

Minimumskrav til dokumentation

Et separat dokument – gerne som pdf-dokument, der indeholder:

- Navn på udvikler, dato og et navn på projektet
- UML diagram (Konceptuelt klasse diagram & relationer)
- En kort teknisk beskrivelse af projektet med begrundelser for de anvendte teknikker.
 - Især skal klasse biblioteket beskrives, så andre kan anvende det.
- Logbog over udviklingen af projektet
 - Ændringer
 - Problemer/Udfordringer
 - Husk: Det er bedre med dokumentering af problemer end ingenting.

Dokumentationen må gerne være mere uddybende end beskrevet – men det skal ikke være en roman.

Hvis I anvender flow charts til at analysere opgaven, må de meget gerne inkluderes i dokumentationen.

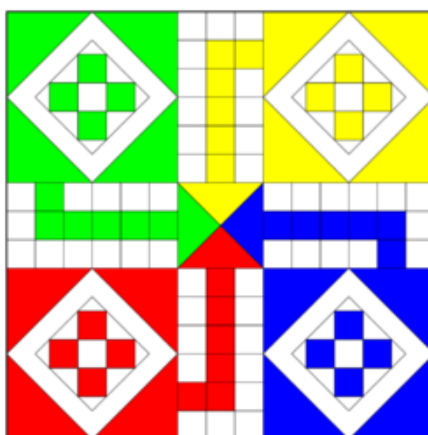
Læs HELE opgaven godt igennem og stil spørgsmål så tidligt som muligt (hvis der er noget, I er i tvivl om).

1. Ludospil

Hvis du har lavet et konsol-baseret Ludospil, skal du arbejde videre med opgaven. Hvis spillet ikke er helt færdigt, så gør det ikke noget, du kan arbejde videre med det, du har.



- Der SKAL laves et UML klasse diagram.
- Grundlæggende skal man kunne spille ludo med 2 – 4 spiller.
- Det er vigtigt, at spillet er fuldt objektorienteret og overholder præmisserne for OOP.
- Du skal lave en mini-**GUI** til spillet, men pointen er, at du arbejder videre med det, du har - dog med visse ombygninger.
- Dit OOP konsolspil skal laves om til et **Class Library** (CL), som du kan anvende i et nyt **WPF** (**Windows Presentation Foundation**) eller i et **WinForms** projekt med reference til dit **CL**. WPF eller WinForms projektet udgør din GUI, og dit Class Library udgør din Business Logik.
- Det er vigtigt, at du tænker over dit OOP design, herunder SKAL du gøre brug af Interface og/eller Abstrakt klasser.
- Spillereglerne for Ludo, kan læses her:
<http://www.papskubber.dk/braetspil/ludo>
- **Udvidelser:**
 1. Lav en variant af dit ludospil, der anvender reglerne for Globeludo. Terningen har globus og stjerne, samt visse felter på pladen.
 2. Lav AI (Artificiel Intelligence) robotter, der spiller de tomme pladser om bordet. Således der altid er 4 spillere med.
 - a. Lav evt. Sværhedsgrad/snyd på "robotterne"



Læs eventuelt dokumentet "Hjælp til nedbrydning af ludo del 2"

2. Udbygning af den lille shop

Baseret på øvelsen med den lille shop skal du udbygge opgaven.

- Det er vigtigt, at de forskellige elementer i shoppen er fuldt objektorienteret og overholder præmisserne for OOP.
- Du skal lave en **GUI** til shoppen, men pointen er, at du arbejder videre med det du har, dog med visse ombygninger.
- Objekterne i din OOP konsolshop skal laves om til et **Class Library** (CL), som du kan anvende i et nyt **WPF** (**W**indows **P**resentation **F**oundation) eller i et **WinForms** projekt med reference til dit **CL**.
WPF eller WinForms projektet udgør din GUI, og dit Class Library udgør din Business Logic.
- Det er vigtigt, at du tænker over dit OOP design, herunder SKAL du gøre brug af Interface og/eller Abstrakt klasser.
- Du skal også anvende en eller flere af de følgende metoder: Overskrivning, abstrakte metoder og/eller polymorfi.
 - Tænk f.eks. forskellige typer af varer, der kan benytte en fælles klasse "vare" og dog adskille sig fra hinanden ved forskellige enheder, som pris er per liter, kilo, stk. eller noget tilsvarende.

Hvis der er tid nok:

- Implementering af et asynkront forløb.
- Lambda udtryk og/eller eventhåndtering i form af delegates.

Andre udvidelser:

- OOP opgaven kan også ændres til at omfatte en **database opgave** – se de separate krav til databasen.

Husk

- At dokumentere opgaven undervejs (log bog). Skriv kort, men dagligt.
- Versionsstyring er ikke et krav men en god ide også at udføre dagligt.
- OOP kan være på rutineret, avanceret eller ekspert niveau

3. Eget projekt

- Skal godkendes på forhånd –
- Skal indeholde tilsvarende elementer, som i de to forgående opgaver.

Vurdering af opgave:

	<i>Niveauet, som opgaven løses på</i>	<i>Personlige kompetence, der knytter sig til niveauet.</i>
Rutineret	<p>Rutinemæssig eller kendt situation.</p> <ul style="list-style-type: none"> • Alene og i samarbejde. • Planlægge og gennemføre en opgave eller aktivitet. • Løse et problem. • Selvstændigt sætte sig ind i mere komplicerede problemstillinger. 	<ul style="list-style-type: none"> • Viser fleksibilitet og omstillingsevne.
Avanceret	<p>Ikke-rutinesituationer.</p> <ul style="list-style-type: none"> • Alene eller i samarbejde. • Vurdere et problem. • Planlægge, løse og gennemføre en opgave eller aktivitet. • Løse et problem. 	<ul style="list-style-type: none"> • Tager selvstændigt ansvar. • Viser initiativ. • Lægger vægt på kvalitetssans og kreativitet.
Ekspert	<p>Bruge allerede opnåede kompetencer.</p> <ul style="list-style-type: none"> • Løse komplekse arbejdsopgaver. • Argumentere for valgte løsninger og opståede problemer. • Bruge opnåede kompetencer i en ny kontekst. • Arbejde med overblik og deltagelse i arbejdspladsens innovative processer. 	<ul style="list-style-type: none"> • Planlægge, tilrettelægge, udføre og evaluere arbejdsprocesser. • Vurdere og begrunde behovet for at forbedre arbejdsprocesser. • Kommunikere om sin faglighed i alle relevante sammenhæng.