TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

TW1090 Inleiding Programmeren
Tentamen
25 januari 2016, 9:00 – 12:00 uur

Normering: Opgave 1 t/m 3 ieder 6 punten. Score: totaal aantal punten + 2. Cijfer: score / 2

Voorbereiding

Lees eerst de instructies op het aparte Instructie blad (Inloggen, IDLE starten, pdf van het boek openen).

Gebruik bij de implementatie van de opgaven alleen de modules / bibliotheken die nu al door middel van imports worden geïmporteerd. Voeg dus geen import statements toe. Verander ook de hoofdprogramma's niet, indien al een hoofdprogramma is gegeven. Voeg dan alleen implementaties toe van de gevraagde functies en test deze functies in het hoofdprogramma. Vul je naam en studienummer ook in het commentaar van de programma's in (zie de inhoud van de .py files). Open in de IDLE omgeving de .py file die je wilt gaan editen. Zorg ervoor dat je steeds maar 1 .py file tegelijk geopend hebt. Save je .py files regelmatig.

Opgave 1 (punten: a: 2, b: 3, c: 1)

Implementeer in het programma **`A1.py`** drie functies.

a) `Turns(n1, n2):`

Bepaalt het kleinste gehele getal *n* zodat $\frac{n_1 n}{n_2}$ geheel is.

b) `CreateCurve(n1, n2):`

Bepaalt punten op de kromme die gegeven is door

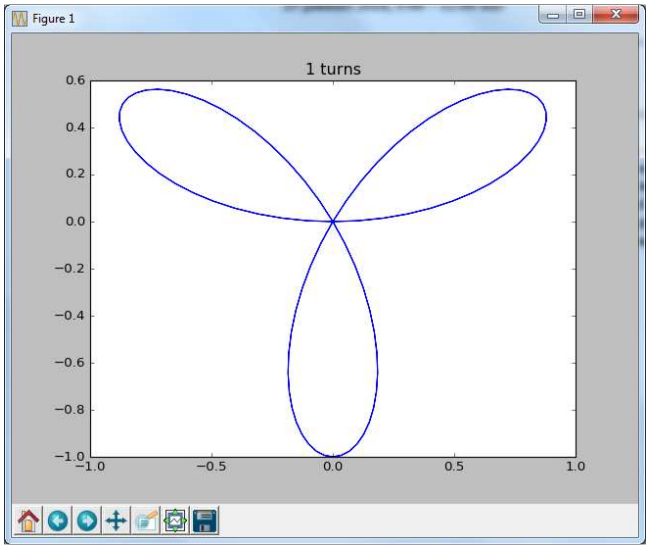
$$x = r\cos\theta \qquad y = r\sin\theta \qquad \text{met } r = \sin\left(\frac{n_1}{n_2}\theta\right)$$

Start bij $\theta = 0$. Laat θ toenemen met stapgrootte $\frac{\pi}{90}$ radialen (2 graden). Geef arrays (of lijsten) met respectievelijk x- en y-coördinaten van de punten op de kromme terug. Geeft tevens de waarde van n terug (zie hieronder). Stop op een moment dat je zeker weet dat je een keer terug bent gekomen bij het begin van de kromme. Je hebt in ieder geval de gehele kromme gehad als θ het interval $[0, 2\pi n]$ doorloopt, waarbij n het gehele getal is dat je met de `Turns` functie bepaalt.

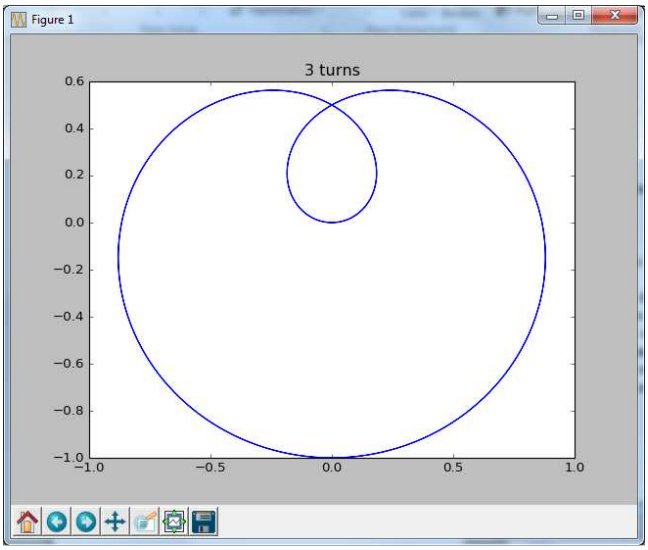
c) `DrawCurve(x, y, n):`

Tekent de kromme, bepaald door de punten in de arrays (of lijsten) met x- en y- coördinaten, in een matplotlib window. Laat de titel van de plot zijn "n turns" met voor n de met de `Turns` functie berekende waarde ingevuld. Bij n1 = 8 en n2 = 5 is de titel bijvoorbeeld: "5 turns".
Test de werking van de functies met het testprogramma in **`A1.py`**. De volgende bladzijde geeft voorbeelden van de uitvoer in de Python shell en de geplotte kromme bij een aantal waarden van n1 en n2.

