**Fout 1:**

**Wat heb je getest ?**

De functie **UpdatePremiumForNoClaimYears**, Hierbij kwam ik achter dat als je hoger dan 5 schadevrije jaren hebt het eindbedrag altijd eindigde op 0, wat uiteraard niet de bedoeling is.

**Wat had je verwacht?**

Ik verwachte dat de functie zou werken zoals verwacht. Als je de functie uitvoert met een waarde lager dan 5, dan zie je meteen dat de berekening wel klopt. Hieruit kon ik halen dat de functie niet correct werkte naar behoren.

**Waar komt deze verwachting vandaan:**

Namelijk dat het wel werkt met een waarde lager dan 5, je zou denken dat het dan ook zou werken als je bijvoorbeeld 6 invoert, maar dit was niet het geval.

**De oplossing:**

De NoClaimPercentage Interger was het probleem, wanneer je deze omzet in een double dan werkt de functie boven de 5 namelijk wel en uiteraard onder de 5 ook. Dit komt omdat je uiteindelijk de (double) premium \* het bedrag doet wat een interger is. Hierdoor weet de functie niet wat hij moet doen en geeft hij altijd null mee terug.

**Fout 2:**

**Wat heb je getest** ?

De functie **PremiumCalculation** waarbij ik heb gekeken naar het object policyholder in de if statement. Het rijbewijs jaar moet kleiner zijn dan 5 jaar, dus niet gelijk staan aan 5.

**Wat had je verwacht?**

Ik verwachte bij mijn test dat als het rijbewijs 5 jaar in bezit was dat de functie de premie \* 1.15 NIET deed. Dit was niet het geval. Hij had namelijk wel \* som gedaan bij het premie bedrag.

**Waar komt deze verwachting vandaan:**

In de documentatie stond er als de policyholder zijn rijbewijs 5 jaar in bezit heeft dat hij onder de korting valt , dus dan wordt het premie bedrag niet \* 1.15 gedaan. In de code staat er “=<”. Hierdoor krijgt iemand met 5 jaar een rijbewijs alsnog geen korting.

**Oplossing:** Deze is best vanzelf sprekend, namelijk de “=” weghalen in de if statement. Hierdoor zijn de mensen die 5 jaar zijn rijbewijs hebben en ouder zijn dan 23 wel korting bij hun premie.

**Fout 3:**

**Wat heb je getest?**

De functie CalculateBasePremium, hierbij was de expected outcome een decimaal getal maar de actual outcome waren halen getallen.

**Wat had je verwacht?**

Aangezien we een berekening doen in euro’s , verwachte ik dat het geen hele bedragen waren maar decimaal getallen.

**Waar komt deze verwachting vandaan:**

Ik heb de berekening nagelopen met een rekenmachine en zelfs daarin kwam ik op decimaal getallen. Niet eens 2 maar soms wel meer dan 2. Hierdoor ging ik wel ervanuit dat dit erin moest. Als het bedrag 81,99 was dan was het alsnog 81, Hij ronde het dus ook niet af naar 82.

**De oplossing:**

De fix zat hem in Vehicle.cs. Hierin hebben we fields met Integers, dit is een beetje het zelfde geval als fout 1. Omdat de result een double is in CalculateBasePremium, moeten de datatypes in de rekensom ook een double zijn. Aangezien alle fields in Vehicle.cs INTS zijn gaat er iets fout in de code. Alleen nu hebben we 3 waardes in plaats van 1. Dus je kan de laatste waarde in de som een double maken, hierdoor wordt alles omgezet in een double en krijg je getallen met decimalen te zien.

**Fout 4:**

**Wat heb je getest?**

De functie **PremiumCalculation**, **CalculateBasePremium**, **UpdatePremiumForNoClaimYears**. Ik wou controleren of het bedrag werd afgerond naar 2 decimalen.

**Wat had je verwacht?**

Aangezien de premie in euro’s ga ik er van vanuit dat alles met 2 decimalen moet.

**Waar komt deze verwachting vandaan:**

Als je de applicatie runt dan is alles ook met 2 decimalen, hierdoor ga ik ervanuit dat het zo moet.

**De oplossing:**

Ik heb Math.Round in elke functie toegevoegd, hierdoor eindigt alles met 2 decimalen.

**Unit testen :**

**PolicyHolderTest:**

**Het doel van de unittest:**

Het doel van de test is om te controleren of de ingangsdatum van het rijbewijs goed wordt berekend door het huidige jaartal min de ingangsdatum te doen.

**Waarom heb je voor deze specifieke (data) invulling van de unittest hebt gekozen:**

Door de ingevulde data kon ik zien of er geen fout zat in de if statement, dus iemand 23 jaar is en langer/korter zijn rijbewijs had en andersom ook. Ook wou ik checken of er een bedrag in de min kon komen(bijvoorbeeld -1).

**Welke technieken je gebruikt hebt om tot deze data te komen:**

xUnit en Theory .

**VehicleTest:**

**Het doel van de unittest:**

Het doel was om t controleren of het basispremie goed werd berekend met 1 voertuig en verschillende voertuigen.

**Waarom heb je voor deze specifieke (data) invulling van de unittest hebt gekozen:**

Door de ingevulde data kon ik zien of het verkeerd werd berekend met verschillende voertuigen en andere waardes. Zo kon ik 100% controleren dat elke voertuig klasse goed berekend werd.

**Welke technieken je gebruikt hebt om tot deze data te komen:**

xUnit en Theory/Fact.

**InsuranceCoverageTest:**

**Het doel van de unittest:**

Het doel is om te kijken of alle verzekeringen per jaar en per maand goed berekend werd. Denk aan WA, WAPLUS en ALLRISK.

**Waarom heb je voor deze specifieke (data) invulling van de unittest hebt gekozen:**

De ingevulde data is gekozen omdat ik puur de focus wou leggen op de maand en jaar berekeningen, alle informatie erin is willekeurig. Het was overzichtelijk voor mij om te waardes hetzelfde te houden zodat ik zelf ook kon inschatten of het bedrag realistisch was.

**Welke technieken je gebruikt hebt om tot deze data te komen:**

xUnit en Fact.

**PremiumCalculationTest:**

**Het doel van de unittest:**

Het doel is om te controleren of de applicatie alle functies/methodes goed draait en uiteindelijk ook de juiste premie waarde terug krijgen in jaar en maand. Ook is het van belang dat het bedrag eindigt op 2 decimalen.

**Waarom heb je voor deze specifieke (data) invulling van de unittest hebt gekozen:**

Omdat ik met deze waardes elke categorie wel dek, iemand die 30 jaar is en wel of niet 5 schade vrije jaren heeft opgebouwd. Hetzelfde als een person die 21 jaar is en ook wel of niet 5 schade vrije jaren heeft opgebouwd. Met deze data kon ik zekerweten dat er niet ergens een fout zat.

**Welke technieken je gebruikt hebt om tot deze data te komen:**

xUnit en Theory (Equivalentieklassen).