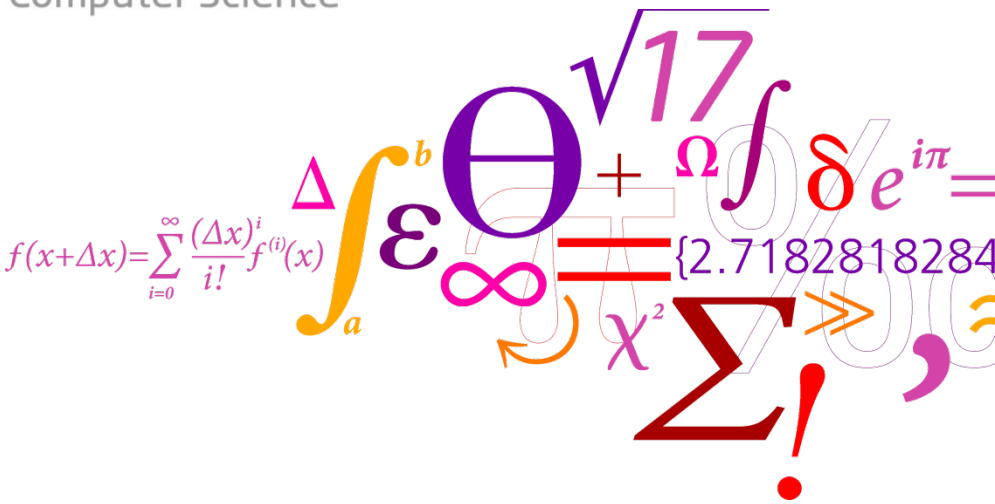
Projekt i software-udvikling (02362)
Forår 2022

Ekkart Kindler

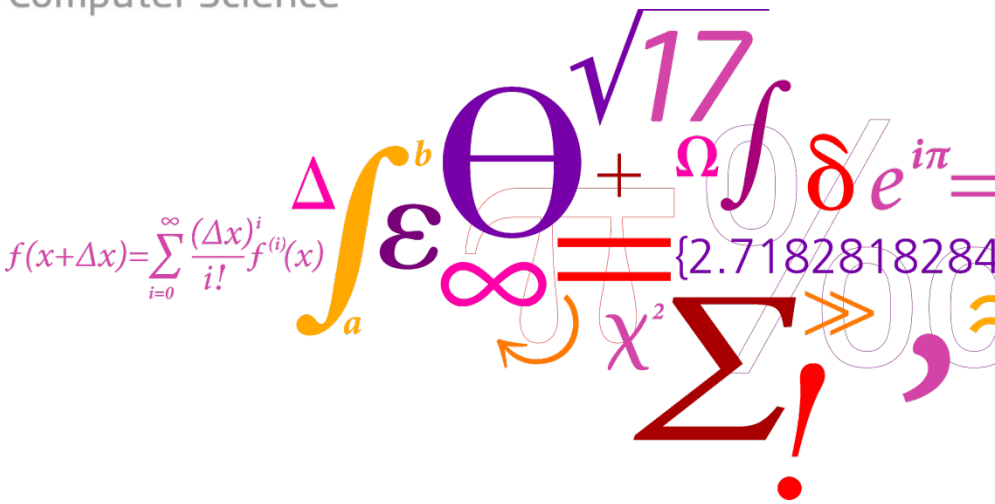
DTU Compute
Department of Applied Mathematics and Computer Science

Bemærk: Kun slides 17 og 25-29 er nye. Alle andre slides blev præsenter allerede!



