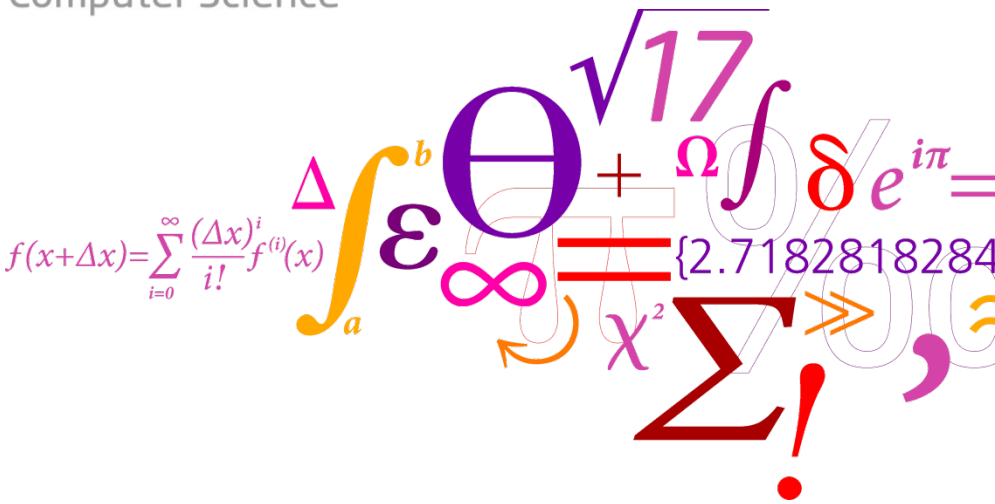
Videregående Programmering (02324)
Projekt i software-udvikling (02362)
Forår 2021

Ekkart Kindler

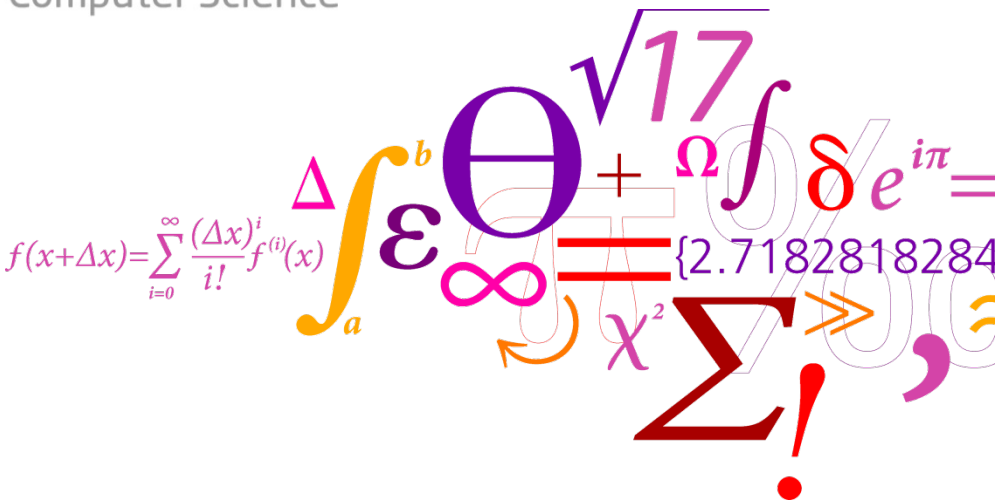
DTU Compute
Department of Applied Mathematics and Computer Science



I. Introduktion

Meget kort rekapitulation af uge 1

DTU Compute
Department of Applied Mathematics and Computer Science



Opfriskning (eksempel)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

```
class Pos {
   int x;
   int y;
}
class PosName {
   String name;
   Pos pos;
   public PosName(String name, Pos pos) {
      this.name = name;
      this.pos = pos;
} }
```

Denne kode er ikke pæn!

Kun et teknisk eksempel, som opfrisking af nogle begreber og tænkemåder!

Opfriskning (eksempel)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

```
class Test {
public static void main(String[] args){
    Pos pos1 = new Pos();
    pos1.x = 1;
    pos1.y = 2;

    Pos pos2 = pos1;
    pos2.x = 3;
    pos2.y = 4;
```

Hvilke værdier har **`pos1.x`**, **`pos2.x`**, **`pos1.y`** og **`pos2.y`** når programmet når her? Og hvorfor?

→ Se diskussion på tavlen!

