Student
- Klasse Datum
- Klasse Stage
- Klasse Stageinformatiesysteem

Een student heeft naam en nummer. Beiden moeten steeds ingegeven worden.(Constructor met String naam en int nummer). Een student kan voor enkele maanden stage doen in een bedrijf. In de klasse Stage is dan ook een begindatum gegeven.

GEVRAAGD

1) Klassediagram met correcte pijlen aanvullen

2) Schrijf de klasse Student volledig met getters, setters, toString en equals

3) Schrijf voor de andere klassen enkel de methodes die vetgedrukt zijn in het klassediagramma. Van de andere methoden mag je aannemen dat ze feilloos werken en zijn geschreven volgens de regels van de kunst. Zie de testklassen (in bijlage) om te weten wat er juist verwacht wordt.

Klasse Datum is gegeven op 1 methode na: Toevoegen van aantal maanden aan een datum.

!OPGELET: Hierbij moet rekening gehouden worden met: Als datum = 31 januari+1 maand --> wordt 28 OF 29 februari.

4) Schrijf de UI klasse en maak volgende objecten aan:

- 2 studenten
- 2 stages
- 1 StageInformatieSysteem
- Elke student krijgt stage
- 2 stages worden toegevoegd aan het Stageinformatiesysteem.

2014 januari examen

Situatie: Professor Weetal heeft een programma nodig voor zijn examens te maken. Elke vraag is een meerkeuzevraag en heeft een vraagstelling (String), een lijst van mogelijke antwoorden (String[]) en het juiste antwoord (int indexnummer). Een lijst van deze vragen wordt een examen. Voor elke vraag die gesteld wordt verandert de volgorde van vragen willekeurig maar moet het juiste antwoord wel nog steeds overeenkomen met de verwisselde mogelijke antwoorden. Elke vraag in de examenreeks moet ook in een willekeurige volgorde overeenkomen. Voor elk juist antwoord krijgt de leerling +1. Voor elk fout antwoord -0.75. Beide scores moeten weergegeven worden. Bv: Je score was 3.0/5 met giscorretie is dit 1.25/5

Vraag: Schrijf beide klassen (Vraag & Vragenreeks/Examen) uit, schrijf de main methode en maak de klassendiagrammen.

2015 januari examen

8 Januari (Lector: F.Vogels)

Het examen ging door op de PC.

Je moest 3 klassen maken:

- e-mailAccount
- e-mail
- e-mailAdress

Er werden 2 klassen gegeven:

- spamFilter
- time

Je moest zorgen dat de testklasses volledig klopte.

2016 januari examen

14 Januari (Lector: E. Steegmans)

Klasse diagram maken op papier aan de hand van een tekstje.

Fout vinden in de code van 2 kleine programma's + uitleggen waarom het fout is.

Reisprogramma maken aan de hand van UML en javadocs.

3 klasses:

- Land
- Etappe
- Reisweg

(zorgen dat de testklasses klopen)

Dan nog 1 programma waarin je de vorige 3 klasses implementeerd:

- ReisAPP

(ook aan de hand van een tekstje, dus niet UML of javadocs)

2017 januari examen

9 Januari (Lector: M. Lens)

Het examen bestond uit 3 vragen:

1. Maak aan de hand van volgende informatie 4 klassen:

Het is de bedoeling om kooklogboeken bij te houden. Elk kooklogboek heeft een naam die moet beginnen met "Kooklogboek van " met daarna de naam van de eigenaar. Een kooklogboek bevat ook etentjes. Je moet ook het totaal aantal nagerechten in een kooklogboek kunnen krijgen en de totale gemiddelde score.

Een etentje bevat een gastlijst (= een lijst bestaande uit gasten), gerechten en een score. Gasten mogen niet toegevoegd worden als er al gerechten zijn toegevoegd aan het etentje. Wanneer er min. 1 vegetarische gast op de gastlijst staat en er een niet-vegetarisch gerecht wordt toegevoegd, dat mag niet. Gasten en gerechten worden pas na de aanmaak van een etentje toegevoegd, wordt dus niet meegegeven met de constructor. Je moet ook alle gerechten van een etentje kunnen krijgen en de gastlijst. Ook moet je gerechten, gasten en scores toevoegen. (score is de som van alle scores die gegeven zijn aan het etentje, dus bij addScore(4) en daarna addScore(5) wordt score 9. Score mag ook negatief zijn/worden.)

Een gerecht heeft een naam (mag niet leeg zijn) en is vegetarisch of niet. Het heeft ook een soort die hoofdgerecht, voorgerecht of nagerecht is. Het mag ingegeven worden als " HooFdGerEcht", maar moet worden opgeslagen met kleine letters. De naam van een gerecht mag later aangepast kunnen worden, ofdat het vegetarisch is mag niet kunnen aangepast worden. Naam en ofdat het vegetarisch is moet ook opgevraagd kunnen worden.