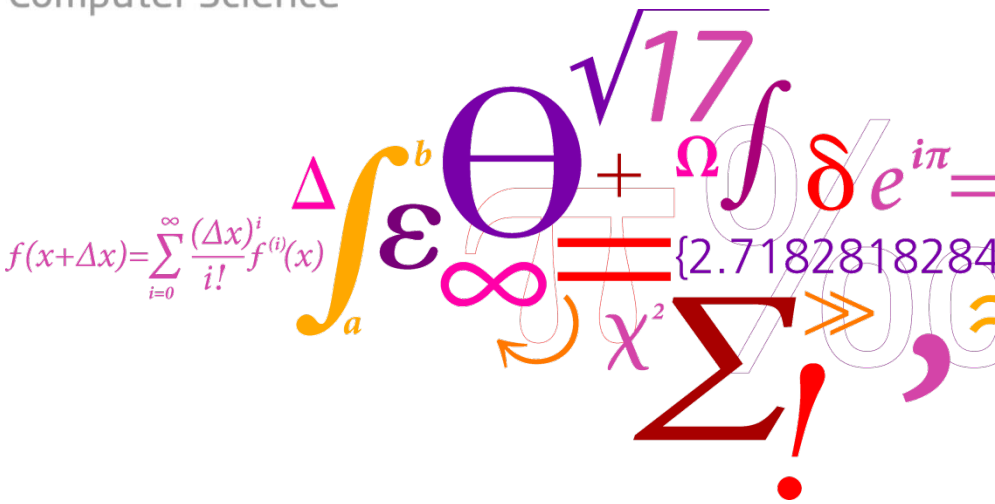
X. Projekt, aflevering og eksamen

DTU Compute
Department of Applied Mathematics and Computer Science



X.1 Projekt

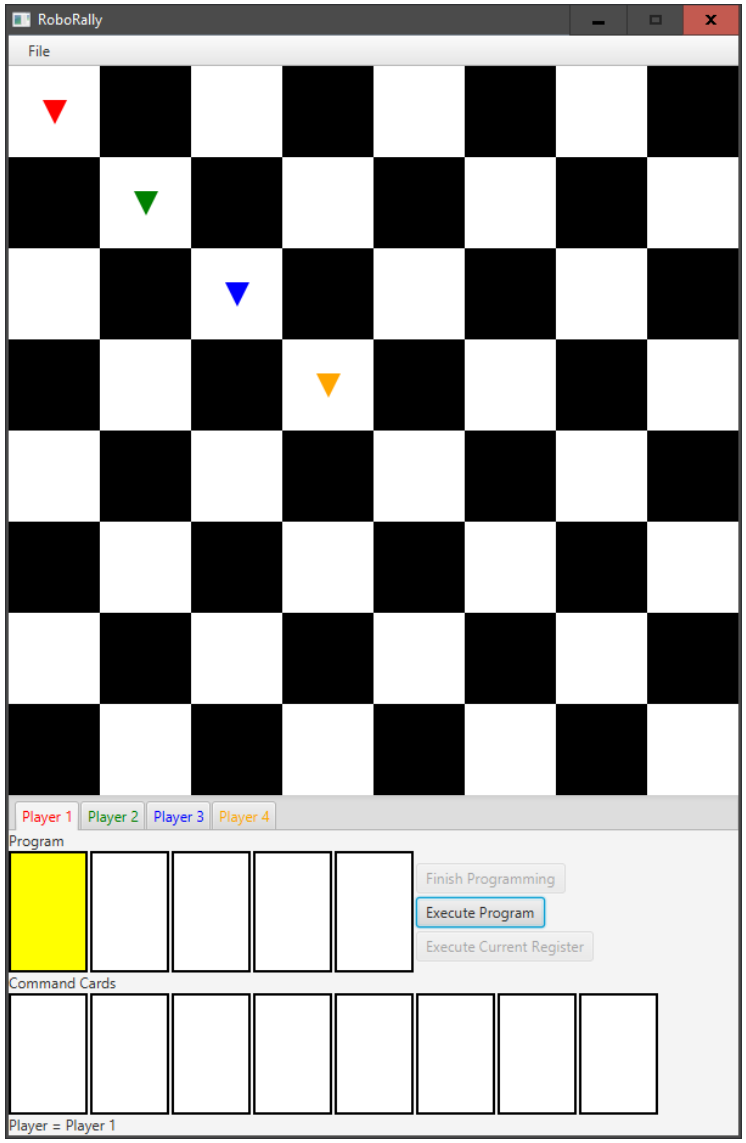
DTU Compute
Department of Applied Mathematics and Computer Science



RoboRally

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Er et spil, hvor hver spiller programmerer sin robot med kommandokort. Hvorefter, robotterne helt automatisk bevæger sig efter programmet i den fabrik (spilleplade) de bevæger sig i.
- Kommandokort siger noget om hvordan robotten flytter sig og drejer rundt få felterne i fabrikken.
- Felterne i fabrikken og andre robotter har også indflydelse på ens robot (flytter den, skyder på den) og der kan også være væg, så at robotten ikke kan flytte sig som programmeret.
- Formålet er at ens robot når alle ”checkpoints” i den rigtige rækkefølge som den første.


RoboRally
File
Player 1
Player 2
Player 3
Player 4
Program
Finish Programming
Execute Program
Execute Current Register
Command Cards
Player = Player 1

?

Opgave

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

I skal implementere **RoboRally** så at

- en **betydelig mængde af spillets regler** er realiseret,
- man skal kunne **gemme** (flere) **spil** i en database, så at man kan fortsætte disse spil senere,
- alle **spillets informationer vises grafisk** på en hensigtsmæssige måde,
- evt. kan gengive et spils forløb og
- evt. kan vælge (og selv definere) nogle spilleplader
- …

RoboRally

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Spillets regler finder man på:

- **`https://avalonhill.wizards.com/games/robo-rally`**
- **`http://media.wizards.com/2017/rules/roborally_rules.pdf`**

At implementere RoboRally med alle dens reger og detaljer ville være meget ambitiøst! Derfor bestå kunsten i at udvælge features og reglerne fornuftig. Det taler vi om under kursets forløb.

