

Code : BFVP15INF2	Tentamen: Informatica 2 Oefententamen	
Datum:	Tijd:	School: ILST
Lokaal:	Klas: BFV1	Duur: 90 min
Docent : Tsjerk Wassenaar	Aantal opgaven: 4	
Hulpmiddelen: Overig hulpmiddelen:		
Opgaven inleveren: Nee Kladpapier inleveren: Nee		
Bijzonderheden: Computertentamen zonder gebruik van internet		
Naam student: Klas:	Studentnummer:	

Instructies

Opdrachten

Het tentamen bevat een aantal opdrachten. Let op,

- Iedere opdracht geeft een *executable* (!) programma.
- Schrijf elk programma met een correcte structuur.
- Geef elk programma een docstring met een korte beschrijving. Vermeld ook je **naam** en je **NAW nummer** in het programma.
- Sla de opdrachten op in de Desktop map, zodat je ze makkelijker kunt inleveren.

TIP: Programmeer per stap en kijk steeds wat er gebeurt. Gebruik idle3 of python3 in de terminal om stukjes code uit te proberen.

Inleveren

Lever de gemaakte opdrachten in door het programma `submit_your_work` uit te voeren met als argumenten de programma's. Open een terminal en voer de commando's uit:

```
cd Desktop
submit_your_work studentnummer bestand1 [bestand2 [...]]
```

Hierin is

studentnummer

je NAW nummer (!) zonder nullen aan het begin

bestand1 [bestand2 [...]]

de namen van de bestanden die je inlevert (minstens een)

Neem na het inleveren contact op met de surveillant; hij/zij laat je zien wat je hebt ingeleverd. Teken pas voor akkoord als je dit hebt gecontroleerd.

Beoordeling

Bij elke opdracht staat vermeld hoeveel punten maximaal te krijgen zijn. Het cijfer is gelijk aan $\max(10, \text{punten}/10)$. De opdrachten worden beoordeeld op de volgende punten:

1. Werkt het programma?
2. Doet het programma wat het moet doen?
3. Is de code goed (leesbaar, effectief)?

N.B. Lees eerst de opdrachten door en bedenk een strategie. Schrijf bijvoorbeeld eerst een template en beschrijving voor elke opdracht. Blijf niet te lang hangen op één opdracht.

Opdracht 1: template.py (18 punten)

- Schrijf een complete programma/module template. (12)
- Als de template als programma wordt gedraaid zonder argumenten, dan moeten de docstring (`__docstring__`) en de auteur geprint worden, anders de argumenten. (6)

Opdracht 2: pdb2spheres.py (30 punten)

Schrijf een programma dat een PDB bestand (met een eiwitstructuur) omzet naar een lijst met 'spheres' (coördinaten, grootte en kleur) die gerendered zouden kunnen worden met pypovray. Laat het programma het eerste en laatste element van de lijst printen.

- Gebruik voor het schrijven en testen het bestand **1cts.pdb** dat de structuur van citraat synthase bevat (een eiwit uit de citroenzuur cyclus).
- Importeer en gebruik de module **chem** (chem.py) die is meegeleverd. Daarin staan de groottes (radii) van de atomen.

Let op! Alleen regels die beginnen met 'ATOM' of met 'HETATM' bevatten coördinaten:

```
#ATOM      13  CA  SER A   3           87.614  40.697  48.804
#           |
# positie   13           30           38           46           54
```

Deze regel geeft als output:

```
[ 87.614, 40.697, 48.804, 2.0, 'grey']
```

- Schrijf in commentaar/pseudocode wat je programma moet doen en wat voor data je op elk moment hebt. Definieer de functies die je hierbij nodig hebt. **Voor deze deelvraag hoeft je geen code te schrijven** (15 punten)

- b) Implementeer de code. Gebruik de volgende kleuren: (C, grey), (O, red), (H, white), (N, blue), (S, yellow), (anders: magenta). (15 punten)

Opdracht 3: fixme.py (22 punten)

Het programma `fixme.py` bevat 11 fouten. Spoor de fouten op en verbeter ze. Schrijf steeds commentaar met uitleg wat er fout was.

Opdracht 4: pdb_analyzer.py (30 punten)

- a) Geef elke functie een docstring die uitlegt hoe de functie werkt. (10)
- b) Schrijf in ieder geval twee for-loops om naar (1) een list-comprehension en (2) een dict-comprehension. (10)
- c) Gebruik string formatting om de resultaten als tabel te presenteren. (10)

>>> EINDE TENTAMEN <<<