Opfriskning (eksempel)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

```
PosName posName1 = new PosName("position 1", pos1);
PosName posName2 = new PosName("position 2", pos2);


Pos[] posArray = new Pos[10];
for (int i = 0; i < posArray.length; i++) {
  posArray[i].x = i;
  posArray[i].y = i;
}

posArray[1] = pos1;
posArray[2] = pos2;
```

Det ville udløse en `NullPointerException`

Opfriskning (eksempel)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

```
PosName posName1 = new PosName("position 1", pos1);
PosName posName2 = new PosName("position 2", pos2);


Pos[] posArray = new Pos[10];
for (int i = 0; i < posArray.length; i++) {
  posArray[i] = new Pos();
  posArray[i].x = i;
  posArray[i].y = i;
}

posArray[1] = pos1;
posArray[2] = pos2;
```

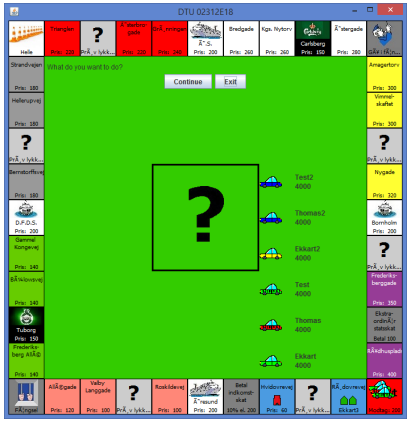
Hvordan ville værdierne se ud her?

→ Se diskussion på tavlen!

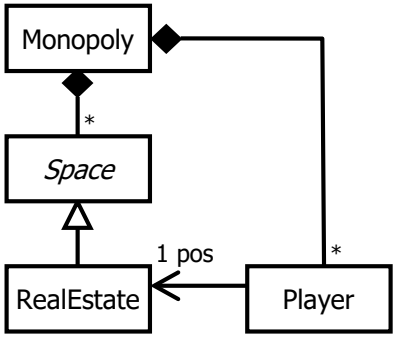
Model View Controller (MVC)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