Realisering

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Fokus skal ikke være på den mest brugervenlige og grafisk tiltalende brugergrænseflade (GUI). Men GUIen skal vise spillets tilstand og køre stabilt.
- Man kan udvide ”brættet” fra den første (V1) eller anden(V2) opgave trinvis.
- Til gengæld skal softwaren have en klar struktur:
  - adskille model (spillets tilstand), view (GUI) og kontrol
  - god design af API (model og kontrol)
  - klart og enkelt interface mellem database og selve software
  - være nemt at udvide (især når I ikke når at implementere alle regler)

Støttede regler

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

I skal finde og argumentere for jeres prioritering af implementerede regler (eller regelområder):

- forskellige programmeringskort (som inkluderer interaktive kort → se V3)
- forskellige fabrikfelter som interagerer med robotten
- evtl. mulighed for at opgradere og lade robotten
- …

Jeres prioritering skal garantere at det giver mening at spille spillet og spillet kan slutte med en vinder

Out of scope

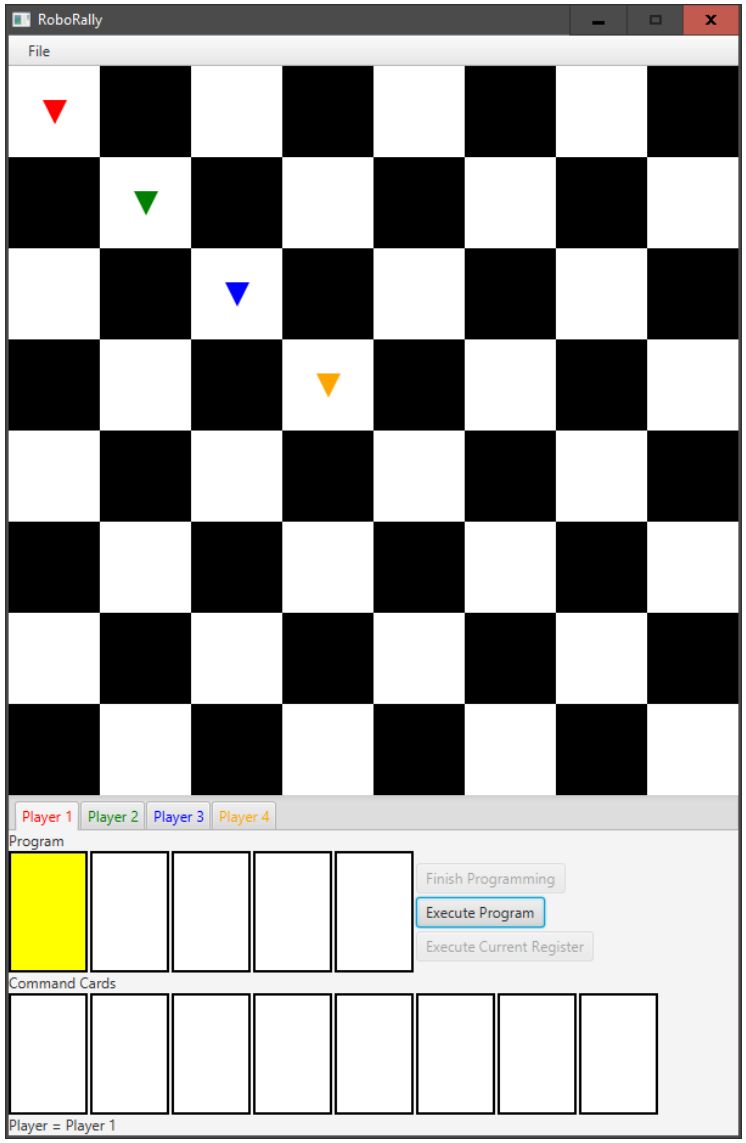
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Det ville give mest mening, at spille RoboRally online!

Men det skal I ikke implementere i 02362!

- Sofistikerede GUI (drag and drop, etc. Ud over det I fik); men særlige felter, skal markeres på spillepladen

Vi diskuterer det nærmere sammen med domæneanalysen og aflevering af projektdefinition.
Og vi taler om hvordan man kan realisere det i kursets forløb.



?

Hvilke features skal der være med ud over det som I fik udleveret i starten?

