

1. Inleiding

Opgave 1: javascript patterns (theorie)

Bestudeer hoofdstuk 1, 2 van Pro JavaScript Design Patterns (Harmes en Diaz, 2008). Hoewel dit al een oude titel is, is de theorie die hierin ter sprake komt dat niet per se. Beantwoord de volgende vragen:

1. Waarom is JavaScript zo'n populaire taal?
2. Wat is een prototype in JavaScript? Geef een voorbeeld.
3. Wat is een *anonymous function* en hoe verhoudt zich dit tot *closure*? Geef een voorbeeld.
4. Hoe kan *closure* in JavaScript ingezet worden om de methoden en variabelen van een klasse *private* te maken?
5. Hoe geef je in Javascript een attribuut aan een functie en wat zou een praktische toepassing daarvan kunnen zijn?
6. Waarom is het beter om bij het programmeren te werken met interfaces?
7. Wat is *Duck Typing* en op welke manier kan die techniek gebruikt worden om het gebrek aan interfaces in Javascript op te lossen?
8. Wat is *encapsulation* in de context van Objectgeoriënteerd programmeren?
9. In deze hoofdstukken worden een aantal methoden beschreven om specifiek OO-gerelateerde technieken te implementeren in JavaScript. Beargumenteer waarom dit wel of geen goed idee is.

Opgave 2: energie website

Steeds meer mensen maken gebruik van eigen energiebronnen, zoals zonnepanelen of windmolens, maar bijvoorbeeld ook mestvergisters. In deze opgave wordt er van uitgegaan dat deze bronnen allemaal de mogelijkheid hebben om hun energieproductie af te lezen. De site in deze opgave is bedoeld om mensen inzicht te geven in deze productie en om dit te vergelijken met anderen.

Deze en de volgende twee opgaven gaan over deze website. Hierbij is het van belang dat aan het eind van elke opgave de code in git wordt ingecheckt, opdat de situatie op dat moment wordt vastgelegd.

Download nrg.zip van Blackboard. Hierin zit een sql-bestand met alleen postcodes en apparaten als stambestanden gevuld. De rest van de database is leeg. Het is bedoeling deze database met behulp het bijgeleverde c-programma te vullen.

1. In de stamtabel apparaat staan nu vijf apparaten. Voeg hier nog enkele aan toe om een wat diverser eindresultaat te krijgen.
2. Bestudeer main.c. Vul hierin de juiste gegevens in om verbinding te kunnen maken met je lokale database. Om het bestand te compileren moet je de juiste paden naar de mysql-libraries opgeven. Op mijn os-x machine gaat het compileren op bijvoorbeeld de onderstaande manier:

```
gcc -o nrgdatabase $(/usr/local/mysql/bin/mysql_config --cflags) $(/usr/local/mysql/bin/mysql_config --libs) main.c
```

In sommige gevallen kan het programma de library libmysqlclient.18.dylib niet vinden. Maak in dat geval een symlink naar deze library in je mysql-distributie:

```
ln -s /usr/local/mysql/lib/libmysqlclient.18.dylib libmysqlclient.18.dylib
```

3. Wanneer je het programma runt, vraagt het naar het aantal huishoudens dat gegenereerd moet worden en het maximale aantal apparaten per huishouden. Momenteel gebeurt er echter nog niks. Pas het programma aan zodat er daadwerkelijk zoveel huishoudens worden aangemaakt met een random aantal apparaten per huishouden.
4. Run het programma zodat de database gevuld wordt. Gebruik hiervoor een flinke dataset, dus bijvoorbeeld enige tientallen huishoudens met gemiddeld tien apparaten.
5. Maak een webinterface naar de database. Implementeer daarbij de onderstaande functionele eisen en pas de database waar nodig aan. Maak gebruik van less of sass om de voorkant er goed uit te laten zien:
 - ✓ Bezoekers kunnen zich op basis van huishouden registreren. Het aantal personen waar een huishouden uit bestaat (één of meer) moet worden opgeslagen, evenals standaardgegevens als e-mailadres en telefoonnummer.
 - ✓ De geregistreerde kan aangeven welke energie-producerende apparaten er in het huishouden aanwezig zijn. Van dit apparaat worden merk en typenummer opgeslagen.
 - ✓ Na registratie kan de bezoeker op gezette tijden van elk van haar apparaten de productie opgeven. Bij zo'n opgave wordt de tijd ook geregistreerd. Elke meting bestaat uit de productie op dat moment (in kW) en de totaal opgeleverde hoeveelheid energie (in kWh).
 - ✓ De bezoeker kan in een tabel de gegevens haar apparaten inzien, op vergelijkbare manier zoals hieronder wordt weergegeven:

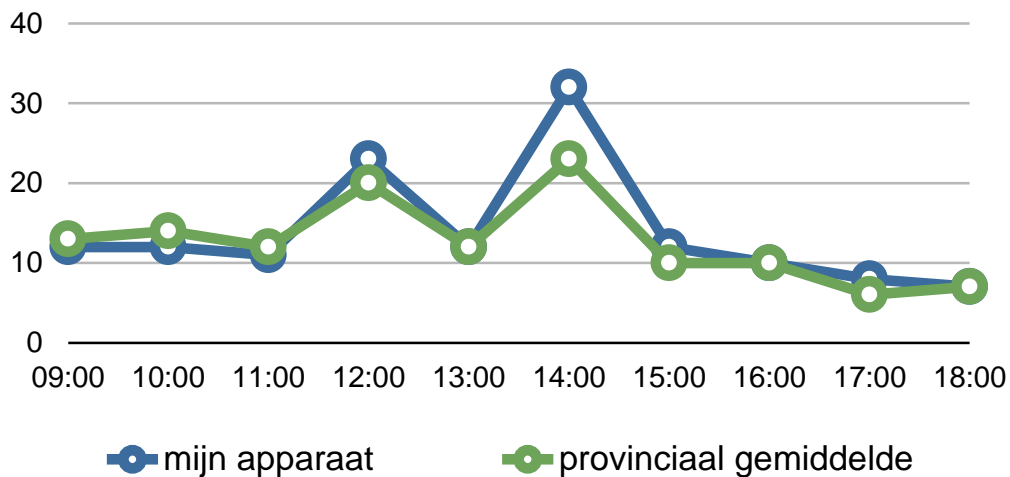
apparaat		09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
zonnepaneel op het dak	i (kW)	0,03	0,05	0,07	0,10	0,20	0,20	0,10	0,90	0,80	0,60
	$\sum i$ (kWh)	0,03	0,08	0,15	0,25	0,45	0,65	0,75	1,65	2,45	3,05
zonnepaneel in de tuin	i (kW)	0,03	0,02	0,02	0,03	0,05	0,10	0,20	0,20	0,10	0,08
	$\sum i$ (kWh)	0,03	0,05	0,07	0,10	0,15	0,25	0,45	0,65	0,75	0,83

apparaat		09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
windmolen in het park	i (kW)	9,00	9,00	9,00	8,00	7,00	6,00	4,00	4,00	5,00	9,00
	$\sum i$ (kWh)	9,00	18,00	27,00	35,00	42,00	48,00	52,00	56,00	61,00	70,00
windmolen op het dak	i (kW)	10,00	10,00	10,00	10,00	6,00	6,00	2,00	2,00	1,00	2,00
	$\sum i$ (kWh)	10,00	20,00	30,00	40,00	46,00	52,00	54,00	56,00	57,00	59,00

6. Demonstreer het resultaat aan de practicumdocent.

Opgave 3: data visualisatie

1. De meeste web-frameworks maken gebruik van het MVC-model. Op welke manier kan dit model ingezet worden om een wrapper om de database te maken die door de client kan worden gebruikt om database-calls uit te voeren?
2. Maak een dergelijke wrapper om de nrg-database die met ajax kan worden benaderd. Hierbij moet er alleen json over de lijn gaan. Het doel van deze wrapper is om een vergelijking tussen de opbrengst van de apparaten van de bezoeker te vergelijken met die van andere gebruikers. De client verstuurt een GET-request met het typenummer van het apparaat van de bezoeker en de vergelijking die zij wil maken:
 - ✓ de gemiddelde opbrengst per uur van alle apparaten van hetzelfde type;
 - ✓ de gemiddelde opbrengst per uur van alle apparaten van hetzelfde soort apparaat
3. Zorg ervoor dat bij deze vergelijking een postcodegebied kan worden opgegeven, zodat een vergelijking gemaakt kan worden met apparaten uit de hele provincie of alleen in de eigen buurt.
4. Maak gebruik van een javascript grafiek-framework (bijvoorbeeld chartjs, highchart, xkcdgraphs: zie jscharts.com voor voorbeelden en vergelijkingen) om deze vergelijking in een grafiek weer te geven. Zie onderstaande voorbeeld (let op: dit is een voorbeeld, het hoeft er uiteraard niet exact zo uit te zien). Geef aan waarom de keuze op dit javascript grafiek-framework is gevallen.



5. Wanneer de bezoeker over een meetpunt heen gaat, moet de exacte waarde van dat punt getoond worden.
6. Geef aan welke design-patterns er bij de implementatie van deze code gebruikt zijn en waarom ("geen enkele omdat het niet nodig was" is hierbij geen geldig antwoord).
7. Demonstreer het werk aan de practicumdocent.

Opgave 4: Access Control List

Deze opgave gaat verder in op de website van opgave 2 en 3. Mocht deze niet of onvoldoende zijn afgerond, dan volstaat het om een eenvoudige dummy-site hiervoor te gebruiken.

1. Wat is ACL en waarin verschilt dit van RBAC?
2. Maak gebruik van de ACL plugin van het toegewezen framework om de volgende groepen en activiteiten te definiëren:
 - ✓ Een bezoeker kan gemiddelde waarden van apparaten bekijken
 - ✓ Een gebruiker is een geregistreerde bezoeker. Zij is het, die de gegevens van een huishouden invult, apparaten beheert (toevoegt, verwijdert of wijzigt) en huisgenoten registreert. Zij kan maximaal vier huisgenoten toevoegen.
 - ✓ Gebruikers en huisgenoten kunnen beide metingen invoeren en vergelijkingen maken
 - ✓ Een beheerder kan gebruikers en huisgenoten en apparaten beheren.
3. Demonstreer de werking en de programmacode aan de practicumdocent.