

Programmeertechnieken & Testen

Philippe De Winne

1 OEFENING

1.1 OPGAVE

Als een bal weerkaatst, bereikt hij drie vierden van zijn vorige hoogte. Schrijf een programma dat de beginhoogte en een eindhoogte inleest, en dat bepaalt hoeveel weerkaatsingen de bal gemaakt heeft de laatste keer dat hij nog de eindhoogte bereikte.

Veronderstel dat begin- en eindhoogte beiden positief zijn, en dat de beginhoogte hoger is dan de eindhoogte.

1.2 UITDAGING

- Controleer dat de beginhoogte hoger is dan de eindhoogte.
- Ik wil 10 keer kaatsen (aantal in te geven), en ik geef een eindhoogte in. Hoe hoog moet de bal dan starten?
- Hou rekening met de zwaartekracht (dus een elastische botsing) ☺

2 OEFENING

2.1 OPGAVE

Maak een programma dat aan de gebruiker z'n lengte en gewicht vraagt en vervolgens z'n berekende BMI (Body Mass Index) toont. Gebruik `Math.Round` om de uitkomst tot maximum 2 cijfers na de komma te tonen.

Reken na met je rekenmachine of je uitkomst wel degelijk klopt!

2.2 UITDAGING

Pas je BMI-programma aan zodat je programma feedback aan de gebruiker geeft naargelang de berekende BMI.

- De kleur tussen haakjes geeft aan in welke kleur je deze tekst zet:
 - Onder de 18,5 (rood): ondergewicht
 - Tussen de 18,5 en de 24,9 (groen): normaal gewicht.
 - Tussen de 25 en de 29,9 (donkergeel): overgewicht. Je loopt niet echt een risico, maar je mag niet dikker worden.
 - Tussen de 30 en de 39,9 (rood): Zwaarlijvigheid (obesitas). Verhoogde kans op allerlei aandoeningen zoals diabetes, hartaandoeningen en rugklachten. Je zou 5 tot 10 kg moeten vermageren.
 - Boven de 40 (magenta): ernstige zwaarlijvigheid. Je moet dringend vermageren want je gezondheid is in gevaar (of je hebt je lengte of gewicht in verkeerde eenheid ingevoerd)
- Hoeveel kilo moet je afvallen of aankomen om 'normaal gewicht' te hebben. Hou dus rekening met de ondergrens en de bovengrens. Test dit uit door verschillende lengtes en gewichten in te geven.

3 OEFENING

Ontwerp een consoletoepassing waarmee je een Belgisch ondernemingsnummer kan controleren. Een Belgisch ondernemingsnummer is als volgt opgebouwd: BE 0xxx.xxx.xxx waarbij BE de landcode is die standaard gevolgd wordt door een spatie en dan een 0. Wanneer je de volgende 7 cijfers restdeelt door 97, dan is het getal gevormd door de laatste 2 cijfers gelijk aan 97 minus de rest van deze gehele deling.

Bij een geldig ondernemingsnummer verschijnt de tekst "Geldig ondernemingsnummer.", bij een ongeldig ondernemingsnummer "Ongeldig ondernemingsnummer.". Indien het ondernemingsnummer niet is ingegeven volgens de standaardopbouw vermeld in deze tekst, dan verschijnt in de console de tekst "Geef het ondernemingsnummer in volgens de juiste format.".

4 OEFENING

Schrijf een methode die, voor een gegeven natuurlijk getal, de som van de delers van dit getal (het getal zelf niet inbegrepen) berekent.

Bijvoorbeeld:

- voor 12 wordt dat $1 + 2 + 3 + 4 + 6 = 16$
- voor 36 wordt dat $1 + 2 + 3 + 4 + 6 + 9 + 12 + 18 = 55$

Schrijf vervolgens een programma dat; gebruik makend van deze methode, alle 'perfecte' getallen tussen 2 en 100 uitschrijft. Een natuurlijk getal wordt 'perfect' genoemd als de som van zijn delers (het getal zelf niet inbegrepen) gelijk is aan het getal zelf. Bijvoorbeeld, 6 is een perfect getal want $6 = 1 + 2 + 3$.

5 OEFENING

5.1 OPGAVE DIERENTUIN

Maak een console-applicatie waarin je een zelfverzonnen abstract klasse Dier in een List kunt plaatsen. Ieder dier heeft een gewicht en een methode Zegt (die abstract is) die het geluid van het dier in kwestie op het scherm zal tonen. Maak enkele childklassen die overerven van Dier en uiteraard de Zegt methode overriden.

Vervolgens vraag je aan de gebruiker wat voor dieren er in deze lijst moeten toegevoegd worden. Wanneer de gebruiker 'q' kiest stopt het programma met vragen welke dieren moeten toegevoegd worden en komt er een nieuw keuze menu. Het keuze menu heeft volgende opties:

- Dier verwijderen , gevolgd door de gebruiker die invoert het hoeveelste dier weg moet uit de List.
- Diergewicht gemiddelde: het gemiddelde van alle dieren hun gewicht wordt getoond
- Dier praten: alle dieren hun Zegt() methode wordt aangeroepen en via WriteLine getoond
- Opnieuw beginnen: de List wordt leeggemaakt en het programma zal terug van voor af aan beginnen.

Probeer zo modulair mogelijk te werken.

We gaan de dieren in zelfverzonnen categorieën steken. Voorzie per categorie een extra feature... wees creatief.

Voorzie per soort een manier om op te slaan met welk andere dier hij kan/wil communiceren.

Vb:

- een olifant wil praten met een leeuw en een tijger
- een leeuw wil met niemand praten
- een tijger wil praten met een olifant en een leeuw

Dus:

- `olifant3.CommuniceerMet(leeuw3) -> output = eerst maak je het geluid van een olifant, waarop de leeuw antwoord "Laat me gerust"`
- `olifant3.CommuniceerMet(tijger1) -> output = eerst maak je het geluid van een olifant, en daarna het geluid van een tijger`

Voorzie de juiste interfaces om dit te implementeren.

Test dit uit.