Formelle krav

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- De næste slides præsenterer minimale krav vedr. Features af jeres software
- Men I skal også huske tekniske og formelle krav til aflevering (der kommer en tjekliste på kursets websider):
  - softwarens arkitektur (MVC, DAL)
  - author tags
  - Javadoc kommentarer
  - test til vigtig spillelogik
  - …

Vigtigt til eksamen: **alle gruppens medlemmer** har sat sig ind i softwaren og kan forklare dens struktur og nogle detaljer (især til MVC, DAL, …)

Minimumskrav vedr. features

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Gemme spil i databasen og lade spil igen fra databasen
  - spillerens håndkort og program skal også gemmes
  - det skal også gemmes hvilken spilleplade spillet var på
- Spilleplader er defineret gennem JSON-filer (**mindst to**)
- Interaktive spillekort
- Vægge på spillepladen
- **Mindst to** feltaktioner
- **Checkpoints**

Selvfølgelig skal også featurens logik implementeres og features kan skulle ses på spillepladen.

Features til lidt højere niveau

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

For at komme lidt højere op skal der være flere features (fx.):

- eksplicitte startfelter til robotter
- flere feltaktioner (af forskellig art)
- damage-kort
- feltaktioner bliver udført på det rigtige tidspunkt i spillets forløb (se reglerne)
- lidt finere grafik
- spil kan afsluttes fornuftigt

Alle informationer om spillets aktuelle situation skal kunne ses på spillepladen!

Men prioriter ikke mængden af features over softwarens kvalitet (især stabilitet)

Avancerede features

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

For at komme endnu længere (fx.):

- antennen (som afgør rækkefølgen robotternes registre og feltaktionerne bliver udført)
- energy cubes, pits, reboot
- upgrade cards

Bemærk at nogle upgrade cards har meget avancerede logik. I er ikke nødt til at have alt det mest avancerede med.

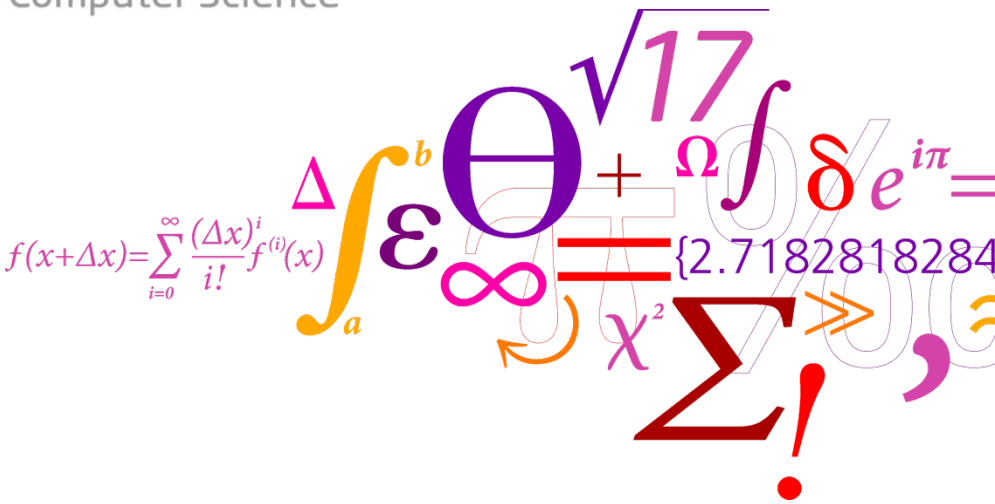
- Hver spiller har deres eget kortdæk
- pænere grafik
-

Også her: prioriter ikke mængden af features over softwarens arkitektur og kvalitet (især stabilitet)

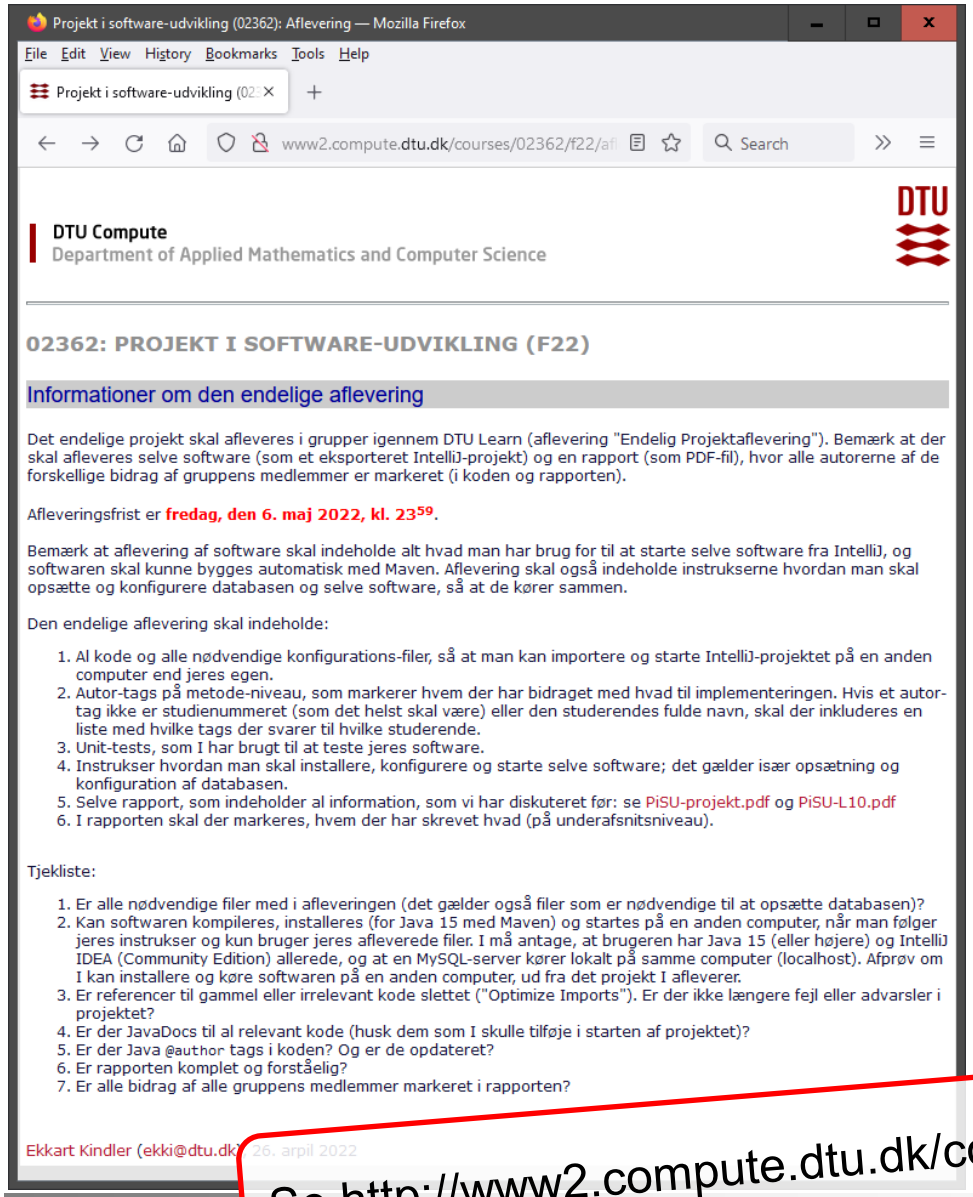
- … eller andre ”coole” features

X.1 Aflevering

DTU Compute
Department of Applied Mathematics and Computer Science



Info og Tjeklist



Projekt i software-udvikling (02362): Aflevering — Mozilla Firefox

File Edit View History Bookmarks Tools Help

Projekt i software-udvikling (02

www2.compute.dtu.dk/courses/02362/f22/afl

Search

DTU Compute
Department of Applied Mathematics and Computer Science

02362: PROJEKT I SOFTWARE-UDVIKLING (F22)

Informationer om den endelige aflevering

Det endelige projekt skal afleveres i grupper igennem DTU Learn (aflevering "Endelig Projektaflevering"). Bemærk at der skal afleveres selve software (som et eksporteret IntelliJ-projekt) og en rapport (som PDF-fil), hvor alle autorerne af de forskellige bidrag af gruppens medlemmer er markeret (i koden og rapporten).

Afleveringsfrist er **fredag, den 6. maj 2022, kl. 23^{59}**.

Bemærk at aflevering af software skal indeholde alt hvad man har brug for til at starte selve software fra IntelliJ, og softwaren skal kunne bygges automatisk med Maven. Aflevering skal også indeholde instrukserne hvordan man skal opsætte og konfigurere databasen og selve software, så at de kører sammen.

Den endelige aflevering skal indeholde:

1. Al kode og alle nødvendige konfigurations-filer, så at man kan importere og starte IntelliJ-projektet på en anden computer end jeres egen.
2. Autor-tags på metode-niveau, som markerer hvem der har bidraget med hvad til implementeringen. Hvis et autor-tag ikke er studienummeret (som det helst skal være) eller den studerendes fulde navn, skal der inkluderes en liste med hvilke tags der svarer til hvilke studerende.
3. Unit-tests, som I har brugt til at teste jeres software.
4. Instrukser hvordan man skal installere, konfigurere og starte selve software; det gælder især opsætning og konfiguration af databasen.
5. Selve rapport, som indeholder al information, som vi har diskuteret før: se PiSU-projekt.pdf og PiSU-L10.pdf
6. I rapporten skal der markeres, hvem der har skrevet hvad (på underafsnitsniveau).

