Deel 1: Introductie

Stap 1: Teaser

Inleiding.

Korte rondleiding door PyPi snoepwinkel, de “cheese-shop”.
Performance en gebruik van C++ modules.

Stap 2: Wat kun je na deze les

Een module-hiërarchie ontwerpen, met gebruikmaking van abstractie-lagen.
Mutual dependence voorkómen door gebruik van een *base*, *utils* en/of *constants module.*
Het module-zoekpad programmatisch aanpassen.
Een package samenstellen met gebruikmaking van *__init__.py* files.

Stap 3: Ophalen benodigde voorkennis

Gezond verstand als bron van voorkennis:

- Van exponentiële naar lineaire ontwerp-complexiteit (NIET runtime-complexiteit).
- Hoe vermijd ik een spaghetti-ontwerp (NIETspaghetti executie).
- Wenselijkheid relatieve paden bij copy-installation.
- Distributie van een groep samenhangende modules.

Deel 2: Kern

Stap 4: Uitleg van de nieuwe lesstof

Korte samenvatting van de bijbehorende videoles, gelegenheid tot vragen.
Hoe download ik met de *-m* switch PyPi packages in de juiste Python versie.
Hoe maak ik een package.
Hoe kan ik het package zoekpad vanuit code beinvloeden.

Stap 5: Voorbeelden en vragen door docent

Structuur van SimPyLC met een facade module en relatieve paden.
Faciliteiten in Transcrypt voor uploaden van Transcrypt naar PyPi en voor installatie.
Structuur van een mix tussen Python en C++ modules bij een medical imaging applicatie.
Waarom zou ik Python en C++ wel of niet integreren via bijvoorbeeld Swig?

Stap 6: Oefenen met de nieuwe lesstof

Teken (informeel) een package diagram van een jukebox programma met een main package dat gebruik maakt van een SQL database package voor metadata en .mp4 package voor audio, beiden met gebaseerd op hetzelfde onderliggende stream abstraction package dat zowel met local files als met cloud storage kan werken. Elk package is een blokje en de pijlen ertussen betekenen “gebruikt” (in tegenstelling tot “wordt gebruikt door”. Waarom is het wel of

niet handig om de namen van de streams in de SQL database op te slaan. Waar in dit ontwerp is sprake van abstractie.

Stap 7: Feedback op gemaakte oefening

Een of meerdere leden van elke groep delen hun programma, docent en medestudenten geven feedback, evt. na vragen om nadere uitleg.
Vraag: Waarom is het vaak (maar niet altijd) verstandig het gebruik van niet-lokale variabelen te vermijden?

Deel 3: Afronding

Stap 8: Evalueren of deze les goed “geland” is

Eén of meer leden van elke groep stellen vragen en/of geven tips en/of tops.

Stap 9: Huiswerk om je de lesstof verder eigen te maken

Zie opdrachten-tab in MS-Teams.

Inleveren van de uitgewerkte opdrachten die bij een les horen is, samen met een positief verlopen eind-assessment, een noodzakelijke voorwaarde voor een voldoende en dient uiterlijk 2 volle dagen voor de volgende les plaats te vinden, uitsluitend op de geëigende wijze in MS-Teams.

De resultaten worden deels in de volgende les, deels individueel besproken. Maak de opdrachten op het door jou gekozen niveau zo goed mogelijk, maar wees ook niet bang om fouten te maken. Het gaat erom dat je een serieuze poging waagt en de docenten je indien nodig kunnen helpen om verder te komen.

--//--