Objectgeoriënteerd Programmeren: WPO 5 (interfaces)

1. Inhoud

Interfaces

2. Oefeningen

- A: Sensoren
- A: LED
- A: Resistance
- E: Sound

2.1 A: Sensoren

In dit programma ga je het gedrag van een aantal sensoren simuleren. Schrijf eerst de interface ISensor waarin je volgende methoden en/of properties voorziet:

- double CurrentValue();
- double Minget;
- double Maxget;

Implementeer deze interface in de klassen thermometer, barometer, anemometer en luxsensor. Deze sensoren worden allemaal uitgelezen via de methode CurrentValue(). Het gedrag van elk van deze sensoren wordt hieronder opgelijst.

- Een thermometer levert waarden uit tussen -40°C en +100°C. Omdat de temperatuur langzaam toeneemt of daalt, mag deze met 2°C verschillen t.o.v. de vorige uitlezing. Gebruik hiervoor een randomgenerator die je aan de huidige temperatuurswaarde optelt (in de methode zelf). De randomgenerator gebruik je tussen -2 en +3 (3 uitgesloten).
- Een barometer levert waarden op tussen 960 hPa en 1050 hPa. De luchtdruk verandert ook zeer traag en mag met maximaal 3 hPa per uitlezing veranderen. Gebruik hiervoor dezelfde methode als voor de thermometer.
- Een anemometer levert waarden op tussen 0 $\frac{m}{s}$ en 50 $\frac{m}{s}$. Door de vele windstoten kan het gedrag volledig willekeurig zijn. Hierdoor mag je bij elke oproep een randomgetal tussen 0 en 51 (exclusief) afleveren.

- De luxsensor levert waarden op tussen 0 lx (lux) en 250 klx op. Hierbij komt 250 klx overeen met de zonnestraling in een zandwoestijn en 0 lx komt overeen met een volledige donkere ruimte. Ook hier kan het gedrag willekeurig zijn. Gebruik hierbij dus een randomgetal tussen 0 en 250001.

Bij elk van deze klassen wordt enkel de numerieke waarde geretourneerd. Via een methode ToString() kan je de huidige waarde tesamen met de eenheid teruggeven. Deze gebruik je om de huidige waarden in textboxen weer te geven. Declareer alle objecten volgens het type ISensor, maar instantieer ze volgens de gewenste kindklasse. Gebruik een timer om de objecten op regelmatige tijdstippen aan te spreken.

2.2 A: LED

In de meeste elektronicaprojecten worden LED's gebruikt. Hoewel er vele varianten commercieel beschikbaar zijn, berusten LED's allemaal op hetzelfde principe. In deze opgave ga je een aantal LED's tekenen. Schrijf een interface ILed waarin je volgende eigenschappen voorziet:

- void Draw(Canvas cvs);
- double Intensity

Implementeer deze interface in de klassen RLed (rood), GLed (groen) en BLed (blauw). Via de methode draw wordt een ronde schijf in het midden van de canvas getekend (50 pixels in diameter). Via de property kan je de intensiteit van de huidige LED aanpassen. Declareer alle objecten volgens het type ILed, maar instantieer ze volgens de gewenste kindklasse.

2.3 A: Resistance

In dit programma ga je een aantal elektronische componenten emuleren. Schrijf eerst een interface IResistance waarin je volgende methoden en/of properties voorziet:

- void setVoltage(double volt);
- double getCurrent();
- double maxPower

Schrijf de kindklassen resistor, varistor en diode. Elk van deze klassen implementeert uiteraard de volledige interface. Elk van deze implementeert de methoden getCurrent en setVoltage als volgt:

- Bij de weerstand schaalt de stroom gelijk met de spanning volgens de wet van Ohm. Als weerstand mag je 100 Ω nemen. Vanaf dat het vermogen over 1 W gaat, brandt de weerstand door en kan er nooit meer stroom door de weerstand vloeien.
- Een varistor biedt een oneindige weerstand zolang de spanning onder 8.2 V blijft. Vanaf 8.2 V neemt de stroom lineair toe met de spanning volgens de wet van Ohm (vereenvoudiging). De weerstandswaarde bedraagt dan 22 Ω. Bij een vermogen van 2 W brandt de varistor door waardoor er permanent een open keten ontstaat.

- Een diode zal pas in geleiding komen vanaf 0.6 V. Vanaf dan neemt de stroom exponentieel (volgens $I = V^3$) toe met de spanning. Vanaf een spanning van 0.85 V brandt de diode door waardoor een open keten ontstaat.

Programmeer de 3 klassen en de interface. Declareer de objecten tot het type IResistor, maar instanieer ze volgens 1 van de 3 kindklassen. Eenmaal een object doorgebrand is, is de stroom erdoor 0. Gebruik een tekstbox als input (samen met een button) waarmee je de objecten gaat testen.

2.4 E: Sound

Voor deze opgave mag je gerust de opdracht "Waves" van een voorgaand WPO kopiëren en van dat project vertrekken. In de echte wereld bestaan er naast geluidsgolven met een periodisch karakter ook geluidsgolven die een randomgedrag vertonen. Schrijf een programma waarin je 4 geluidsgolven gaat produceren. Schrijf de interface ISound waarin je de basiseigenschappen van een golf opneemt:

- De maximale amplitude van de geluidsgolf,
- De tekenfunctie Draw(Canvas cvs);

Erf deze interface over in de abstracte klasse van de opdracht Waves. Nadien schrijf je een nieuwe (gewone) klasse die de interface onmiddelijk implementeert. Deze klasse zal rondomgeluiden produceren. Dit doe je a.d.h.v. randomgetallen. Je voorziet voor elke x-coördinaat een ander randomgetal (y-coördinaat). De maximale waarde van de randomgetallen bevinden zich tussen -amplitude en +amplitude. Met de methode draw wordt dit randomgeluid onmiddelijk op de canvas getekend. Dit geluid wordt in het jargon ook "noise" (ruis) genoemd.