

OGP Oefentest nov 2024

NAAM: ...

COMPUTERNR: ...

PRAKTISCHE RICHTLIJNEN

De nodige informatie en startcode staat in het bestand `C:\temp\ogp_start.zip`.

- Unzip het bestand eerst met 7-Zip, *Extract Here*. Dan zou je de map `ogp` moeten hebben.
- De map *Exemplaren* is de projectmap waarin je werkt.

Bewaar geregeld je oplossing!

Op het einde van de test: oplossing indienen

- Heb je het bestand `naam.txt` (staat in het project) ingevuld?
- Bewaar alle bestanden (of laat het project lopen, dan is dat automatisch gebeurd).
- Zet IntelliJ uit, en navigeer naar `C:\temp`.
- Rechtermuisklik op de projectmap `ogp`, en kies 7-Zip en dan *Add to "ogp.zip"*.
- Controleer: staat het bestand `ogp.zip` onmiddellijk in `c:\temp` (niet verpakt in een extra map)? En is de extensie inderdaad `.zip` en niet `.7z`? Alleen dan zal de code juist afgehaald worden.
- Laat de computer zo staan, niet uitloggen, niet afsluiten.

Veel succes!

OPDRACHT

In het voorliggende project schrijven we code om de administratie van een bibliotheek (en het uitlenen van boeken) te automatiseren.

Een korte voorstelling van de klassen volgt hieronder.

- Een object van de klasse **BoekInfo** of **PrentenboekInfo** houdt informatie bij die een stuk literatuur karakteriseert. Zoals titel, auteur en isbn-nummer.
- Om makkelijk een stuk literatuur op te zoeken (bvb. alle titels van een bepaalde auteur) is er de klasse **Catalogus**.
- Van elk stuk literatuur bestaan er meerdere fysieke exemplaren. Informatie over een fysiek boek bevat zowel een verwijzing naar titel, auteur e.d. als een id (de sticker die een bibliotheek in een boek plakt, om het te kunnen onderscheiden van alle andere boeken in de bib, zelfs die met dezelfde titel en auteur). Deze info wordt verzameld in de klasse **Exemplaar**.
- De klasse **Bibliotheek** bevat zowel een lijst van exemplaren, als een catalogus waarin gezocht kan worden. De gebruiker zal aan deze klasse vragen of hij een specifiek boek kan uitlenen.

Belangrijk: verander nergens ongevraagd returntype of parameterlijst van de methodes. Klassen aanvullen zonder dat daar expliciet de opdracht toe gegeven wordt, is wel toegelaten (denk bvb. aan hulpmethodes).

In het hoofdprogramma zijn een aantal tests voorzien. Haal deze op het juiste moment uit commentaar. Blijf niet te lang hangen als je foutjes in de output ziet. Voor deze test wordt er op code verbeterd, niet op de output.

Tot slot: bij technische problemen verwittig je onmiddellijk de toezichter. (Qwerty/Azerty-probleem, cursus eet letters op, imports zijn toegevoegd maar blijven fouten geven,...)

Klasse Main

Hier vind je een aantal (al dan niet gebruiksvriendelijke) testen. Haal ze uit commentaar naarmate het coderen vordert.

Klasse BoekInfo

De informatie van een boek die bewaard wordt bestaat uit drie private attributen (type **String**): het isbn-nummer, de titel en de auteur. Dat staat al in de code. Vul de klasse **BoekInfo** verder aan.

1. Bij aanmaken van een **BoekInfo**-object wordt er één stuk tekst meegegeven. Dit kan je hieronder in het voorbeeld zien.

```
BoekInfo boek = new BoekInfo("magweg;12345644X;Over het kanaal;Annelies Beck");
```

Merk op: de parameter bestaat uit verschillende onderdelen, gescheiden door punt-komma's. Het eerste stukje wordt genegeerd. Het tweede is het isbn-nummer. Het derde is de titel. Het vierde is de auteur.

2. Schrijf een getter voor isbn-nummer en auteur, *niet* voor de titel.

3. Zorg ervoor dat een object van deze klasse leesbaar uitgeschreven kan worden. Voor het bovenstaande voorbeeld komt er (volg dit letterlijk):

```
BoekInfo 12345644X Over het kanaal door Annelies Beck
```

Klasse PrentenboekInfo

Implementeer de klasse `PrentenboekInfo`, wees zuinig met code.

4. Zorg dat deze klasse afgeleid is van de vorige klasse.
5. Bij constructie bevat de parameter achteraan ook nog de naam van de tekenaar (of illustrator) van het prentenboek. Bijvoorbeeld:

```
PrentenboekInfo boek
= new PrentenboekInfo("magweg;456789846;Toen de zee stil was;Annelies Beck;
                      Karolien Raeymaekers");
```

6. Schrijf een getter die de naam van de tekenaar (of illustrator) teruggeeft.
7. Zorg dat een object van deze klasse leesbaar uitgeschreven kan worden. Ook deze keer start de output dan met de naam van de klasse, gevolgd door alle informatie. Volg letterlijk wat hieronder staat - dan zullen de testen in het hoofdprogramma slagen. (Let wel: alles staat op een lijn; alleen is dit blad hiervoor te smal...)

```
PrentenboekInfo 456789846 Toen de zee stil was door Annelies Beck --- getekend door Karolien
Raeymaekers
```

Test de klassen `BoekInfo` en `PrentenboekInfo` uit door in het hoofdprogramma de oproep en de implementatie van de methode `testKlassenBoekEnPrentenboek()` uit commentaar te halen.

Klasse Catalogus

Vul de klasse `Catalogus` aan. De instantievariabele staat er al; voeg er geen aan toe. Merk op: de instantievariabele zal toelaten om info over boeken op te zoeken als het `isbn`-nummer gekend is. De keys van de gegeven map zijn dus `isbn`-nummers.

8. Vul de constructor aan. Er wordt een bestandsnaam meegegeven, op basis waarvan de instantievariabele opgevuld wordt. De structuur van het bestand kan je afleiden uit onderstaand voorbeeld; en kan ook bekeken worden door het bestand `cat.csv` te openen.

```
BoekInfo;936667353;Achter gesloten deuren;B.A. Paris;
PrentenboekInfo;320006023;Toen de zee stil was;Annelies Beck;Karolien Raeymaekers
BoekInfo;559933538;Casa ibiza;Linda van Rijn;
```

Op elke regel is er informatie te vinden over een boek of prentenboek. Elke lijn voldoet aan de structuur die de constructor van een `BoekInfo`- of `PrentenboekInfo`-object vraagt; maak daar gebruik van om de code kort te houden.

Zorg ervoor dat je *geen* `if/else`-structuur gebruikt bij het inlezen van het bestand. Gebruik een methode die op basis van de ingelezen informatie (nl. het eerste stukje van elke regel) automatisch het juiste object aanmaakt.

Als het bestand niet gevonden wordt, dan is de catalogus leeg.

Als een bepaalde regel verkeerde informatie bevat, wordt die regel genegeerd. (Geen specifieke actie vereist.)

9. Schrijf de methode `BoekInfo getBoekInfo(String isbn)` die, gegeven een isbn-nummer, het juiste `BoekInfo`-object teruggeeft.
Als het isbn-nummer niet gevonden wordt, dan wordt er een exceptie opgeworpen van type `IsbnNietGevonden` met als boodschap "`isbn ... niet gevonden`" (waarbij de puntjes ingevuld zijn). Deze exceptie moet niet verplicht opgevangen worden.
10. Schrijf een methode die ervoor zorgt dat een object van de klasse `Catalogus` leesbaar uitgeschreven kan worden. Voor elk boek komt er isbn-nummer en auteur (geen titel). Een catalogus van 3 boeken zal er dan bvb zo uit zien:

535758773 Linda van Rijn
878590631 Suzanne Vermeer
857925272 Samuel Bjørk

11. Vul de methode `List<BoekInfo> zoekOpAuteur(String auteur)` aan. Die krijgt (een stuk van) de naam van een auteur mee. Dan worden alle boeken opgezocht die door deze auteur geschreven werden. (Laat hoofd- of kleine letters geen verschil maken in het resultaat.) Deze boeken worden in een list teruggegeven. Voorbeeld: als er op "`ANNE`" gezocht wordt, komt er een lijst van 7 boeken terug (2 van *Suzanne Vermeer*, 3 van *Annelies Beck*, 1 van *Annet Schaap*, 1 van *Annegreet van Bergen*).
12. Vul de methode `PrentenboekInfo[] drieVanIllustrator(String illustrator)` aan. Deze methode krijgt de volledige naam van een illustrator mee. Alle boeken worden overlopen; de eerste drie prentenboeken getekend door die illustrator worden in een array van lengte drie bewaard en teruggegeven. Merk op: hoofd- en kleine letters van de parameter moeten ook juist zijn, anders wordt de illustrator niet herkend. Als er minder dan drie prentenboeken van die illustrator zijn, dan zal er toch een array van lengte 3 teruggegeven worden (de laatste elementen zijn dan `null`).
Tot slot: zit je in met de efficiëntie van je code? Niet doen; je mag gerust de hele collectie boeken doorlopen. De theorie die nodig is om hier echt efficiënt te werken werd nog niet gezien.

Test de klasse `Catalogus` uit door in het hoofdprogramma de oproep en de implementatie van de methode `testKlasseCatalogus()` uit commentaar te halen.

Klasse Exemplaar

De instantievariabelen van de klasse `Exemplaar` zijn al gegeven. Er werd voor gekozen om de informatie die hier bewaard wordt, heel summier te houden. Alleen het id van een fysiek exemplaar (een geheel getal; een uniek nummer dat de bibliothecaris in het boek plakt), het isbn-nummer van het boek, en het feit of het exemplaar uitgeleend is.

13. Schrijf de constructor, die enkel id en isbn-nummer meekrijgt. Een boek is bij de start niet uitgeleend.
14. Schrijf voor elke instantievariabele een getter.
15. Schrijf de methode `leenUIT()` die de uitleen-status van het boek verandert: het is nu uitgeleend. Hier wordt geen expliciete controle op de oorspronkelijke status gedaan.
16. Indien er twee `Exemplaar`-objecten met hetzelfde id aan een set zouden worden toegevoegd, dan zal enkel het eerste bewaard mogen worden. Daarom: zorg ervoor dat `Exemplaar`-objecten met hetzelfde id als gelijk beschouwd worden.

Voor deze klasse is er geen expliciete testmethode in het hoofdprogramma.

Klasse Bibliotheek

De klasse bibliotheek heeft drie instantievariabelen. Verander daar niets aan. Hieronder volgt eerst wat uitleg; nog geen actie gevraagd. (Pas vanaf puntje 17 schrijf je zelf weer code.)

- De instantievariabele `catalogus` is een `Catalogus`, en zal toelaten om boekinformatie op te zoeken. Zo kan een klant van de bib in de `catalogus` bijvoorbeeld alle boeken van een bepaalde auteur zoeken. (Zie de klasse `Catalogus`.)
 - De instantievariabele `boeken` bewaart alle exemplaren uit de bibliotheek in een (niet-gesorteerde) set.
 - De instantievariabele `exemplarenVanIsbn` is een map. Deze instantievariabele zal toelaten om na te gaan hoeveel (en welke) exemplaren van een bepaald boek behoren tot de collectie van de bib. (En zo is er ook meteen geweten of deze exemplaren uitgeleend zijn of niet.)
17. Implementeer de constructor van de klasse `Bibliotheek`. Deze krijgt twee bestandsnamen mee: de naam van het bestand waar de `catalogus` in te vinden is (zoals bvb het bestand *cat.csv*), en de naam van het bestand waar de exemplaren van de bib in te vinden zijn. Dat laatste bestand (zoals bvb *bib.txt*) bevat per lijn twee gegevens, gescheiden door een punt-komma: het id van het exemplaar, en het isbn-nummer.
- Vul in eerste instantie alleen de instantievariabelen `catalogus` en `boeken` in. Op het einde van de constructor wordt de hulpmethode `vulExemplarenVanIsbn()` aangeroepen. Die zal de instantievariabele `exemplarenVanIsbn` opvullen, zie volgend punt.
- Nog dit: indien een van de bestanden niet gevonden wordt, dan zal de bijhorende instantievariabele leeg blijven.
18. Implementeer de hulpmethode `vulExemplarenVanIsbn()`. Deze zal, op basis van de reeds opgeslagen informatie in de set `boeken`, de map `exemplarenVanIsbn` invullen. Overloop alle exemplaren uit de instantievariabele `boeken`. Zorg dan dat dit exemplaar in de map komt te staan. De keys van de map zijn isbn-nummers. Bij elk isbn-nummer hoort een lijst van `Exemplaar`-objecten.
19. Implementeer de methode `void vraagBoekAan()`. Deze methode zal in interactie gaan met de gebruiker. We geven eerst het mogelijk verloop van deze methode, zoals door de gebruiker gezien op de console.

```
Welke auteur?  
annelies  
KIES UIT, tik isbn in  
PrentenboekInfo 320006023 Toen de zee stil was door Annelies Beck --- getekend door Karolien Raeymaekers  
BoekInfo 107391952 Over het kanaal door Annelies Beck  
PrentenboekInfo 364991117 Gedachten denken door Annelies Beck --- getekend door Hanneke Siemensma  
111111111  
Opnieuw graag  
107391952  
U leende exemplaar uit met id 40
```