Een gast heeft een naam (geen checks voor nodig) en is vegetarisch of niet. De naam mag niet kunnen veranderd worden, ofdat de gast vegetarisch is wel. De naam en ofdat de gast vegetarisch is moet opgevraagd kunnen worden.

2. Maak aan de hand van deze klassen een KookLogboekApp:
Maak 2 etentjes met gasten en gerechten en voeg deze toe aan een kooklogboek. Laat daarna het kooklogboek zien in de console. Maak daarna een derde etentje aan met een fout erin. Laat de foutmelding zien in een dialoogbox (= JOption Messagebox).
Het aangemaakte kooklogboek is van Elke met volgende etentjes:

Etentje1:
Gast Stijn en Elke (beide vegetarisch). Eten vegetarische lasagne (hoofdgerecht) en chocomousse (nagerecht). Scores worden gegeven van 8 en 10.

Etentje2:
Gast Jan en Mieke (beide niet-vegetarisch). Eten tomatensoep (voorgerecht) en pasta (hoofdgerecht) en pudding (nagerecht). Scores worden gegeven van 8 en -1.

Etentje3:
Gast Elke en Mieke (veg. en niet-veg.). Eten biefstuk (hoofdgerecht).

In de console:

Kooklogboek van Elke

Etentje 1
Gerechten:
hoofdgerecht: Vegetarische Lasagne - V
nagerecht: Chocomousse - V
Gasten:
Stijn - vegetarisch
Elke - vegetarisch

Etentje 2
Gerechten:
voorgerecht: tomatensoep - V
hoofdgerecht: pasta
nagerecht: pudding - V
Gasten:
Jan
Mieke

Het aantal nagerechten is: 2
De gemiddelde score is: 6,25

3. Teken het geheugen (Stack & Heap) van de KookLogboekApp na het uitvoeren van de laatste lijn code op papier.

2019 januari examen

Het examen bestond uit 2 delen:

Deel 1:

Vraag1) Geef het verschil tussen het gebruiken van == en .equals. (best met voorbeeld met stack en heap).

Vraag2) Waarvoor wordt get methode gebruikt?

Vraag3) Schrijf 2 constructoren voor een gegeven klasse één met 3 parameters en één met 2 parameters. Hoe noemt het als er 2 constructoren zijn?

Vraag4) Je krijgt de uitleg van 3 klassen Rit, Chauffeur en Bedrijf en moet aan de hand hiervan een UML schema opstellen.

(Voor dit deel heb je ongeveer 45 minuten van de 3 uur)

Deeel 2:

Je krijgt bij het afgeven van deel 1 het juiste UML-schema en mag nu op de pc de klasse gaan implementeren aan de hand van de teksten en dit schema.

2020 januari examen

Het examen bestond uit 2 delen:

Deel 1:

Je krijgt de uitleg van 3 klassen (iets meer dan één pagina) Tocht, Deelnemer en Controlepunt en moet aan de hand hiervan een UML-schema opstellen.
(Voor dit deel heb je ongeveer 45 minuten van de 3 uur)

Deel 2:

Je krijgt bij het afgeven van deel 1 het juiste UML-schema en mag nu op de pc de klassen gaan implementeren aan de hand van de teksten en dit schema.

De uitvoer zag er als volgt uit:

Dodentocht 2020: 2020-08-20

Wandelroute:
1. Grote Markt, (50.84511, 4.45197), EHBO POST
2. Campus Proximus, (53.429712, 4.75126)
(Zo waren er 5 controlepunten)

3 deelnemers hebben zich ingeschreven, namelijk:
1. Jan Janssen heeft de tocht beëindigd in 15 uur en 30 minuten.
2. Klaas Claes heeft de tocht nog niet beëindigd, laatste controlepunt: 2
3. ...

Daarnaast waren er nog verschillende methodes die je moest voorzien zoals geef het laatste controlepunt voor een bepaalde deelnemer, kunnen controleren of een deelnemer is gestopt, deelnemer toevoegen, controlepunt toevoegen, totale tijd van een deelnemer berekenen, tussen twee controlepunten zonder EHBO-post moet er zeker een controlepunt met EHBO-post zijn, ... Tot slot moest je bij de klasse Tocht ook gebruik kunnen maken van constructor overloading (1: datum + aantal controlepunten, 2: enkel datum (controlepunten is dan automatisch 15)).

De examen opgaves met dank aan ISW en Avermate Ben: Januari (1).zip

Algemene tips

Ga zo vaak mogelijk naar de les, maak de opdrachten en vraag hulp indien nodig, scoor goed op testen en haal de simpele punten door je opdrachten af te maken. Het examen wordt op papier gemaakt, vergeet zeker en vast niet je getters en setters correct te schrijven.

VANAF 2015 wordt dit examen op de PC gemaakt, moet je methodes maken aan de hand van een UML en javadocs.

Samenvatting Pieter Van der Elst

datum van upload: onbekend

Samenvatting_BOP_Pieter_Van_der_Elst (1).docx