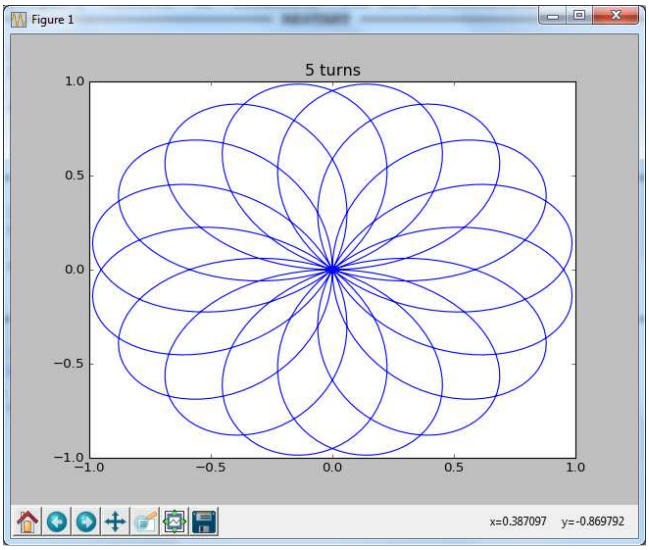
```
>>>
Turns(9, 3) is: 1
Turns(3, 9) is: 3
Turns(8, 5) is: 5
Turns(23, 37) is: 37
Turns(9, 48) is: 16

n1 = 9
n2 = 3
>>>
```



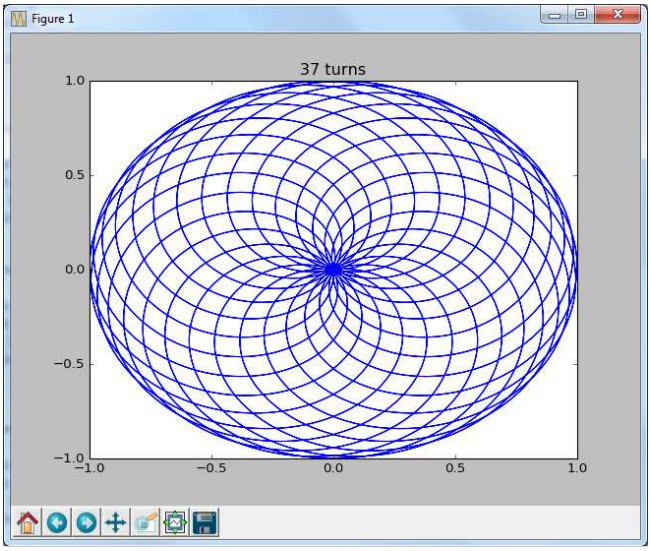
n1=9; n2 = 3



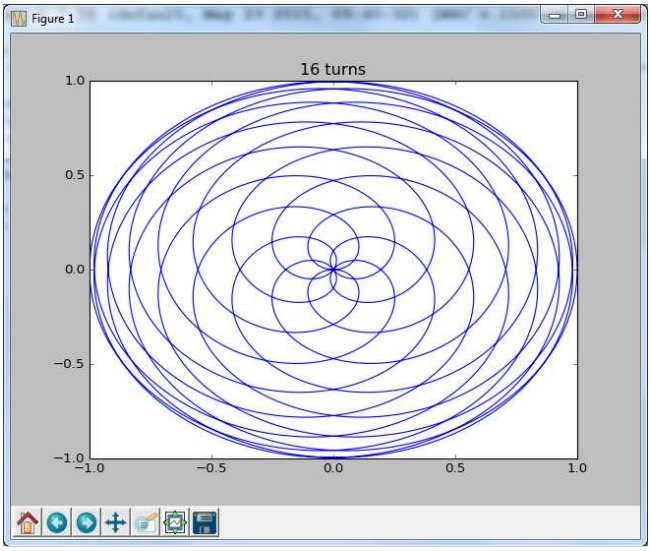
n1 = 3; n2 = 9



n1 = 8; n2 = 5



n1 = 23; n2 = 37



n1 = 9; n2 = 48

Opgave 2 (punten: a: 2, b: 4)