Op bovenstaand screenshot is het volgende te zien:

- (a) Eerst vraagt de methode welke auteur opgezocht moet worden.
- (b) De gebruiker geeft *annelies* in.

- (c) Het programma antwoordt met *KIES UIT, tik isbn in* en laat dan alle boeken zien die door de gevraagde auteur geschreven werden. Roep hiervoor de methode `zoekOpAuteur` uit de klasse `Catalogus` op.
- (d) Als de gebruiker een isbn-nummer ingeeft dat niet te vinden is als sleutel in de map `exemplarenVanIsbn`, moet hij/zij opnieuw proberen. (Dit gaat door tot hij een juist nummer ingeeft/copieert).
- (e) Als het boek in voorraad is, wordt dit boek ook effectief uitgeleend. (Of nog: als er een niet-uitgeleend exemplaar aanwezig is.)

In volgend screenshot staat er een alternatief verloop.

```
Welke auteur?
anne
KIES UIT, tik isbn in
BoekInfo 878590631 IJskoud door Suzanne Vermeer
PrentenboekInfo 320006023 Toen de zee stil was door Annelies Beck --- getekend door Karolien Raeymaekers
BoekInfo 736202159 Lampje door Annet Schaap
BoekInfo 107391952 Over het kanaal door Annelies Beck
PrentenboekInfo 364991117 Gedachten denken door Annelies Beck --- getekend door Hanneke Siemensma
BoekInfo 653990791 Winternacht door Suzanne Vermeer
BoekInfo 251577090 Het goede leven door Annegreet van Bergen
107391952
Alle exemplaren zijn uitgeleend.
```

- (a) De start is idem.
- (b) Maar blijkbaar vroeg de gebruiker een boek aan waarvan er geen enkel exemplaar meer uitleenbaar is. Dat wordt dan gemeld.

Tot slot nog een laatste alternatief verloop.

```
Welke auteur?
ik weet t niet
Auteur niet gekend.
```

- (a) De gebruiker geeft een ongekende auteur op.
- (b) Dan kan de gebruiker niet opnieuw ingeven; hij krijgt enkel de melding dat deze auteur niet gekend is.

Test de klasse `Bibliotheek` uit door in het hoofdprogramma de oproep en de implementatie van de methode `testKlasseBibliotheek()` uit commentaar te halen.

De laatste methode uit de klasse (`vraagBoekAan()`) wordt ook twee maal opgeroepen; daar wordt jouw actieve deelname gevraagd om te testen. Mogelijke testwaarden: vraag naar de auteur `elsa`, en leen het boek met isbn-nummer `777888999` uit. De tweede keer vraag je opnieuw naar de auteur `elsa`, en leen je hetzelfde boek `777888999` uit. Dat zou moeten resulteren in de boodschap *Alle exemplaren zijn uitgeleend*.