

PoP evaluering hold 02 (Christoffer)

Funktionel programmering, OOP, F#

- Giv (bedre) begrundelse for hvorfor vi bruger F#.
- Begrund hvorfor vi starter med funktionel programmering og senere går over i imperativ/OOP. (Lav forventningsafstemmelse.)
- Sværhedsgraden steg meget kraftigt, da vi gik fra funktionel programmering til OOP.
- Savner forklaring af hvornår funktionel programmering skal bruges og hvornår OOP skal bruges.

Nødder

- Festligt med prisuddeling.
- Lav gerne en lidt større præmie og fortæl på forhånd hvad præmien er.
- Start med lidt lettere nødder og lad sværhedsgraden stige løbende. På den måde kan flere studerende være med på nødderne.
- Lav eventuelt en "nød light" og en "nød hardcore". Det kan give incitament til at udfordre studerende, der ikke er blandt de stærkeste.
- Udvid lektiecaféen med en nød workshop, som en instruktør afholder.

Forelæsninger

- Jons rumskib forelæsning var ret sej, men alt for kompleks. Tupler stod ikke så godt frem pga. al matematikken/fysikken.
- Torben skal forberede sin kode og droppe live kodning. Det fungerer meget bedre, når koden er på powerpoints og gennemgås eller på anden måde er forberedt.
- Der har været for mange forelæsninger, hvor der var problemer med projektoren uden, at det blev løst.

UML

- Gennemgangen til forelæsninger var dårlig.
- Kravene omkring UML diagrammer var for uspecificeret i afleveringer. Det burde stå mere tydeligt, hvilke typer diagrammer, der blev forventet. (Eventuelt også hvordan det forventes, at de bruges - min egen kommentar.)

Afleveringer

- Giv altid feedback - også til genafleveringer.
- Mere feedback på kode i rapport opgaver. Overvej at omstrukturere opgaverne så det bliver lettere for instruktører at give feedback på koden. Vi bør sikre, at de studerende får god kode stil i dette kursus.
- Informør i 10g om udvidelse i 12i.
- Fordel tiden til afleveringer bedre. Specielt var tiden sat af til 11g meget kort, mens tiden var lang til 10g.

Scratch

- Scratch mindre nødvendig når der også gennemgås pseudokode (imperativ programmering) i DMA.
- Scratch er godt til nybegyndere, men langhåret for erfarne. Vi brugte dog lidt for lang tid på det.
- Fungerede ikke så godt som introduktion til imperativ programmering.

Andet

- Godt med gennemgang af ugens emne til øvelsestid. (Især) også når det er repetition af ugens emne.
- Vær opmærksom på at slå ned på dårlig kode stil. (Fx kan det ikke være i orden at en gruppe består en aflevering, hvor 20x20 grid printes ud med 20 linjer print statements.)