

CDIO del opgave 3

Indledning

IOOuterActive har fået endnu en opgave.

Kundens vision:

I skal udvikle et **Monopoly Junior** spil. Vurder hvad der er det vigtigste for at spillet kan spilles! Implementer de væsentligste elementer for at spillet kan spilles. I må **gerne** udelade regler - prioritér!

[Spilleregler på dansk](#)

[Spilleregler på engelsk](#)

[Spilleplade](#)

[Chancekort](#)

Nu har vi terninger og spillere på plads, men felterne mangler stadig en del arbejde. I dette tredje spil ønsker vi derfor at forrige del bliver udbygget med forskellige typer af felter, samt en decideret spilleplade.

Spillerne skal altså kunne lande på et felt og så fortsætte derfra på næste slag. Man går i ring på brættet.

Der skal nu være 2-4 spillere.

Analyse- og designdokumentation

Der skal udarbejdes følgende artifacts:

- Kravliste
- Use case diagram
- Eksempler på use case beskrivelser - vælg mindst én, der beskrives fully dressed
- Domænemodel
- Et eksempel på systemsekvensdiagram
- Et eksempel på sekvensdiagram
- Designklassediagram

Der er vigtigt, at der er tydelig sammenhæng mellem beskrivelser og diagrammer. Det skal således være muligt at trace fra kravliste til analysedokumentation, designdokumentation og implementering. Beskriv ligeledes jeres brug af GRASP-mønstre.

Implementering

- Lav passende konstruktører.
- Lav passende get og set metoder.
- Lav passende toString metoder.
- Lav en klasse GameBoard der kan indeholde alle felterne i et array.
- Alt skal udvikles efter Objektorienterede principper, hvilket for felternes vedkommende vil sige at de skal laves med et fornuftigt arvehierarki.
- Tilføj en toString metode der udskriver alle felterne i arrayet.

- Lav det spil kunden har bedt om med de klasser I nu har.
- Benyt GUITen. Gui' skal importeres fra Maven: [Maven repository](#)

Dokumentation

- Forklar hvad arv er.
- Forklar hvad abstract betyder.
- Fortæl hvad det hedder hvis alle fieldklasserne har en landOnField metode der gør noget forskelligt.
- Dokumentation for test med screenshots.
- Dokumentation for overholdt GRASP.

Test

Lav mindst tre testcases med tilhørende fremgangsmåde/testprocedure og testrapporter.
 Lav mindst én Junit test til centrale metoder. Inkluder code coverage dokumentation.
 Lav mindst én brugertest. Husk at brugeren skal være en der ikke kan kode.

Versionsstyring

Lav et lokal Git-repository som en del af IntelliJ-projektet.
 Alternativt kan afleveres et link til et repository på nettet eks.

<https://github.com/cbudtz/TestRepo42.git>.

Rapporten skal indeholde en vejledning i hvordan man importerer Git-repository i IntelliJ.

Konfigurationsstyring

Udviklingsplatformen er alt det software i bruger under udviklingen af jeres projekt.
 Produktionsplatformen er alt det software der skal bruges til at køre jeres færdige program. I dette projekt er de ens. I skal dokumentere platformens dele med versionsnummer så den kan genskabes til senere brug. Jeres udviklingsplatform består af operativsystem, java, og IntelliJ samt biblioteket matadorgui.jar der hentes fra Maven.

I bedes definere hvordan I vil sikre jer, at I på et hvert tidspunkt vil kunne finde den sidste nye version af samtlige artefakter OG hvordan I vil sikre jer at dokumentationen altid er opdateret for jeres system

- Hvor er filerne?
- Hvordan finder vi den nyeste version?
- Hvordan sikrer vi at vi ved om versionen er opdateret?
- Hvordan finder vi versioner som passer sammen?

Endvidere skal i beskrive hvordan man importerer jeres projekt i IntelliJ fra git, og hvordan man kører jeres program.

Det er også et krav at I bruger Maven til at hente junit.