Tjekliste:

1. Er alle nødvendige filer med i afleveringen (det gælder også filer som er nødvendige til at opsætte databasen)?
2. Kan softwaren kompileres, installeres (for Java 15 med Maven) og startes på en anden computer, når man følger jeres instrukser og kun bruger jeres afleverede filer. I må antage, at brugeren har Java 15 (eller højere) og IntelliJ IDEA (Community Edition) allerede, og at en MySQL-server kører lokalt på samme computer (localhost). Afprøv om I kan installere og køre softwaren på en anden computer, ud fra det projekt I afleverer.
3. Er referencer til gammel eller irrelevant kode slettet ("Optimize Imports"). Er der ikke længere fejl eller advarsler i projektet?
4. Er der JavaDocs til al relevant kode (husk dem som I skulle tilføje i starten af projektet)?
5. Er der Java `@author` tags i koden? Og er de opdateret?
6. Er rapporten komplet og forståelig?
7. Er alle bidrag af alle gruppens medlemmer markeret i rapporten?

Ekkart Kindler (ekki@dtu.dk), 26. arpil 2022

Se http://www2.compute.dtu.dk/courses/02362/f22/aflevering.shtml

Rapport (indhold)

Se PiSU-projekt.pdf og **også eksemplet** (materiale på DTU Learn uge 12)!

- Introduktion
  - Kort overblik over projektet og hovedpunkter, som opgaven indebærer
  - Overblik over selve rapporten og hvordan den skal læses
- Problembeskrivelse
  - Kontekst og baggrund
  - Overblik over hovedfunktioner (i grupper) med prioritering og argumentation for denne (kan fx bruge MoSCoW-kategorier: Must, Should, Could, Won't)
  - Husk også at nævne, hvilke data, der skal gemmes permanent (persistence)
  - Hvad kan I bygge videre på (fx GUI og kode i fik fra Ekkart v1.1.1 og v1.4.1)

Rapport (indhold)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Kravspecifikation (Analyse)
  - Beskrivelse af krav fra brugerens perspektiv
  - Beskriv funktioner og use-cases (som brugeren opfatter dem)
  - Beskriv de relevante informationer som domænemodel:
    - Klassediagrammer
    - Aktivitetsdiagrammer
    - evt. tilstandsdiagrammer
  - Beskriv ikke-funktionelle krav: fx brugbarhed (usability), vedligeholdbarhed (maintainability), …
  - Evtl. UC-diagram med beskrivelse af de vigtigste use-cases

Husk: Alle diagrammer skal forklares i teksten!

Rapport (indhold)

- Database- og softwaredesign

  Overblik over de vigtigste komponenter, som inkluderer, også tilkobling a databasen: arkitektur

  - Software design: hovedkomponenter af software og deres sammenspil

    → Især MVC, DAL, …

    - Klassediagrammer (indeholder især de vigtigste model-, kontroller- og view-klasser)
    - Sekvensdiagrammer som viser samspil mellem vigtige klasser/komponenter
  - Database design → se DB kursus

Rapport (indhold)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Implementering
  - Hvordan er designet bliver implementeret (men kun nogle interessante udtræk, ikke hele koden)
  - Dette omfatter software og database
  - Herunder kan I også diskutere evt. mangler af jeres implementering
- Udviklingsproces og test
  - Udviklingsproces og brugte værktøjer
  - Diskussion af JUnit-test (automatisk)
  - Diskussion af acceptance-test (manuelle test, fx use-cases)

Rapport (indhold)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Det skal være muligt at bruge softwaren uden anden information!

- Håndbog
  - Hvordan installerer man selve software
  - Hvordan starter man softwaren
  - Hvordan bruger man software (den eksisterende GUI er I ikke nødt til at diskutere med alle detaljer); evt. med nogle screenshots
- Konklusion
  - Opsamling af hele resultatet
  - Nogle afsluttende bemærkninger

Rapport (indhold)

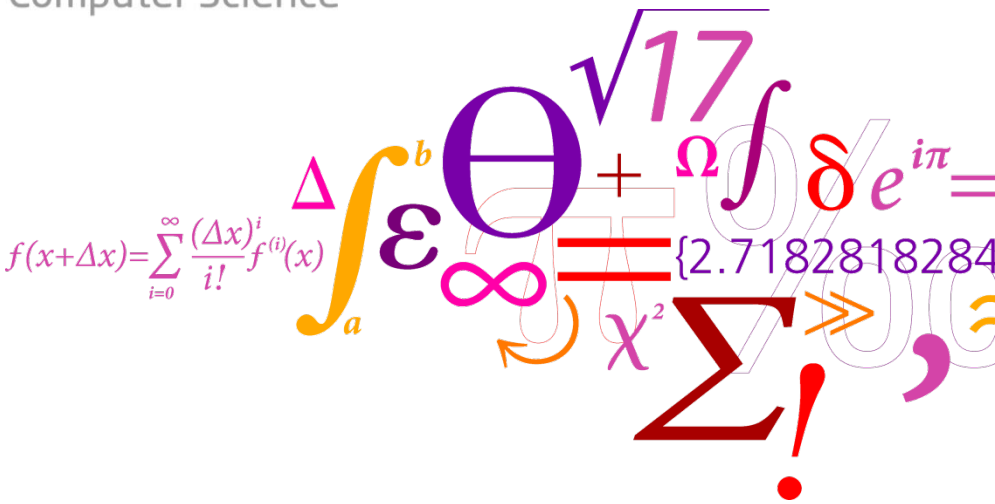
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Referencer
  - Litteratur og
  - Websider I har bugt
- Appendiks (evt.)
  - Nogle relevante diagrammer som ikke kunne være med i hoveddelen
  - Komplet ordliste / glossar

Hus at sige: Hvem der har skrevet hvad!

