***Reflectiedocument ( Wat ik heb aangepast)***

*Inleiding*

In dit verslag beschrijf ik de verbeteringen en aanpassingen die ik heb uitgevoerd aan het verkiezingssysteemproject om een beter resultaat te behalen. Ik streef ernaar om hierdoor 2 extra punten te verdienen ten opzichte van mijn vorige resultaat. Voorheen had ik een score van 13/20, maar ik geloof dat de implementatie van de D’Hondt methode in mijn project de score zou kunnen verhogen naar 15/20. Ik heb mijn uiterste best gedaan om dit onderdeel zo perfect mogelijk uit te leggen. Dit reflectieverslag richt zich op de implementatie van de D'Hondt methode voor de zetelverdeling en de aanpassingen in de andere klassen, ook de html en css opmaak, om de uitslagen op een overzichtelijkere manier weer te geven. In dit verslag leg ik alle veranderingen en verbeteringen uit.

**Implementatie van de D'Hondt methode**

Mijn belangrijkste verandering aan dit project was de implementatie van de D'Hondt methode voor het verdelen van zetels. D’Hondt methode zorgt ervoor dat de zetels in verhouding worden verdeeld op basis van het aantal stemmen dat elke lijst heeft ontvangen. Hieronder volgt een beschrijving van de stappen die ik heb genomen om dit te realiseren.

* **Berekening van Quotienten:** Voor elke ronde wordt het aantal stemmen voor elke lijst gedeeld door het aantal zetels dat de lijst al heeft plus 1. Dit quotient wordt berekend en de lijst met het hoogste quotient zal de zetel voor die ronde winnen.
* **Toewijzing van Zetels:** Het proces wordt herhaald totdat alle zetels zijn verdeeld. De berekening zorgt ervoor dat de zetels worden verdeeld over de lijsten op basis van het aantal ontvangen stemmen.

**De code voor de D'Hondt methode:**

Afbeelding met tekst, schermopname, software, Lettertype

Automatisch gegenereerde beschrijving

De code implementeert de D'Hondt methode voor de verdeling van zetels bij de verkiezing. Hieronder leg ik stap voor stap uit wat elke regel van de code doet en waarom dit overeenkomt met de D'Hondt methode:

*Definitie van de functie:*



Deze functie verdeelt zetels over de gegeven lijsten op basis van de D'Hondt methode. De parameter lijsten is een lijst van lijsten, en totaal\_zetels is het totale aantal zetels dat verdeeld moet worden.

*Initialiseren van stemmen en zetels:*



* **stemmen:** Dit creëert een dictionary ‘stemmen’ waar de naam van elke lijst gekoppeld is aan het totale aantal stemmen dat die lijst heeft ontvangen
* **zetels:** Ook een dictionary ‘zetels’ waarin voor elke lijst (op basis van de naam) het aantal zetels wordt bijgehouden. In het begin is dat 0 voor elke lijst.

*Zetelverdeling met behulp van D’Hondt methode:*

Afbeelding met tekst, schermopname, scherm, software

Automatisch gegenereerde beschrijving

Deze buitenste for loop, loopt een aantal keer gelijk aan het totaal aantal beschikbare zetels. Elke iteratie wordt 1 zetel toegekend.

Voor elke zetel die verdeeld moet worden (for \_ in range(totaal\_zetels)), wordt het volgende gedaan

* max\_quotient wordt geïnitialiseerd op 0 en lijst\_met\_max\_quotient op None. Ze worden gereset aan het begin van elke iteratie.
* Voor elke lijst wordt het quotient berekend met de formule: quotiënt = stemmen[lijst.naam] / (zetels[lijst.naam] + 1). Dit quotient is het aantal stemmen gedeeld door het aantal al toegewezen zetels plus 1. Dit is eigenlijk de kern van de D’Hondt methoden. Als het berekende quotient groter is dan het huidige maximale quotient, wordt het maximale quotient geupatet en word de lijst met het hoogste quotient bijgehouden.
* De lijst met het hoogste quotient wint de zetel voor die ronde. Als dit quotient groter is dan max\_quotient, wordt max\_quotient bijgewerkt en wordt lijst\_met\_max\_quotient ingesteld op de naam van deze lijst.
* Nadat alle lijsten zijn doorlopen, wordt een zetel toegevoegd aan de lijst met het hoogste quotient (zetels[lijst\_met\_max\_quotient] += 1).

Daarna retourneren we de zetelverdeling. De regel return zetels, retourneert de zetelverdeling na het verdelen van alle zetels volgens de D’Hondt methode.