In het programma **`A2.py`** worden gerichte grafen gerepresenteerd op de manier zoals op de collegeslides van lecture 7, dus als een dictionary van dictionaries. Knoopnamen zijn strings. Bijvoorbeeld:

```
g = {'1': {'2': 1, '3': 2, '4': 2},
     '2': {'1': 1, '3': 1},
     '3': {'4': 1},
     '4': {'1': 2, '2': 1}}
```

Voor de leesbaarheid zijn in de bovenstaande weergave van de dictionary van dictionaries extra regelovergangen opgenomen, maar de dictionary is natuurlijk identiek aan:

```
g = {'1': {'2': 1, '3': 2, '4': 2}, '2': {'1': 1, '3': 1}, '3': {'4': 1}, '4': {'1': 2, '2': 1}}
```

De graaf `g` heeft pijlen van knoop `'1'` naar knoop `'2'` van lengte 1, van knoop `'1'` naar knoop `'3'` van lengte 2, van knoop `'1'` naar knoop `'4'` van lengte 2, van knoop `'2'` naar knoop `'1'` van lengte 1, enz.

a) Implementeer in het programma **`A2.py`** een functie `TotalLength(g)` die van een graaf `g` de som van de lengtes van alle pijlen berekent en het resultaat teruggeeft.

b) Implementeer tevens een functie `Graph(n)` die een random gerichte graaf met precies n knopen genereert en teruggeeft. Voor ieder tweetal knopen n1 en n2 van de graaf is er met gelijke kansen:
   - Geen pijl van n1 naar n2
   - Een pijl van lengte 1 van n1 naar n2
   - Een pijl van lengte 2 van n1 naar n2

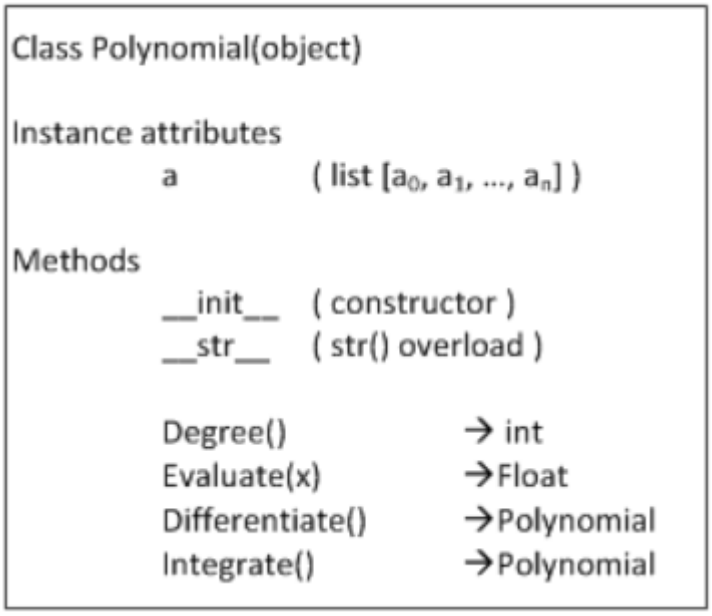
   Het is dus mogelijk dat er voor een tweetal knopen n1 en n2 zowel een pijl van n1 naar n2 als ook een pijl van n2 naar n1 in de graaf aanwezig is. De graaf mag geen pijlen bevatten van een knoop naar zichzelf.

Hieronder volgt voorbeelduitvoer van het testprogramma. Natuurlijk zullen de grafen 3, 4 en 5 bij runnen anders zijn dan in de voorbeelduitvoer, want deze grafen zijn random grafen.

```
>>>
g is
{'1': {'3': 2, '2': 1, '4': 2}, '3': {'4': 1}, '2': {'1': 1, '3': 1}, '4': {'1': 2,
'2': 1}}
The total length of the arcs in graph g is: 11
The total length of the arcs in an 'empty' graph is: 0
The total length of the arcs in a graph
with 3 nodes and without arcs is: 0

Graph(3) is
{'1': {'3': 1, '2': 2}, '3': {'1': 1, '2': 2}, '2': {'1': 2, '3': 2}}

Graph(4) is
{'1': {'3': 2, '4': 2}, '3': {'1': 2, '4': 1}, '2': {'3': 2, '4': 2}, '4': {'1': 2,
'3': 1, '2': 1}}

The total length of the arcs of graph
Graph(5) is
{'1': {'3': 2, '2': 2, '4': 2}, '3': {'1': 1, '5': 2}, '2': {'1': 1, '3': 2, '5': 2},
'5': {'1': 2, '3': 2, '2': 2, '4': 2}, '4': {'2': 1, '5': 2}}
is 25

>>>
```

