

# CDIO delopgave 1

## Indledning

I denne opgave i kurserne Indledende programmering og Udviklingsmetoder til IT-systemer har spilfirmaet IOOuterActive fået en opgave. I er udviklerne og er opdelt i grupper, som hver især skal analysere, designe, implementere og teste denne opgave. Til brug for dette får I kundens vision og projektlederens bemærkninger. I har mulighed for at stille spørgsmål til hjælpelærerne, som spiller rollen som projektleder, som har mulighed for at tale med kunden. Derfor er det vigtigt at I specificerer hvem, det er I stiller et spørgsmål til. I må acceptere at især kunden og til dels projektlederen nogle gange er lang tid om at svare.

## Kundens vision:

Vi vil gerne have et system, der kan bruges på maskinerne (Windows) i databarerne på DTU. Det skal være et spil mellem 2 personer. Spillet skal gå ud på at man slår med et raflebæger med to terninger og ser resultatet med det samme. Summen af terningernes værdier lægges til ens point. Vinderen er den, der opnår 40 point. Hvis der er ressourcer til det, er der følgende ekstraopgaver:

1. Spilleren mister alle sine point hvis spilleren slår to 1'ere.
2. Spilleren får en ekstra tur hvis spilleren slår to ens.
3. Spilleren kan vinde spillet ved at slå to 6'ere, hvis spilleren også i forrige kast slog to 6'ere uanset om det er på ekstrakast eller i forrige tur.
4. Spilleren skal slå to ens for at vinde spillet, efter at man har opnået 40 point.

Vi har fået lavet en [GUI](#) til et andet projekt, der gerne må bruges til at vise terningernes værdi på. (I må gerne udforske GUI'en og evt. benytte flere funktioner.)

Vi forventer at alle almindelige mennesker kan spille det uden en brugsanvisning.

Vi vil gerne se en test, der beviser at raflebægeret virker korrekt, hen over 1000 kast.

Det er op til jer om dokumentation og kode skal være på dansk eller engelsk, dog skal fagudtryk være naturlige.

## Projektlederens bemærkninger:

### Kravspecificering og use cases

Når man skal forstå hvad andre siger, kan det være en god ide at gentage med egne ord, hvad man forstod ved det, som de andre sagde. Hermed reduceres risikoen for misforståelser. På samme måde kan det i forbindelse med udvikling af programmer være en god idé at forsøge med egne ord at udtrykke de krav, som man modtager, for at sikre at man forstår kravene.

Husk: Krav er målbare.

Leg djævlens advokat og påpeg alle de områder på hvilken beskrivelsen er ufuldstændig. Med ufuldstændig mener vi f.eks. steder, hvor der er implicitte antagelser om viden hos jer, som er dem der skal honorere disse krav (NB! Listen kan blive meget lang)

Omskriv kravspecifikationen, således at et passende specifikationsniveau er til stede.

### Dokumentation

Der skal udarbejdes en lille rapport, der indeholder kapitler, som beskriver resultatet af jeres udviklingsarbejde. I vælger selv de artifacts (diagrammer, beskrivelser ol.), som er relevante i forhold til den konkrete opgave.

## Kode

I skal bruge IntelliJ til udvikling og vi vil gerne sikre ensartethed på tværs af både sprog og landegrænser. Derfor skal I bruge UTF-8 som tegnsæt. Man finder indstillingen på workspace niveau her:

File -> Settings -> Editor -> File Encodings:

Global Encoding: UTF-8

Project Encoding: UTF-8

Opret et projekt med navnet **o6\_del1** (vigtigt at **o6** udskiftes med jeres gruppenummer)

Det er altid dårlig stil at bruge default-pakken. Opret derfor følgende pakker `spil` og `test` hvor I placerer jeres klasser.

Husk at i `IOOuterActive`, ligesom i de fleste andre firmaer, er attributter `private`. Derfor skal der være `public` get- og set metoder hvis attributter skal kunne tilgås udefra. Vi tror også vi får brug for en `getSum():int` metode, som giver summen af øjnene på de to terninger og en `getEns():boolean` metode, som fortæller, hvorvidt de to terninger viser samme værdi.

Vi forudser, at vi skal bruge noget tilfældighedsgenerering, og der kan I enten bruge metoden `nextInt(int n)` på et objekt af klassen `java.util.Random` eller metoden `random()` på klassen `java.lang.Math`. I kan læse i Java API'et hvordan metoderne fungerer.

Vi har kigget lidt på GUI'en, og der er en metode `setDice(int, int)`, der placerer to terninger på en spilleplade med de angivne værdier.

Kunden vil gerne se resultatet af et kast med det samme. Det betyder oversat til vores sprog, at det ikke må tage mere end 333 millisekunder. Vi har testet GUI'en, og det tager ca. 600 ms første gang man kalder `setDice`, men det ignorerer vi, da det de efterfølgende gange kun tager ca. 2 ms.

## Versionsstyring

Kunden har bedt om mulighed for at kunne inspicere koden og udviklingen i koden. Derfor skal I i jeres aflevering inkludere et link til jeres github repo. De vil gerne kunne tjekke den seneste fungerende kode ud fra master-branchen, lav derfor ændringer som ikke er kørt og testet i en development-branch. Det skal kunne ses af historien på github at alle gruppens medlemmer har committet et eller andet.

## Test

Kunden har bedt om en test. Vi vil foreslå, at I optæller antallet af forekomster for hver mulig værdi af summen (2-12). Optæl desuden antallet af kast, hvor terningerne er ens. Kontrollér at resultaterne stemmer overens med de teoretiske sandsynligheder.