X.3 Eksamen

DTU Compute
Department of Applied Mathematics and Computer Science



Gruppepræsentation

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

15 minutter

Den røde tråd: Why, What, How, Demo

- Why: Indledning & Motivation

og måske "what" på meget højt abstraktionsniveau.

- What: Hvad har i udviklet fra brugerens synsvinkel (Analyse)
  - Hvilke hoved-usecases og -funktioner omfatter jeres software
  - GUI (hvis der er noget specielt hos jer)
  - Tekniske perspektiv af det: domænemodel (klassediagrammer, tilstandsdiagrammer, aktivitetsdiagrammer)

Why, what, how: gerne med nogle Powerpoint-slides

Gruppepræsentation

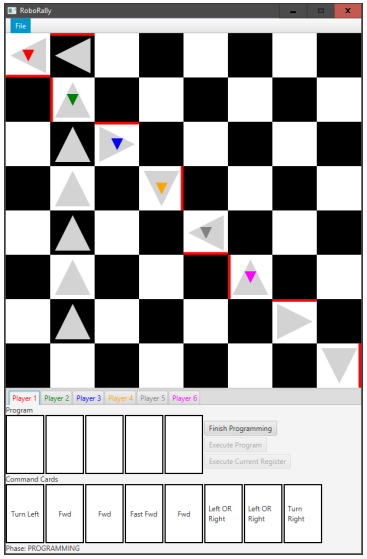
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- How: Hvordan er softwaren realiseret
  - Arkitektur & Design
  - Nogle få implementeringsdetaljer

Især: MVC og DAL

Et kort overblik af jeres IntelliJ-projekt og måske database (MySQL workbench, etc.)

- Demo: Vis hvad jeres software kan



Tip: Gem et spil i en spænende situation, så at I kan vise situationer, som kun sker senere hen i spillet (checkpoint, vinde, skubbe, pit, …)

Hvert foredrag skal have en indledning og afslutning (afrunding)

“Deltagere”

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

I skal tale til

- en med ikke så meget teknisk forstand (CEO), som skal forstå hvad softwaren kan gøre
- og end med mere teknisk forstand (CTO), som I skal overbevise at det I gøre giver mening og I har styr på jeres design

så at begge to får noget ud af det

Individuelle eksaminer

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

10-15 minutter

I skulle kunne svare på, hvordan de forskellige emner fra forelæsningen kan findes I jeres analyse, design og implementering.

Det gælder især:

- Domænemodel og diagrammer
- MVC
- Detaljer af jeres
  - controllere: app- og spillelogik
  - DAO / filer
- Exceptions
- Generics

Hver studerende skal bringe en computer med softwareprojektet installeret i IntelliJ, så at I kan pege på tingene i implementeringen i jeres IDE.

I skal også have rapporten på jeres computere, så at vi fx. kan spørge om jeres diagrammer i rapporten

Tidsplan

Se http://www2.compute.dtu.dk/courses/02362/f22/eksamen.shtml

Onsdag, 11. maj

8^{00}: Group 4

10^{30}: Group 1

13^{15}: Group 5

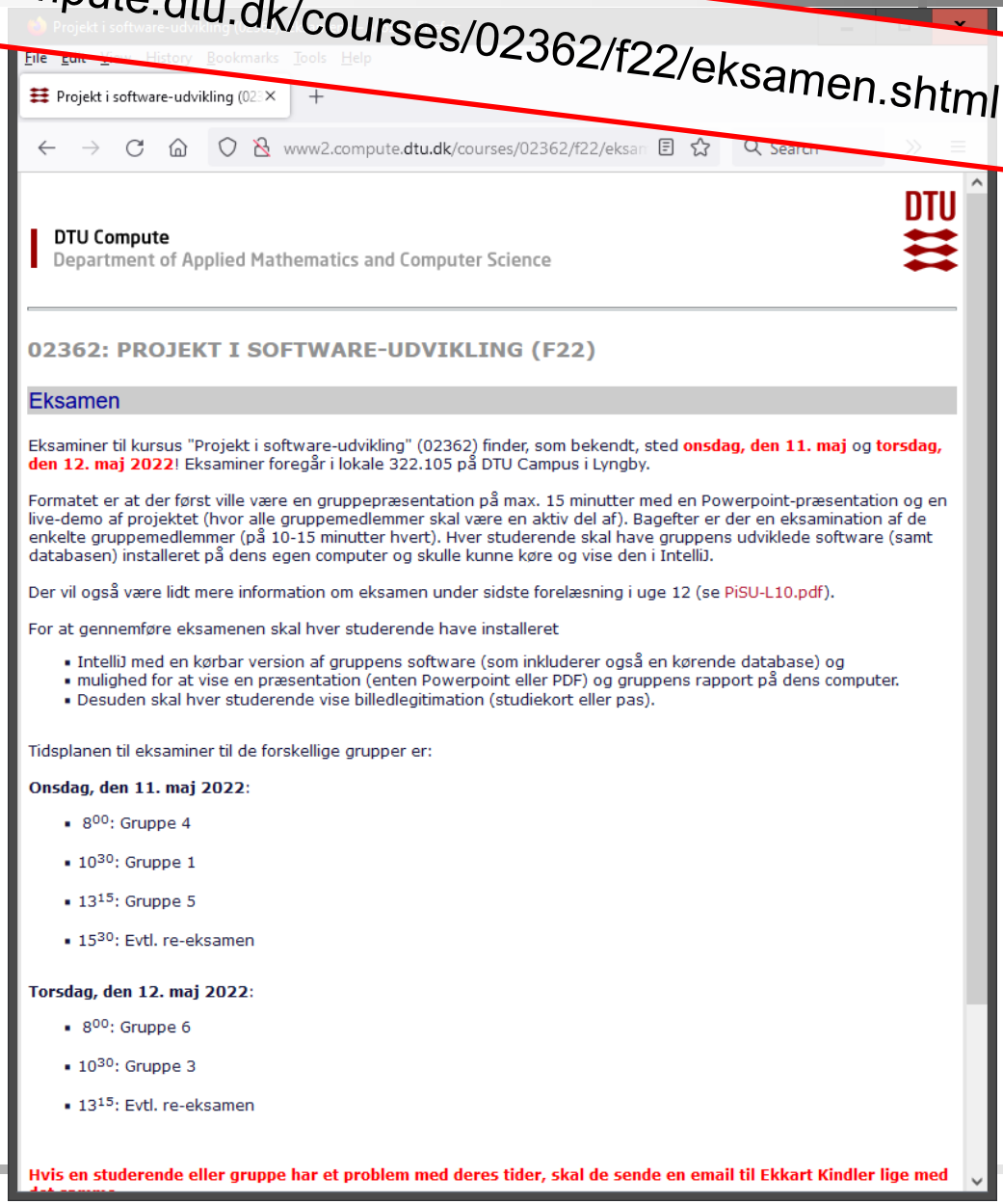
...

Torsdag, 12. maj

8^{00}: Group 6

10^{30}: Group 3

...



Projekt i software-udvikling (02

www2.compute.dtu.dk/courses/02362/f22/eksa

DTU Compute
Department of Applied Mathematics and Computer Science

02362: PROJEKT I SOFTWARE-UDVIKLING (F22)

Eksamen

Eksaminer til kursus "Projekt i software-udvikling" (02362) finder, som bekendt, sted **onsdag, den 11. maj** og **torsdag, den 12. maj 2022**! Eksaminer foregår i lokale 322.105 på DTU Campus i Lyngby.

Formatet er at der først ville være en gruppepræsentation på max. 15 minutter med en Powerpoint-præsentation og en live-demo af projektet (hvor alle gruppemedlemmer skal være en aktiv del af). Bagefter er der en eksamination af de enkelte gruppemedlemmer (på 10-15 minutter hvert). Hver studerende skal have gruppens udviklede software (samt databasen) installeret på dens egen computer og skulle kunne køre og vise den i IntelliJ.

Der vil også være lidt mere information om eksamen under sidste forelæsning i uge 12 (se PiSU-L10.pdf).

For at gennemføre eksamenen skal hver studerende have installeret

- IntelliJ med en kørbar version af gruppens software (som inkluderer også en kørende database) og
- mulighed for at vise en præsentation (enten Powerpoint eller PDF) og gruppens rapport på dens computer.
- Desuden skal hver studerende vise billedlegitimation (studiekort eller pas).

Tidsplanen til eksaminer til de forskellige grupper er:

Onsdag, den 11. maj 2022:

- 8^{00}: Gruppe 4
- 10^{30}: Gruppe 1
- 13^{15}: Gruppe 5
- 15^{30}: Evtl. re-eksamen

Torsdag, den 12. maj 2022:

- 8^{00}: Gruppe 6
- 10^{30}: Gruppe 3
- 13^{15}: Evtl. re-eksamen

Hvis en studerende eller gruppe har et problem med deres tider, skal de sende en email til Ekkart Kindler lige med