

# Programmering og Problemløsning

## Datalogisk Institut, Københavns Universitet

### Uge(r)seddel 12 - individuel opgave

Martin Petersen, Jon Sparring og Christina Lioma

Deadline 19. januar

Forelæsningerne vil afslutte kurset. Vi vil behandle opsamlingsemner bla. polimorfi og evaluering

Til øvelserne på alm. skema forventer vi at I arbejder efter nedenstående plan.

**Mandag-Tirsdag 11-12/1:**

Der arbejdes med 11g.

**Fredag 15/1:**

Der arbejdes med 12i.

**Mandag-tirsdag 18/1-19/1:**

Der arbejdes med 12i.

Afleveringsopgaven er:

**12.1** Du skal implementere en udvidelse til SIMPLEJACK som indeholder en omstrukturering af nogle af klasserne, samt indførelse af en række nye strategier. Du skal simulere nogle SIMPLEJACK spil hvor du afprøver forskellige strategier, for at afgøre hvilken strategi som lader til at være den bedste.

Implementér super-klassen **Player**, og klasserne **Dealer**, **Human** og **AI** som nedarver fra **Player**. **Player** skal indeholde attributter og metoder som implementerer den fælles funktionalitet som alle tre typer "spillere" har, f.eks. en metode som vælger "Hit" eller "Stand".

Implementér super-klassen **Strategy**, samt en klasse for hver af følgende strategier, som alle nedarver fra **Strategy**

1. Vælg altid "Hit", medmindre summen af egne kort kan være 15 eller over, ellers vælg "Stand".
2. Vælg altid "Hit", medmindre summen af egne kort kan være 17 eller over, ellers vælg "Stand".
3. Vælg altid "Hit", medmindre summen af egne kort kan være 19 eller over, ellers vælg "Stand".

4. Vælg tilfældigt mellem "Hit" og "Stand". Hvis "Hit" er valgt, trækkes et kort og der vælges igen tilfældigt mellem "Hit" og "Stand" osv.
5. Følg strategi 2. hvis ét af egne kort er et Es, ellers følg strategi 1.

Simulér 3000 spil SIMPLEJACK med 5 AI spillere som følger de 5 ovenstående strategier. Dealer skal følge strategi 2. Konkluder hvilken af strategierne som lader til at være bedst.

Du skal også

- Opdatere dit UML-diagram
- Lave Unittests
- Kommentere ny kode jævnfør kommentarstandarden for F#

Afleveringsopgaven skal afleveres som både LaTeX, den genererede PDF, samt en fsx tekstfil med løsningen, som kan oversættes med fsharp, og hvis resultat kan køres med mono. Det hele skal samles i en zip fil efter sædvanlig navnekonvention:

`<instructor's-initial>_<firstname.lastname>_<exercise-number>.zip`

I zip filen skal en delopgave navngives ved opgavenummer, således at filen for opgave 12.1 hedder `opg12_1.fsx`.

God fornøjelse.