Opgave 3 (punten: a: 1, b: 2, c+d: 3)

Een polynoom van graad n is van de vorm $P(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x^1 + a_0$ met $a_n \neq 0$. Uitzondering hierop vormt het polynoom $P(x) = 0$ (met $n = 0$ en $a_0 = 0$). Dit polynoom heeft graad 0.
In programma **`A3.py`** vind je een opzet van een klasse voor polynomen van willekeurige (eindige) graad. Het instance attribuut `a` bevat een lijst met de coëfficienten $a_0, a_1, ..., a_{n-1}, a_n$ van het polynoom. De `__init__` en de `__str__` methoden zijn al geïmplementeerd. De coëfficient a_n mag geen 0 zijn, want dan zou deze zijn weggelaten en was het polynoom van graad $n-1$ geweest in plaats van graad n. Het gegeven programma test de klasse. Indien je het programma probeert te runnen dan komt er een foutmelding. De reden daarvan is dat er een aantal methoden die worden aangeroepen nog niet in de klasse zijn geïmplementeerd. De klasse moet na aanvulling met de nu nog ontbrekende methoden voldoen aan de beschrijving in het klassendiagram hieronder.



Implementeer de methoden;

a) `Degree()`: Deze methode geeft de graad van een polynoom terug. We houden hierbij de volgende regels aan:
   $P(x) = c$ heeft graad 0. Let op: Dit moet ook zo zijn als $c = 0$, terwijl in dit geval het `a` attribuut van het polynoom gelijk is aan `[]`.
   $P(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x^1 + a_0$ met $a_n \neq 0$ heeft graad n
b) `Evaluate(x)`: Deze methode geeft voor een polynoom (zeg P) de waarde van $P(x)$ terug, d.w.z. een reëel getal.
c) `Differentiate()`: Deze methode geeft de afgeleide van een polynoom terug. Het polynoom waarvoor `Differentiate` wordt aangeroepen verandert niet.
d) `Integrate()`: Deze methode geeft een primitieve van een polynoom terug met integratieconstante 0 (d.w.z. $a_0 = 0$). Ook deze methode verandert het polynoom waarvoor de methode wordt aangeroepen niet.

Test de methoden door runnen van het testprogramma en aanroepen van de te testen methoden in de Python shell na runnen van de klasse. Hieronder is de gewenste uitvoer van het testprogramma gegeven.

```
>>>
Test function Degree
n(x) = 0
Degree of n(x): 0

p(x) = 4.0x^3+3.0x^2+2.0x^1+1.0
Degree of p(x): 3

q(x) = 1.0x^7+1.0x^6+8.0x^5+1.0x^4+0.0x^3+1.0x^2+3.0x^1+1.0

Test function Evaluate
p(-2) = -23
q(3) = 4960

Test function Differentiate
q1(x) = 7.0x^6+6.0x^5+40.0x^4+4.0x^3+0.0x^2+2.0x^1+3.0

Test function Integrate
Q(x) = 0.125x^8+0.143x^7+1.333x^6+0.2x^5+0.0x^4+0.333x^3+1.5x^2+1.0x^1+0.0
>>>
```

Afsluiten

Sluit alle .py bestanden. Controleer in IDLE door middel van Kies File | Open nog een keer extra of je in je directory **Homedrive(H:)** drie .py bestanden hebt staan met in de naam jouw naam en studienummer. **Deze drie bestanden vormen het werk dat je inlevert.**

Druk nu op het sluithokje rechts bovenin het IDLE window. Sluit op deze manier alle Editor en Python Shell windows van IDLE. Sluit ook de pdf van het boek. Kies tenslotte Start | Exit | OK. Je wordt nu uitgelogd. Dit duurt vrij lang.

AAN DE HAND VAN UITSLUITEND DE DRIE .PY BESTANDEN WORDT JE WERK BEOORDEELD.

INDIEN DE .PY BESTANDEN NIET OP HOMEDRIVE(H:) STAAN MAAR OP DE DESKTOP, DAN HEB JE GEEN WERK INGELEVERD.

einde opgaven