

Software Ontwikkeling Labo 2: Object Oriented Programming

Algemene afspraken labo's

- a) op het einde van het labo worden de uitgewerkte oplossingen op de dropbox van minerva geplaatst.
- b) de niet uitgewerkte oplossingen worden zelfstandig uitgewerkt tegen het volgende labo en ingediend op het einde van het volgende labo (tenzij anders vermeld wordt).
- c) noteer ook telkens de tijd die je aan elke oefening gewerkt hebt in de code van de oplossing. Dit zal voor een klein deel bijdragen aan de finale punten.

Oefeningen

1. Willekeurige getallen

Vraag een maximum getal aan de gebruiker. Schrijf daarna 10 willekeurige getallen tussen 0 en dit maximum op het scherm. Je kunt hiervoor de klasse `System.Random` gebruiken. Vergeet de initialisatie van de getallengenerator niet.

2. Invoer van getallen

Nu je zelf met objecten kunt werken kan je ook zelf invoer van de gebruiker verwerken. Schrijf 2 methoden, een voor gehele getallen en een voor kommagetallen. Voorzie ook de nodige parameters om de gebruiker correct om input te kunnen vragen (melding als het getal niet binnen de correcte range (min/max) valt, vragen om het opnieuw te proberen, etc.). Probeer ook zinnige standaardwaarden op te geven voor de meeste parameters (de minimale en maximale waarden van het integer type voor het minimum en maximum bijvoorbeeld).

Vergeet niet je nieuwe methoden te testen.

Tip: gebruik de methoden `int.TryParse()` en `double.TryParse()` voor de omzetting in plaats van de meer standaard `Parse()` methoden. Zo hoef je nog geen rekening te houden met excepties.

3. Complexe getallen

Maak de benodigde objecten om complexe getallen voor te stellen, en zorg dat je met die complexe getallen ook kan rekenen (de 4 basisbewerkingen). Alle informatie over het uitwerken van berekeningen met complexe getallen kan je vinden op http://en.wikipedia.org/wiki/Complex_number.

Voor de uitwerking van dit probleem apart op de volgende twee manieren uit (die kunnen in eenzelfde uitwerking):

- a) alle bewerkingen worden in statische methodes uitgevoerd en het resultaat wordt opgeslaan in een nieuw complex getal.
- b) alle bewerkingen worden direct op een complex getal uitgevoerd. Het resultaat wordt dan ook opgeslaan als nieuwe waarde in het complex getal waarop de berekening wordt uitgevoerd.

Zorg ervoor dat de complexe getallen niet kunnen aangepast worden extern aan de klasse zelf.

Zorg dat een complex getal kan ingegeven worden in de vorm (Re ; Im) waarbij het reële en het imaginaire deel beide een kommagetal of gewoon getal kunnen zijn. Kijk zeker op MSDN naar meer documentatie over het verwerken van tekst.

4. Domotica

Voor een nieuw te ontwikkelen domotica systeem moet alle benodigde informatie op de juiste manier aan elkaar gekoppeld worden. De domotica stuurt toestellen en lichtpunten aan, de lichtpunten kunnen aan en uitgeschakeld worden. De toestellen hebben typisch een sturing die lineair verloopt en waarvan de sturingmaximum en minimum bij eerste gebruik kunnen ingesteld worden (bijvoorbeeld Oven: 0 – 250 graden). Er zijn verschillende gebruikers die kunnen schakelen (een persoon heeft hiervoor een gekoppelde smartphone). De lichtpunten worden tenslotte ook nog bijgehouden per kamer in het huis.

Teken dit probleem eerst uit in een UML schema. Het is hierbij absoluut niet nodig om alle data en function members op te sommen in het UML schema.

Zorg ervoor dat je de toestellen en lichten kan aansturen, en dat je dit vanuit de main-functie kan uitvoeren. Doe dit op een zo logisch mogelijke manier.

5. Kaartspel

In deze oefening maken we een model van een kaartspel. Dit betekent dat we de volgende structuren moeten opstellen en koppelen

- a) De 52 kaarten in het spel
(we gaan hier uit van een standaard kaartspel zonder joker)
- b) De spelers rond de tafel en de kaarten die ze in hun hand houden
- c) Een ronde waarin elke speler een kaart op de tafel legt
- d) Een evaluatie-object dat in staat is om een ronde te beoordelen en punten te verdelen naar de verschillende spelers

Implementeer een eenvoudig spel dat volledig automatisch kan gespeeld worden. De kaarten hiervoor worden bij het begin allemaal aangemaakt. Deze kaarten worden op een willekeurige manier aan het aantal spelers verdeeld (kies in dit geval voor 4 spelers). De spelers zullen elke ronde een willekeurige kaart uit hun stapel halen en die op tafel leggen.

Evaluatie gebeurt op het einde van alle rondes op een van de volgende manieren:

- a) Degene die de hoogste kaart gelegd heeft wint 1 punt
- b) De speler met de hoogste kaart wint 2 punten, de speler met de tweede hoogste kaart wint 1 punt, de speler met de derde hoogste kaart verliest 1 punt en de speler met de laagste kaart verliest 2 punten.

Zorg ervoor dat het spel volledig automatisch loopt bij het opstarten en dat het daarbij output in het volgende formaat genereert:

Voorbeeld output voor een spel
Een nieuw spel kaarten wordt gegenereerd (h)arten, (r)uiten, (k)laveren, (s)choppen, (J)ack, (Q)ueen, (King)
De kaarten worden verdeeld Speler 1: h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, hJ, hQ, hK Speler 2: r1, r2, r3, r4, r5, r6, r7, r8, r9, r10, rJ, rQ, rK Speler 3: ...
Het spel wordt gespeeld Ronde 1: h1, r1, k1, s1 Ronde 2 ...
Het spel wordt op de eerste manier geevalueerd Ronde 1: Speler 1 +1 Ronde 2: ...
De finale score op de eerste manier is Speler 1: 13 Speler 2: ...
Het spel wordt op de tweede manier geevalueerd ...