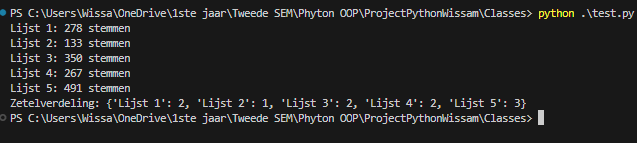
*Waarom ik denk dat dit klopt en werkt*

Ik ben zeker dat dit klopt en werkt, omdat bij de D'Hondt methode wordt voor elke lijst een quotient berekend om te bepalen welke lijst de volgende zetel krijgt. Het quotient wordt berekend door het totale aantal stemmen van een lijst te delen door het aantal zetels dat de lijst al heeft gekregen plus 1. De quotient wordt kleiner als een lijst meer zetels krijgt, wat ervoor zorgt dat de verdeling van de zetels eerlijk blijft.

In elke ronde wordt de zetel gegeven aan de lijst met het hoogste quotient. Vervolgens wordt het quotient van die lijst aangepast door het aantal zetels te verhogen met 1, waardoor het quotient voor de volgende ronde kleiner wordt. Dit proces wordt herhaald totdat alle zetels zijn verdeeld.

Deze methode voorkomt dat lijsten met een groot aantal stemmen en aantal zetels krijgen. Door het quotient aan te passen, wordt het moeilijker voor een lijst om zetels te blijven winnen terwijl ze meer zetels krijgen, wat resulteert in een eerlijkere en gebalanceerde verdeling. Het doel van de D’Hondt methode is om de zetels zo eerlijk mogelijk te verdelen. Partijen met meer stemmen zouden meer zetels moeten krijgen, maar het systeem zorgt ervoor dat elke partij een kans heeft om zetels te winnen.

Ik heb een andere .py bestand aangemaakt waar ik D’Hondt methode heb getest.



Zoals we kunnen zien de zetelverdeling volgens de D’Hondt methode werkt zoals verwacht. Dit met verschillende verdelingen van zetels op basis van het aantal stemmen dat elke lijst heeft ontvangen.

Bij mijn project krijgt elke lijst 2 zetels. Het is geen fout maar een kenmerk van de gegeven stemmenverdeling en de D’Hondt methode. De D’Hondt methode is dus correct geïmplementeerd in mijn project.

Ik heb de classes Chipkaart, USBStick en Stemcomputer ook aangepast. De rest van de classes zijn min of meer hetzelfde gebleven. Hieronder leg ik uit wat ik heb aangepast:

*Class Chipkaart*

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

Het controleert of een chipkaart geautoriseerd is om te stemmen.

*Class stemcomputer:*

Afbeelding met tekst, schermopname, software

Automatisch gegenereerde beschrijving

Deze functie zorgt ervoor dat een kiezer alleen kan stemmen als de chipkaart geautoriseerd is en niet eerder heeft gestemd. Controleert dus of de kiezer al heeft gestemd. Als de kiezer al in de lijst van gestemde kiezers staat, wordt een bericht weergegeven dat de kiezer al gestemd heeft.

Het controleert of de chipkaart van de kiezer al geautoriseerd is via de USB stick. Als de chipkaart geautoriseerd is wordt er verder gecontroleerd of de chipkaart geïnstalleerd is.

De USB stick en chipkaart worden gereset na het stemmen. Er wordt een foutbericht weergegeven als de chipkaart niet is geïnstalleerd en geautoriseerd.

*Class lijstengenerator*

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Ik heb de namen hier ook geïmplementeerd, omdat het makkelijker is om te volgen. De functie genereert lijsten met kandidaten. Dus het genereert 5 lijsten met elk tien willekeurige kandidaten voor de stemproces. Voor elke kandidaat wordt een voornaam en een achternaam willekeurig gekozen uit vooraf lijsten. De genereerde lijsten worden geretourneerd wat het mogelijk maakt om de simulatie te voldoen voor de verkiezing.

**Andere aanpassingen in de HTML**

Ik heb mijn output style veranderd. Dit is nu meer overzichtelijker, eerder was dit niet zo. Ik heb de achtergrond veranderd naar de Belgische vlagkleuren met een lineaire gradiënt: zwart, geel en rood.

De inhoud van de pagina is gecentreerd in een .content

De tabellen voor lijststemmen, voorkeurstemmen en zetelverdeling zijn gescheiden en duidelijk gemarkeerd met kopteksten. Ik heb iconen erin gevoegd. Leek me goed om dit erbij te doen. Ik heb dit gedaan met behulp van de online tools. Ik moest een iconenbibliotheek zoals de Font Awesome integreren. Dus ik heb eerst een Font Awesome geopend. Dit doe je met de link. Dan de link voegen naar de Font Awesome. Heel makkelijk om te doen. Dan voeg je de iconen aan de HTML.

Nu ziet mijn output er zo uit:

