Miniproject Pretpark – deel 4

Algemeen

- Maak je code leesbaar, onderhoudbaar en efficiënt.
- De namen van de jsp pagina's kies je zelf
- Zet in elke file je naam en studentennummer in commentaarlijnen

a. Uitbreidingen met sessie-attributen:

1. **ArrayListen opgevuld met bezoekers, personeelsleden en pretparken toevoegen aan de sessie**
   Voorzie op de beginpagina een link die we "Opvullen" noemen. Via deze link moet je naar een nieuwe servlet "OpvulServlet" gaan. Hierin moeten 3 ArrayListen worden opgevuld met de code die we je hieronder geven:

```
//personen en bezoekers
        ArrayList<Personeelslid> personeelsleden = new ArrayList<>();
        personeelsleden.add(new Personeelslid("Jitse", "Verhaegen"));
        personeelsleden.add(new Personeelslid("Bert", "De Meulenaere"));
        personeelsleden.add(new Personeelslid("Sanne", "Beckers"));
        ArrayList<Bezoeker> bezoekers = new ArrayList<>();
        bezoekers.add(new Bezoeker("Daan", "Mertens"));
        bezoekers.add(new Bezoeker("Wim", "Wijns"));
        bezoekers.add(new Bezoeker("Gert", "Pauwels"));
        bezoekers.get(0).voegToeAanWishlist("Weerwolf");
        bezoekers.get(0).voegToeAanWishlist("Vampire");
        bezoekers.get(1).voegToeAanWishlist("Glijbaan");
        bezoekers.get(1).voegToeAanWishlist("Vampire");
        bezoekers.get(1).voegToeAanWishlist("Jump");
        bezoekers.get(1).voegToeAanWishlist("Aquaglide");
        bezoekers.get(2).voegToeAanWishlist("Weerwolf");
//pretparken
        ArrayList<Pretpark> pretparken = new ArrayList<>();
        Pretpark pretpark1 = new Pretpark("Park 2020");
        Pretpark pretpark2 = new Pretpark("Jong en oud");
        Pretpark pretpark3 = new Pretpark("ThomasMoveMore");
        Attractie attractie1 = new Attractie("Vampire", 20);
        Attractie attractie2 = new Attractie("Weerwolf", 15);
        Attractie attractie3 = new Attractie("Cobra", 10);
        Attractie attractie4 = new Attractie("Glijbaan", 10);
        Attractie attractie5 = new Attractie("Jump", 20);
        Attractie attractie6 = new Attractie("Aquaglide", 10);
        attractie1.setFoto("vampire.jpg");
        attractie2.setFoto("weerwolf.jpg");
        attractie3.setFoto("cobra.jpg");
        attractie1.setVerantwoordelijke(personeelsleden.get(0));
        attractie2.setVerantwoordelijke(personeelsleden.get(1));
        attractie3.setVerantwoordelijke(personeelsleden.get(2));
        pretpark1.voegAttractieToe(attractie1);
        pretpark1.voegAttractieToe(attractie2);
        pretpark2.voegAttractieToe(attractie3);
        pretpark2.voegAttractieToe(attractie4);
        pretpark3.voegAttractieToe(attractie5);
        pretpark3.voegAttractieToe(attractie6);
        pretparken.add(pretpark1);
        pretparken.add(pretpark2);
        pretparken.add(pretpark3);
```

Na het kopiëren en plakken van deze code, zal je heel veel rode lampjes zien. Doe de nodige imports om dit op te lossen…

Als je deze servlet oproept door in index.jsp te klikken op "Opvullen" dan heb je 3 ArrayListen opgevuld:

- een ArrayList<Personeelslid> personeelsleden met 3 Personeelslid-objecten
- een ArrayList<Bezoeker> bezoekers met 3 Bezoeker-objecten
- een ArrayList<Pretpark> pretparken met 3 Pretpark-objecten die elk 2 attractie-objecten bevatten.

Zorg er nu voor dat die 3 ArrayListen als attribuut worden toegevoegd aan de sessie. Als dit gebeurd is, dan ga je vanuit de OpvulServlet gewoon terug naar index.jsp.

