Java OO & Relational Databases

Project Vakantiedomein

K. Van Assche, M. De Schoenmacker, P. Coussens, B. Derudder
E.J. Jacobs, Y. Blancquaert, K. Verbeeck

Inhoudsopgave

1. Java OO

1.1. Opdracht

Via dit project zal je een grafische toepassing schrijven in Java die een overzicht geeft van de kosten voor het algemeen beheer en onderhoud van een vakantiedomein.

Het domein omvat meerdere gebouwen. Elk gebouw bevat één of meerdere verblijfsunits, die in private eigendom zijn. Er zijn verschillende types units, zoals 2-, 4-, 6-, … persoonsunits. Een syndicus staat in voor het beheer van de kosten en de jaarlijkse afrekening ervan naar de eigenaars toe.

De syndicus stelt een afrekening op voor elk gebouw in het domein afzonderlijk. Eigenaars van de verblijfsunits in het betreffende gebouw betalen hun aandeel in de afrekening van het gebouw volgens het aantal en type verblijfsunits dat ze bezitten.

Ter ondersteuning van het project vind je op Toledo een set van .csv-bestanden terug, met daarin concrete data van het vakantiedomein:

- **ConfiguratieGebouwen.csv**
  Bevat een overzicht van de beschikbare gebouwen, hun verblijfsunits en respectievelijke eigenaars

- **Onkostencategorieen.csv**
  Bevat een overzicht van de onkostencategorieën. Zo is er een categorie voor de brandbeveiliging, de liften, het sanitair en verwarming, enz…

6100	Brandbeveiliging
6101	Liften
6103	Sanitair en verwarming
…	

- **Onkostenstaten<gebouw><jaar>/<onkostencategorie>.csv**
  Voor elk gebouw wordt jaarlijks een lijst opgesteld van de onkostenstaten. Elke onkostenstaat hoort bij een bepaalde onkostencategorie. In de map ‘Onkostenstaten<gebouwX><jaarY>’ vind je voor <gebouwX> een overzicht van de onkostenstaten van <jaarY>, gegroepeerd per onkostencategorie (6100, 6101, …). Aan elke onkostenstaat wordt een bedrag gekoppeld, alsook een of meerdere bewijsstukken.

- **Bewijsstukken.csv**
  Hierin zijn de diverse bewijsstukken van de onkosten terug te vinden. Sommige bewijsstukken zijn facturen, andere zijn overzichten, bv. opgemaakt door de syndicus, … Een bewijsstuk kan gekoppeld zijn aan verschillende onkostenstaten.

- **Leveranciers.csv**
  Bevat een overzicht van de leveranciers. Zij zijn de partijen die de bewijsstukken opstellen en hiermee kosten aanrekenen aan het vakantiedomein.

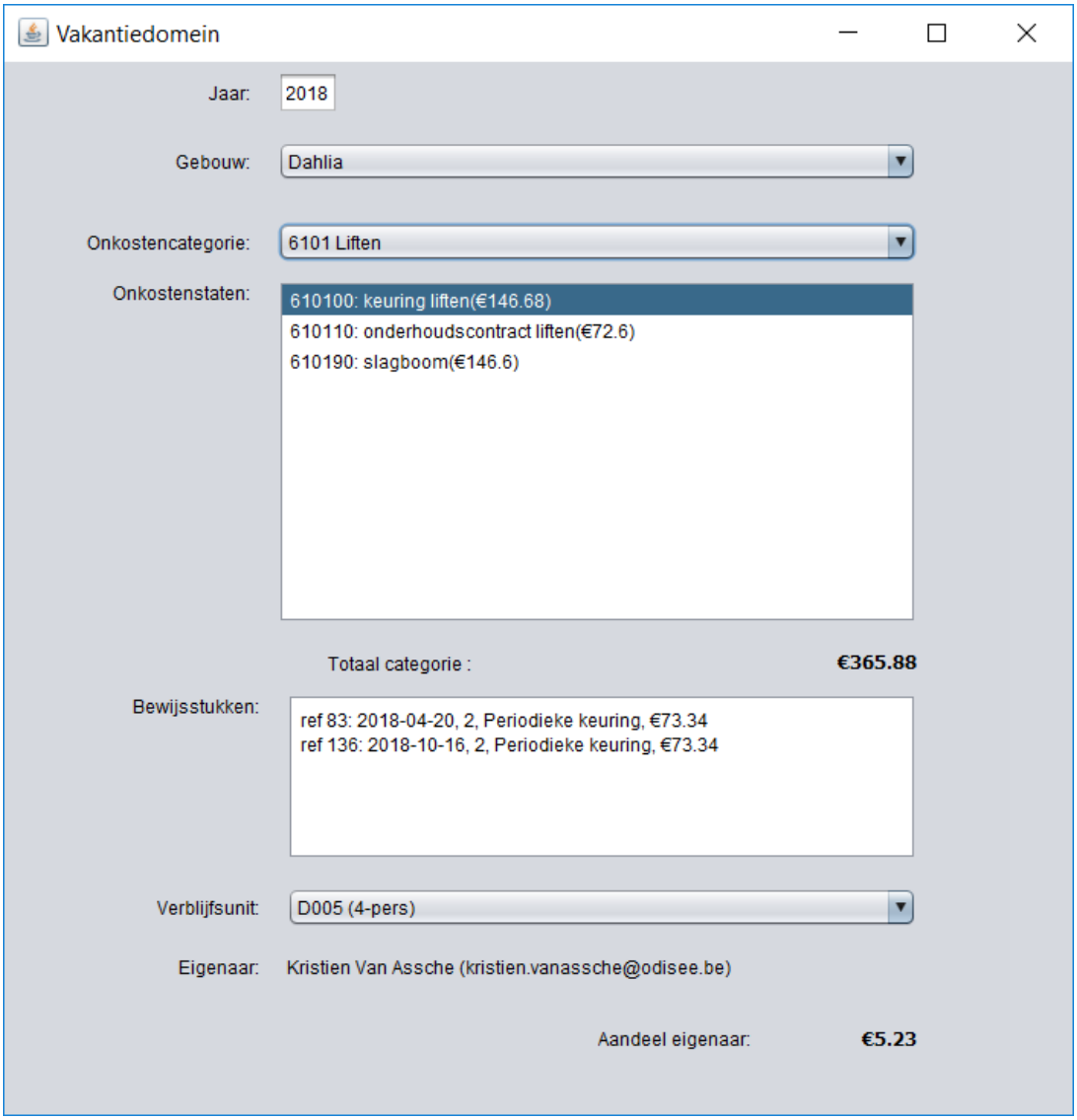
Het is de bedoeling dat je alle gegeven data inleest en vervolgens integreert in een grafische toepassing. Je zal m.a.w. de gemaakte en bijgevolg te betalen kosten tonen in een grafisch venster, en dat op basis van een reeks filters die je dynamisch kan instellen (zie figuur):

- **Jaar:** Het jaartal stel je in via een tekstveld. Merk op dat in de gegeven dataset enkel onkosten zijn opgenomen voor het jaar 2018.
- **Gebouw:** Alle gebouwen van het domein worden opgenomen in een combobox.

- **Onkostencategorie:** Alle onkostencategorieën zijn opgenomen in een combobox. Neem hier een bijkomend item 'ALL' op, waarmee je alle onkosten van het geselecteerde jaar en gebouw kan opvragen.
- **Onkostenstaten:** In deze lijst (JList-object) toon je alle onkostenstaten van het gekozen jaar/gebouw/onkostencategorie. Het is de bedoeling om items van deze lijst te selecteren
- **Bewijsstukken:** Hier krijg je een overzicht te zien van de bewijsstukken die gekoppeld zijn aan een geselecteerde onkostenstaat.
- **Verblijfsunit:** In deze combobox neem je alle verblijfsunits op van het geselecteerde gebouw.

De essentie van de opdracht is om te berekenen wat de kost is die overeenstemt met het gekozen niveau van selectie. Zo zal bijvoorbeeld voor gebouw Dahlia in 2018 de totaalkost voor de onkostencategorie ‘Liften' 365.88 euro bedragen.

Een bijkomende functionaliteit is om te berekenen wat de te betalen bijdrage is van de diverse eigenaars van een verblijfsunit van het gebouw. Eerder werd reeds vermeld dat het aandeel van elke eigenaar bepaald wordt volgens het type verblijfsunit. De som van de capaciteit van alle verblijfunits van een gebouw bepalen de totaalcapaciteit van dat gebouw. De eigenaar van een 4-persoonsunit in gebouw Dahlia zal bijgevolg voor $4/280^{e}$ bijdragen aan de te betalen som.


Vakantiedomein
Jaar:
2018
Gebouw:
Dahlia
Onkostencategorie:
6101 Liften
Onkostenstaten:
610100: keuring liften(€146.68)
610110: onderhoudscontract liften(€72.6)
610190: slagboom(€146.6)
Totaal categorie :
€365.88
Bewijsstukken:
ref 83: 2018-04-20, 2, Periodieke keuring, €73.34
ref 136: 2018-10-16, 2, Periodieke keuring, €73.34
Verblijfsunit:
D005 (4-pers)
Eigenaar:
Kristien Van Assche (kristien.vanassche@odisee.be)
Aandeel eigenaar:
€5.23

1.2. Evaluatie

FUNCTIONALITEIT VAN JE TOEPASSING <20 punten>	Punten
1. Overzicht van de gebouwen	2
2. Overzicht van de onkostencategorieën	2
3. Overzicht van de onkostenstaten voor geselecteerd jaar/gebouw/categorie	3
4. Weergave van berekend categorietotaal	3
5. Overzicht van de bewijsstukken gekoppeld aan geselecteerde onkostenstaat	2
6. Overzicht van totaal én bewijsstukken bij categorieselectie 'ALL'	2
7. Weergave en update van verblijfsunits volgens geselecteerd gebouw	2
8. Weergave van eigenaar van geselecteerde verblijfsunit	2
9. Weergave van berekende bijdrage voor eigenaar van verblijfsunit	2

OPBOUW VAN JE TOEPASSING <20 punten>	punten
Object-georiënteerd ontwerp	10
Programmastructuur	2
Efficiënte programmatie	2
Robuust programma	2
Grafische opbouw	2
Java-stijlregels gevolgd	2

2. Relational Databases

2.1 Design

Ontwerp een gepast databankschema in MySQL Workbench voor de gegeven toepassing en plaats deze op je databanksysteem via forward engineering. Respecteer de regels zoals gezien tijdens de theorie en de labo's.

Voorzie in je ontwerp ook de mogelijkheid om bij te houden welke eigenaars hun aandeel in de onkosten reeds betaald hebben.

2.2 Implementatie

Schrijf een Java consoletoepassing die je databank opvult met alle beschikbare data uit de opgegeven .csv-bestanden. Merk op dat je de functionaliteit om .csv-bestanden uit te lezen reeds geïmplementeerd hebt in de Java OO opdracht. De ingeladen gegevens wegschrijven naar de databank doe je via Sql2o (zie http://www.sql2o.org/). Op http://www.sql2o.org/docs/configuration/ vind je relevante sql2o-documentatie terug.

Bevraag je databank en geef een overzicht weer van de gebouwen, elk met zijn verblijfsunits en hun respectievelijke eigenaar.
Bevraag je databank en geef het totaalbedrag weer van de onkosten van gebouw Dahlia voor het jaar 2018, alsook een overzicht van de onkosten per categorie.

Je uitvoer moet er ongeveer als volgt uitzien:

```
=== Overzicht gebouwen, verblijfsunits en hun eigenaar ===
Azalea
        - A000 (10-pers) -> Luk Schoofs
Boterbloem
        - B000 (12-pers) -> Mia Janssens
Clematis
        - C000 (2-pers) -> Piet Coussens
        - C001 (4-pers) -> Dirk Claus
        - C002 (6-pers) -> Patrik Debbaut
        - …

Totaal onkosten Dahlia 2018: €53892.52

=== Overzicht onkosten per categorie ===
6100 Brandbeveiliging: 350.9
6101 Liften: 365.88
6103 Sanitair en verwarming: 951.24
…
```

2.3 Afspraken

Voor wat betreft het databankonderdeel van de opdracht gelden volgende afspraken:

- gebruik als login `root` en als wachtwoord `Azerty123`
- maak een lokale databank aan met als naam `VoornaamFamilienaam` (hier vul je uiteraard je eigen voor- en familienaam in)
- neem alvorens in te dienen een dump van je MySQL databank (zodat wij je databank zonder enige aanpassing kunnen gebruiken) en geef het bestand volgende naam: `VoornaamFamilienaam.sql` (hier vul je uiteraard ook je eigen voor- en familienaam in)
- zorg ervoor dat er slechts één .sql-bestand in je volledige zip-bestand aanwezig is
- plaats dit .sql-bestand in de startfolder van je NetBeans-project
- plaats ook het design van je databank (.mwb) in de startfolder van je NetBeans-project

Opgelet: We zullen vóór het verbeteren een script laten lopen over jullie projecten dat op zoek gaat naar alle databank-dumps om die automatisch te importeren naar onze lokale databankserver. Daarom is het heel belangrijk dat je bovenstaande richtlijnen heel nauwgezet volgt! Indien er, door het niet volgen van bovenstaande regels, een probleem optreedt bij het testen van jullie project, zal dit leiden tot een 0 voor het onderdeel Functionaliteit van de OPO Relational Databases.

2.4 Evaluatie

<20 punten>	punten
Databank ontwerp	10
Opvullen databank via code	4
Bevragen databank m.b.t. gebouwen, hun verblijfsunits en resp. eigenaar	3
Bevragen databank m.b.t. totaalbedrag onkosten voor gebouw/jaar & bijhorend overzicht per categorie	3

3. Indienen

Je project wordt als **één zip** - géén RAR - op Toledo verwacht ten laatste op **vrijdag 31 mei 2019 om 20u**. Vermeld in de naam van de zip duidelijk je naam én je labogroep. Zorg ervoor dat je hele project in de zip is opgenomen, het project moet zonder meer gecompileerd en uitgevoerd kunnen worden op een labo-pc! Probeer een kopie van je project ook eens op een andere computer uit te voeren om te zien of alles nog werkt (bij voorkeur een pc uit het labo).
Let er hierbij op dat je de nodige bibliotheken in een lib-folder binnen je Netbeans solution plaatst en dat je naar deze locatie refereert. Neem regelmatig back-ups. Kopiëren of laten kopiëren resulteert in een 0 voor dit project.

De hierboven aangegeven evaluatiecriteria vind je ook terug op het **zelfevaluatieformulier** dat je bij de opgave vindt. **Vul dit in en doe hiervan ook een upload naar Toledo!**

Veel succes !