### GSO3 Opdracht 5 - Internetbankieren

**Inleveren in week 17 via Canvas:**

Het volledige Netbeans-project nadat alle opdrachten zijn uitgevoerd. Zorg ervoor dat bij iedere klasse, interface en unittest de naam van de verantwoordelijke auteur (@author) vermeld staat.

**Opdrachten week 14**

Op SharePoint staat een gedeelte van een bankapplicatie gereed. Deze applicatie ga je de komende weken uitbreiden met unittesten, het observer ontwerppatroon en een centrale bank. Deze week maak je hiermee een begin:

1. Creëer binnen NetBeans een nieuw project (van het type JavaFX FXML Application). Gebruik de aangeleverde broncode in BankierenNoObserverFX.zip. Verplaats deze code naar de sourcemap van het nieuwe project.
2. Bestudeer de code en maak een klassendiagram van de gegeven code in Visual Paradigm. Het is niet nodig om alle velden en methoden in het klassendiagram op te nemen, het gaat met name om de compositie-relaties en associaties.
3. Schrijf een unittest voor Bank m.b.t. de methoden uit IBank; ga in deze test voorlopig uit van de situatie dat er slechts één bank is.
4. In Core Java (volume 1, edition 9) is het probleem van *race condition* beschreven en aan de hand van een voorbeeld geïllustreerd. Het verkozen voorbeeld was die van het overmaken van de ene bankrekening naar de andere bankrekening. Hoe voorkom je dit soort van corruptie van gedeelde informatie voor de Bank-klasse? En hoe zit het met het aanmaken van een nieuwe bankrekening? Verbeter de Bank-klasse.
5. Maak een klassendiagram en een componentendiagram voor toepassing van het observer-patroon om het banksaldo zichtbaar te maken op het bankiersessiescherm (zie ook opdrachten week 15).

**Opdrachten week 15**

Het doel van deze opgave is om meer ervaring op te doen met het observer-patroon waarmee een lossere koppeling tussen GUI en domein tot stand kan worden gebracht.

De bankierapplicatie functioneert NIET correct bij het overboeken van de ene rekening naar de andere rekening. Het geldbedrag wordt overgemaakt, van de rekening waarvoor de klant is ingelogd, naar de rekening van de begunstigde, maar de applicatie toont NIET het nieuwe saldo bij de ingelogde klanten van de twee betrokken rekeningen. Door opnieuw in te loggen kun je zien dat het banksaldo wel is aangepast.

1. We staan op het punt een kleine aanpassing te doen op een ‘goed’ werkende applicatie. Om niet steeds handmatig te verifiëren of onze aanpassingen ongewenste neveneffecten hebben beginnen we met het maken van een aantal nieuwe Unittesten. Je hebt al een Unittest geschreven voor Bank m.b.t IBank. Schrijf nu ook een test voor Balie m.b.t. IBalie en BankierSessie m.b.t. IBankiersessie. Zorg ervoor dat ook deze testen een goede dekking van testgevallen biedt; vergeet uiteraard niet de IllegalArgumentException’s wegens overschrijden precondities en alle andere excepties te testen.
2. Pas het observer-patroon toe om het banksaldo zichtbaar te maken op het bankiersessiescherm. De weg van RemotePropertylListener naar RemotePublisher en vice versa is nu wel een stuk ingewikkelder, want het patroon loopt nu via meer schijven. Als er iets aan één of meer van de interfaces moet veranderen, heeft dat natuurlijk ook consequenties voor de bijbehorende unittesten; die moet je dan natuurlijk ook aanpassen.
3. Let op: een bankiersessie eindigt bij het uitloggen of wanneer je langer dan 10 minuten geen actie richting de internetbalie onderneemt. Bij het beëindigen van de bankiersessie moet de “listener” ontkoppeld worden aan de internetbalie. Tijdens de testfase kan 10 minuten wachten nogal lang zijn. Wijzig dan tijdelijk de constante in IBankierSessie naar bijv. 60000.
4. Demonstreer de aangepaste applicatie aan de docent en geef uitleg.

**Opdrachten week 16 – week 17**

In deze opgave ga je de bankierapplicatie uitbreiden met een centrale bank waardoor het mogelijk wordt om geld over te boeken van de ene naar de andere bankrekening van verschillende banken. Onderdelen 3 en 4 zijn vereist indien je een beoordeling G(oed) wilt krijgen:

1. Specificeer de interfaces tussen de bank en centrale bank m.b.v. JavaDoc.
2. Bouw de bankierapplicatie met de centrale bank.
3. Optioneel: maak een component-test voor de centrale bank (zie slides week 15).
4. Optioneel: maak een integratie-test voor de bank en de centrale bank (zie slides week 15).
5. Demonstreer de eindapplicatie aan de docent en geef een toelichting.