2. Overzicht bezoekers + overzicht pretparken

1. Voorzie op de beginpagina een link die doorverwijst naar een pagina die een overzicht geeft van alle bezoekers in het session-attribuut. Toon van de bezoekers zowel hun naam (voor- en achternaam) als hun pretparkcode.
2. Voorzie op de beginpagina ook een link die doorverwijst naar een pagina die een overzicht geeft van alle pretparken in het session-attribuut. Toon van de pretparken enkel de naam.
3. Zorg op beide overzichtspagina voor een geschikte tekst wanneer de gebruiker de pagina zou oproepen zonder eerst de link "Opvullen" te hebben aangeklikt.

Als je je project tot hier volledig werkend hebt gekregen heb je alvast 12/20.
Maak zeker een kopie van dit werkend project en bewaar het apart.
In het vervolg van de opgave moeten werkende delen aangepast worden. Krijg je dan fouten waardoor er nog minder werkt dan tevoren, dien dan best je vorig werkend project in.

---

3. Bestaande pagina's aanpassen zodat deze gebruikmaken van de session-attributen

!!!Je zal de link "Opvullen" telkens moeten aanklikken als je je programma uitvoert zodat de ArrayListen netjes opgevuld worden en bestaan. Anders krijg je nullpointers…

1. Zorg ervoor dat een **nieuw personeelslid** dat aangemaakt wordt, ook wordt toegevoegd aan het session-attribuut met de personeelsleden. Het nieuw gecreëerde attribuut moet je nu niet meer toevoegen aan het request-object. In je pagina waarin je de gegevens over het nieuwe personeelslid toont, haal je gewoon het laatste element uit de betrokken ArrayList van de sessie op.
2. Zorg ervoor dat **een nieuwe bezoeker** die aangemaakt wordt, ook wordt toegevoegd aan het session-attribuut met bezoekers. In je pagina waarin je de gegevens over de nieuwe bezoeker toont, haal je ook weer gewoon het laatste element uit de betrokken ArrayList in de sessie op.
   Op de pagina van nieuwe bezoekers zorg je er nu ook voor dat de pretparknamen die getoond worden overeenkomen met de namen van de pretparken die in het session-attribuut zitten (zie a.1). De ArrayList van Strings die je eerder gebruikte zal je dus moeten vervangen. Gebruik de werkwijze met de index die we in de les over sessies hebben gezien zodat je weet welk pretpark de gebruiker gekozen had. Doordat je werkt met een ArrayList van Pretparken (ipv Strings) kan je nu ook meerdere bezoekers registreren voor hetzelfde pretpark. In de MaakServlet

zorg je ervoor dat een nieuwe bezoeker geregistreerd kan worden voor het gekozen pretpark en daardoor een correcte pretparkcode krijgen.
NB de ArrayList<String> waarin mogelijke attractienamen staan voor de wishlist laat je gewoon staan.

3. Zorg ervoor dat **een nieuw pretpark** dat aangemaakt wordt, ook wordt toegevoegd aan het session-attribuut met pretparken. In je pagina waarin je de gegevens over het nieuwe pretpark toont, haal je ook weer gewoon het laatste element uit de betrokken ArrayList in de sessie op.
   Op de pagina waar een nieuwe attractie wordt toegevoegd aan een nieuw te creëren pretpark toon je de achternaam en de voornaam (met spatie ertussen) van alle Personeelsleden in de ArrayList die als session-attribuut werd gemaakt (zie a.1). In MaakServlet zorg je ervoor dat dit personeelslid als verantwoordelijke voor de attractie wordt vastgelegd.
   - Als je de EXTRA uitbreiding van deel 3 hebt gemaakt, wijzig je deze code zodat je in de jsp-pagina geen ArrayList<Personeelslid> meer moet opvullen maar gebruik kan maken van het session-attribuut. Zo ook in de Servlet
   - Heb je de EXTRA uitbreiding nog niet gemaakt, volg dan voor de opbouw van de dropdown en het toewijzen van het gekozen personeelslid als verantwoordelijke voor de attractie de werkwijze we in de les over sessies gezien hebben.

   Op de detailpagina waarin je vervolgens de naam van het pretpark toont met zijn attracties zorg je nu dat in de lijst van alle attracties naast de naam en de foto nu ook de **duur** en de **verantwoordelijke** (achternaam spatie voornaam) getoond wordt.

4. **Uitbreiding / wijziging overzicht en detailpagina pretparken:**

Maak linken van de namen van de pretparken die je toont in het overzicht van pretparken (zie a.2.2). Elk van deze linken moet naar de MaakServlet gaan en geeft daarbij ook de overeenkomstige index in de ArrayList door. Zorg ervoor dat je daarna dezelfde detailpagina kan hergebruiken als degene die je aan het einde van a.3 gebruikt om de details van het gekozen pretpark te tonen.
(Dit kan je doen door de index als attribuut toe te voegen aan de request en in de detailpagina altijd de index op te halen om te weten welk pretpark getoond moet worden. Na het maken van een nieuw pretpark (zie a.3.3) zal je daarom ook de index van het laatste element moeten doorsturen…).

5. Attracties zoeken en aanpassen

Op de beginpagina moet je nu ook een **zoekmogelijkheid** voorzien. Iemand kan dan de naam van een attractie ingeven en als deze naam een bestaande attractie is binnen één van de pretparken, dan kom je (via de MaakServlet) op een pagina terecht waar je van die attractie alle informatie behalve de verantwoordelijke kan zien én aanpassen (dus naam attractie, duurtijd en foto).
Tips:

- Doorloop alle pretparken en maak gebruik van de bestaande methode in de klasse Pretpark om deze functionaliteit te maken. Als je geen enkele attractie met die naam gevonden hebt, stuur je de gebruiker naar een pagina waar je

een gepaste foutboodschap toont met een link om terug naar index.jsp te gaan.

- Als je wel een attractie met ingegeven naam gevonden hebt, zet deze dan in de sessie anders zal je de wijzigingen niet kunnen aanbrengen aan de gevonden attractie. Als de gebruiker op de wijzigingspagina alle gewenste gegevens heeft aangepast, worden deze in de MaakServlet gebruikt om de informatie over de attractie aan te passen. Daarna stuur je de gebruiker na terug naar de index-pagina.

Als dit goed werkt, kan je ook de mogelijkheid geven om de verantwoordelijke aan te passen. Je laat een dropdownlist zien van alle personeelsleden maar je zorgt ervoor dat het personeelslid dat al als verantwoordelijke werd ingegeven voor de attractie geselecteerd is. Dit doe je door het attribuut "selected=selected" te zetten bij de juiste option.

Als je je project tot hier volledig werkend hebt gekregen, dan heb je 17/20!
Wil je 20/20 werk dan ook onderstaand deel uit.

---

b. Uitbreidingen algemeen:

1. **PretparkMatch:**
   We voegen een functionaliteit toe waarbij we per bezoeker gaan bekijken welk pretpark het meest geschikt is voor deze bezoeker.
   a. Voeg in de klasse Pretpark een methode **pretparkMatch(Bezoeker bezoeker)** toe die als resultaat een geheel getal (**int**) teruggeeft.
      i. Om dit resultaat te berekenen zal bekeken moeten worden hoeveel attracties uit de wishlist van de bezoeker voorkomen in het pretpark. Hiervoor zal je moeten itereren over alle (String) items van de wishlist en voor elk van deze items moeten bekijken of deze als attractienaam voorkomen in het pretpark. Hiervoor kan je gebruik maken van de reeds bestaande methode zoekAttractieOpNaam. Als resultaat geef je terug hoeveel procent van de items van de wishlist aanwezig zijn in het pretpark. Wanneer de gebruiker bijvoorbeeld 3 items op zijn wishlist heeft staan en deze zijn alle 3 aanwezig in het pretpark dan geeft de methode 100 terug. Wanneer er maar 1 van de 3 items aanwezig is in het pretpark geeft de methode 33 terug enzovoort. Opgelet! Wanneer de wishlist van de bezoeker leeg is, geef je in elk geval de waarde 0 terug. Doe je dit niet, dan zal je een fout krijgen ("java.lang.ArithmeticException: / by zero")
   b. Pas nu het overzicht van de bezoekers zo aan dat per bezoeker naast zijn naam en voornaam ook voor alle pretparken zijn "pretparkMatch" score wordt weergegeven.

   Als je na het opvullen van de ArrayListen het overzicht opvraagt, zou je dit resultaat moeten krijgen

Alle bezoekers

Bezoeker MERTENS daan met pretparkcode -1

Dit zijn je pretparkMatchen:

Park 2020 : 100

Jong en oud : 0

Westelpark : 0

Bezoeker WIJNS wim met pretparkcode -1

Dit zijn je pretparkMatchen:

Park 2020 : 25

Jong en oud : 25

Westelpark : 50

Bezoeker PAUWELS gert met pretparkcode -1

Dit zijn je pretparkMatchen:

Park 2020 : 100

Jong en oud : 0

Westelpark : 0

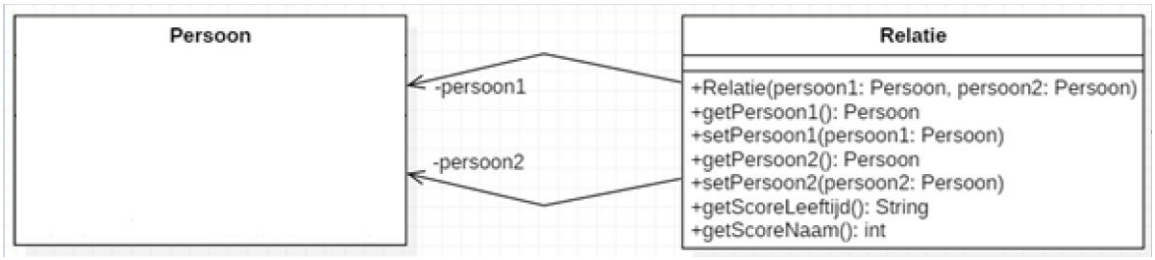
2. **PretparkDate:**

We voegen een functionaliteit toe waarbij we kunnen bekijken wat de kans is dat een pretparkdate tussen 2 geregistreerde bezoekers goed afloopt.

   a. Hiervoor voegen we eerst het attribuut geboortejaar (geheel getal) met bijhorende getter en setter toe aan de klasse Persoon. In de reeds aanwezige constructoren wordt dit geboortejaar standaard op 2000 gezet.

   b. Voeg aan de jsp pagina waarop een bezoeker kan worden ingegeven een mogelijkheid toe om het geboortejaar in te geven en doe hiervoor ook de nodige aanpassingen in de Servlet

   c. Maak een nieuwe klasse Relatie aan in de package fact.it.www.beans met twee associaties naar de reeds bestaande klasse Persoon



- de methode ***getScoreLeeftijd()*** berekent de relatiescore op basis van het leeftijdsverschil. Als de leeftijd van de jongste persoon in de relatie groter is dan de helft van de leeftijd van de oudste + 7 geef je volgende tekst terug:

  "*naam1* en *naam2* zijn voor elkaar gemaakt"

  Indien dat niet het geval is geef je volgende tekst terug:

"*naam1* en *naam2* blijven best gewoon vrienden"

naam1 en *naam2* vervang je door de namen van de betrokken personen.

Weetje: Deze berekening is ook gekend als de "Standard Creepiness Rule"

Tip: in de Java API kan je terugvinden dat je met *Year.now().getValue()* het huidige jaar kunt opvragen

- de methode **getScoreNaam()** berekent de relatiescore op basis van de namen (=voornaam + familienaam) van de betrokken personen. Hiervoor gebruiken we volgende formule:
((lengte van kortste naam) * 100/(lengte van langste naam)).

- Voeg aan de indexpagina een link toe "Pretparkdate" die leidt naar een pagina waarop 2 dropdownlists getoond worden die elk alle geregistreerde bezoekers bevatten. Verder voorzie je een knop om de berekeningen op naam en op leeftijd uit te voeren.

- Wanneer de gebruiker 2 bezoekers en op de knop drukt, moet het volgende gebeuren:

    - Er wordt dan met deze 2 Bezoeker-instanties een Relatie-instantie aangemaakt. Dit Relatie-object geef je door aan een volgende jsp pagina waarop beide relatiescores getoond worden.
    - Vb. voor Input: Bezoeker Nicole Josy met geboortejaar 1947 en Bezoeker Hugo Sigal met geboortejaar 1947 waarbij zowel berekening op naam als berekening op geboortejaar gevraagd wordt krijgen we bijvoorbeeld volgend resultaat te zien.

Op basis van de naam passen jullie voor 90% bij elkaar

Nicole Josy en Hugo Sigal zijn voor elkaar gemaakt!

Of volgende output als je de bestaande bezoekers aanklikt:

Op basis van de naam passen jullie voor 72% bij elkaar!

Daan Mertens en Wim Wijns zijn voor elkaar gemaakt

Veel succes!