

Deel 1: Introductie

Stap 1: Teaser

Inleiding

Demo Mondriaan programma:

- Run vanaf command prompt
- Compileer en run in browser

Uitleg broncode Mondriaan:

- Functie “rect”, uitleg lokale variabelen
- Functie “draw”, uitleg stack frame voor recursie

Stap 2: Wat kun je na deze les

Functies met parameters en return values gebruiken om je broncode een overzichtelijke hiërarchische structuur te geven en dubbel werk te vermijden.

Parameters gebruiken om informatie een functie in en uit te transporteren.

Samengestelde return values van willekeurige complexiteit te gebruiken.

Named- en default-parameters gebruiken.

Een doorgegeven lijst gebruiken als losse parameters.

Doorgegeven losse parameters gebruiken als lijst.

Een doorgegeven dictionary gebruiken als losse named parameters.

Losse named parameters gebruiken als een dictionary.

Functies maken die met elke willekeurige mix van parameters zinvol kunnen worden aangeroepen.

Functies doorgeven als parameter.

Lambda functies definiëren en gebruiken.

Stap 3: Ophalen benodigde voorkennis

D(on't)R(epeat)Y(ourself) principe: “Luiheid is een vorm van intelligentie”.

Abstractie en hiërarchische decompositie.

Functies die iets doen, functies die iets uitrekenen en functies met side-effects.

Deel 2: Kern

Stap 4: Uitleg van de nieuwe lesstof

Korte samenvatting van de bijbehorende videoles, gelegenheid tot vragen.

Uitleg en demo van gebruik van functies als parameter.

Uitleg en demo van lambda functions.

Uitleg en demo over de praktische bruikbaarheid van recursie in niet-functionele talen.

Stap 5: Voorbeelden en vragen door docent

Review resultaat top-down functional design semi-automatische kraan Borealis.

Bespreken voordelen top-down vs. bottom-up en means-ends design.

Bespreken opbouw software voor beveiligings- en toegangscontrole RUG en “het probleem bij software”.

Bespreken natuurlijke recursie a.d.h.v. broncode topologisch correcte hart-segmentatie (i.t.t. surgical planning, kortsluiting via een geleidend vezeltje).

Stap 6: Oefenen met de nieuwe lesstof

Schrijf top-down een programma met de volgende 5 functies:

1. Functie *readNumbers* vraagt aan de gebruiker om een reeks getallen en bergt deze op in een lijst.
2. Functie *computeAverage* berekent het gemiddelde van deze getallen.
3. Functie *computeExtremes* berekent een tuple dat het hoogste en laagste cijfer bevat.
4. Functie *reportResults* drukt gemiddelde, hoogste en laagste cijfer af.
5. Functie *evaluateStatistics* maakt achtereenvolgens gebruik van de 4 bovenstaande functies en wordt in een herhalingslus aangeroepen in je programma.

Gebruik parameters om getallen een functie in te transporteren, en return values om de resultaten er weer uit te halen. Vermijd rechtstreeks gebruik IN een bepaalde functie, van variabelen die BUITEN die functie gedefinieerd zijn.

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.

--//--