

Cheaters Be Gone

Voor elk tournament van een dobbelspel worden alle dobbelstenen onderworpen aan een test. Hierbij wordt de dobbelsteen een aantal keer geworpen en wordt telkens bijgehouden hoeveel ogen er worden geworpen. Dit wordt typisch in een frequentietabel gedaan. Dat is een array waar de index het aantal ogen (-1 natuurlijk omdat een dobbelsteen vanaf 1 start en een array vanaf 0) voorstelt en de waarde op die index het aantal keer dat werd geworpen.

Een goeie dobbelsteen geeft evenveel kans om elk aantal ogen te werpen, namelijk 1 op het aantal dobbelsteen vlakken.

Stel je hebt bijvoorbeeld volgende resultaat van het gooi-experiment: `[51, 52, 47, 53, 48, 72]`

Dan kan je stellen dat kans om een 1 te gooien $51 / \text{TOTAAL_WORPEN} = 51/323 = 0.1578$, wat ongeveer $1/6$ is.

De class `DieChecker` vormt het begin van een validatie class die nagaat of er met de dobbelsteen niet werd geprutst. Volgende implementatie dient te worden voorzien:

- `int CountTotalRolls(int[] frequencyTable)`: deze methode krijgt de frequentietabel als input en bepaalt hoe dikwijls er in totaal werd gegooid.
- `double[] DetermineThrowChances(int[] frequencyTable)`: deze methode krijgt de frequentietabel als input en bepaalt de kans voor elk aantal ogen op basis van het experiment.
 - Return een array van `double` waarden die telkens de gooikans voorstelt.

Puntenverdeling: 1/2 voor `CountTotalRolls` en 1/2 voor `DetermineThrowChances`

Verwachte Werking

De `Main()` methode voorziet reeds een kleine demo applicatie die je kan uitvoeren. Indien je alles implementeert zoals verwacht, zou je volgende output moeten krijgen van de applicatie:

```
Welcome to the Official Die Validation App

Die 1 throw chances: 0.165% 0.165% 0.165% 0.169% 0.162% 0.173%
Die 2 throw chances: 0.080% 0.166% 0.172% 0.153% 0.153% 0.276%

Thank you for using 'Cheaters be Gone'
```

Unit Tests

Dit project bevat ook unit tests. Je kan deze uitvoeren via `Test => Run All Tests`.