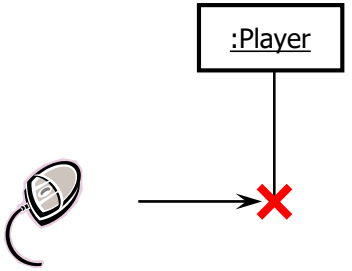
View



Model

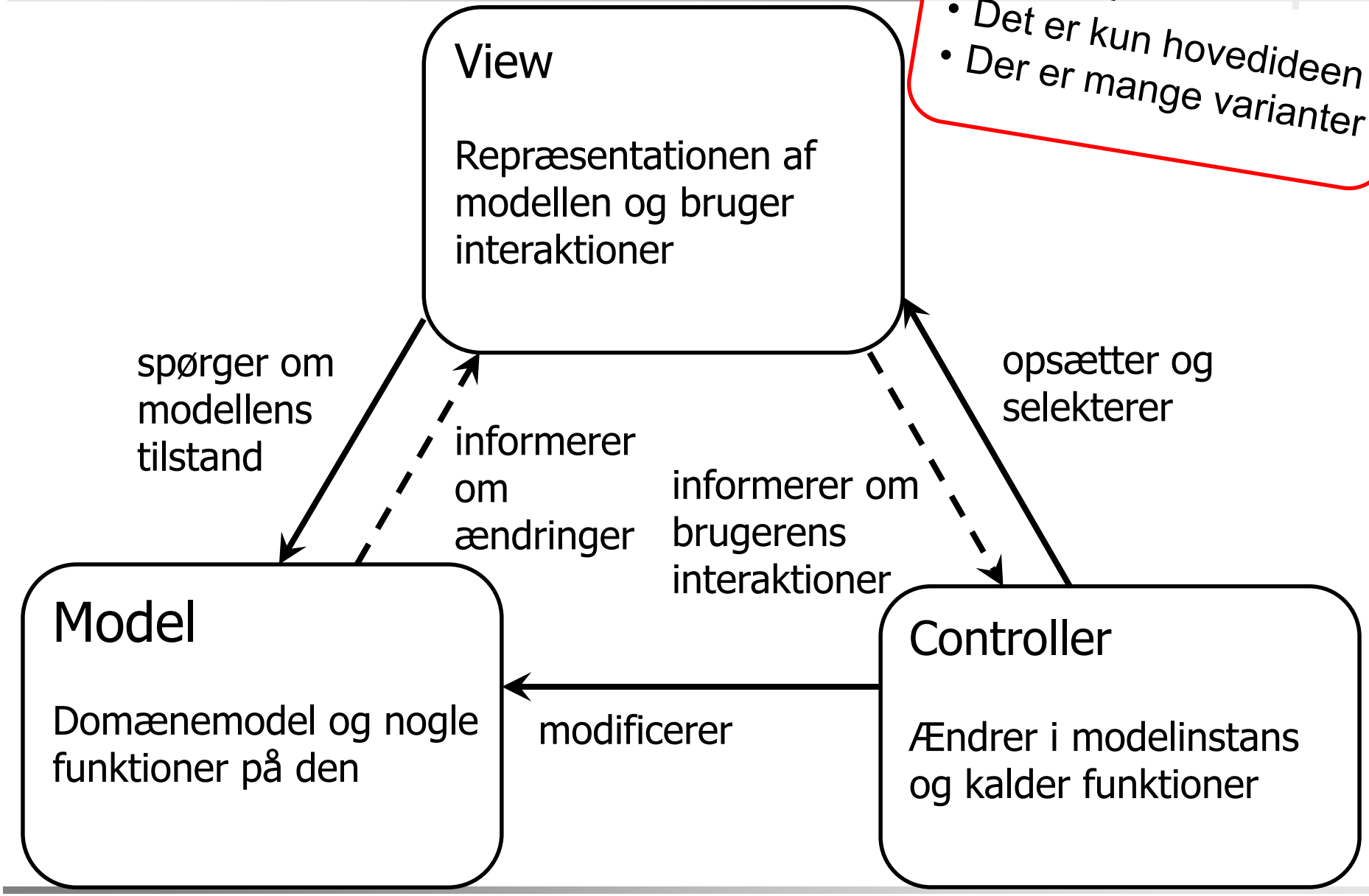


Controller



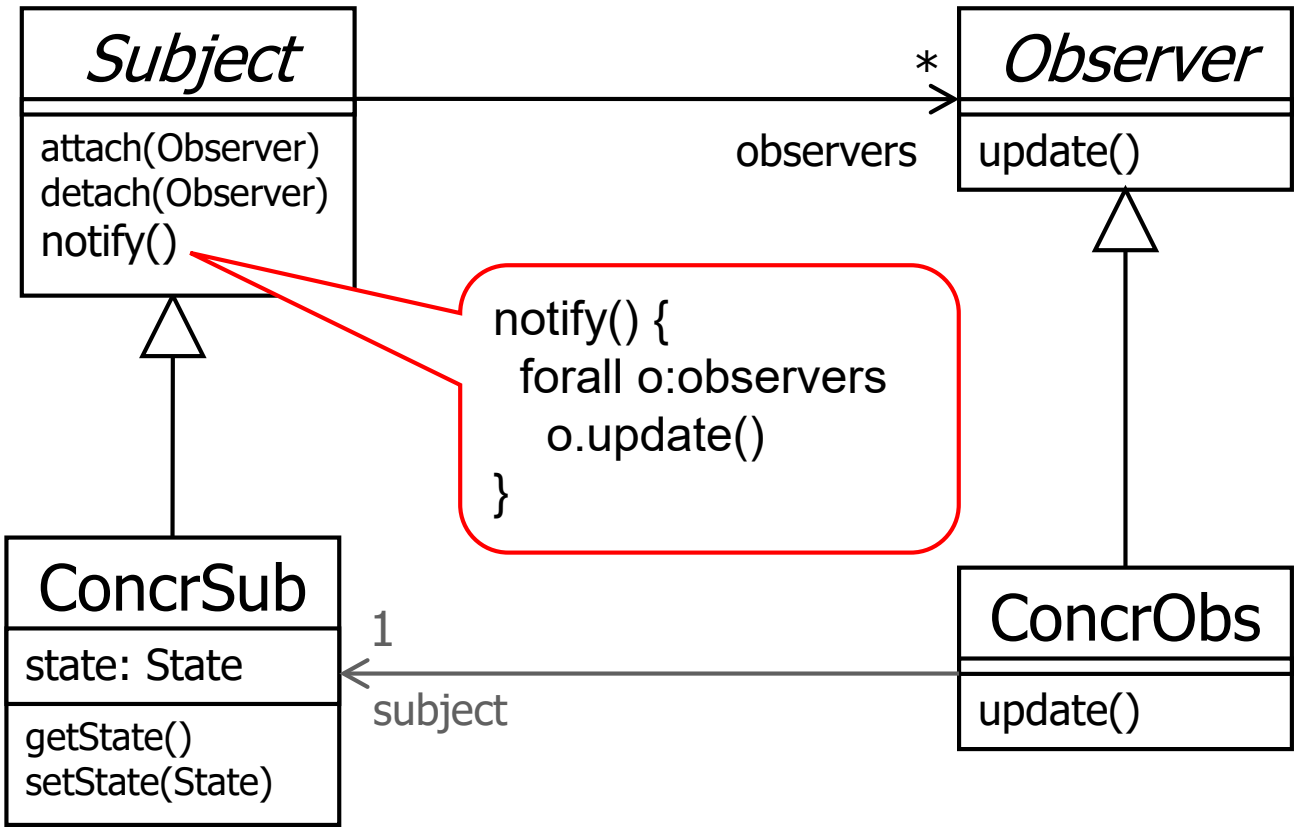
MVC

Bemærk:
- Det er kun hovedideen
- Der er mange varianter



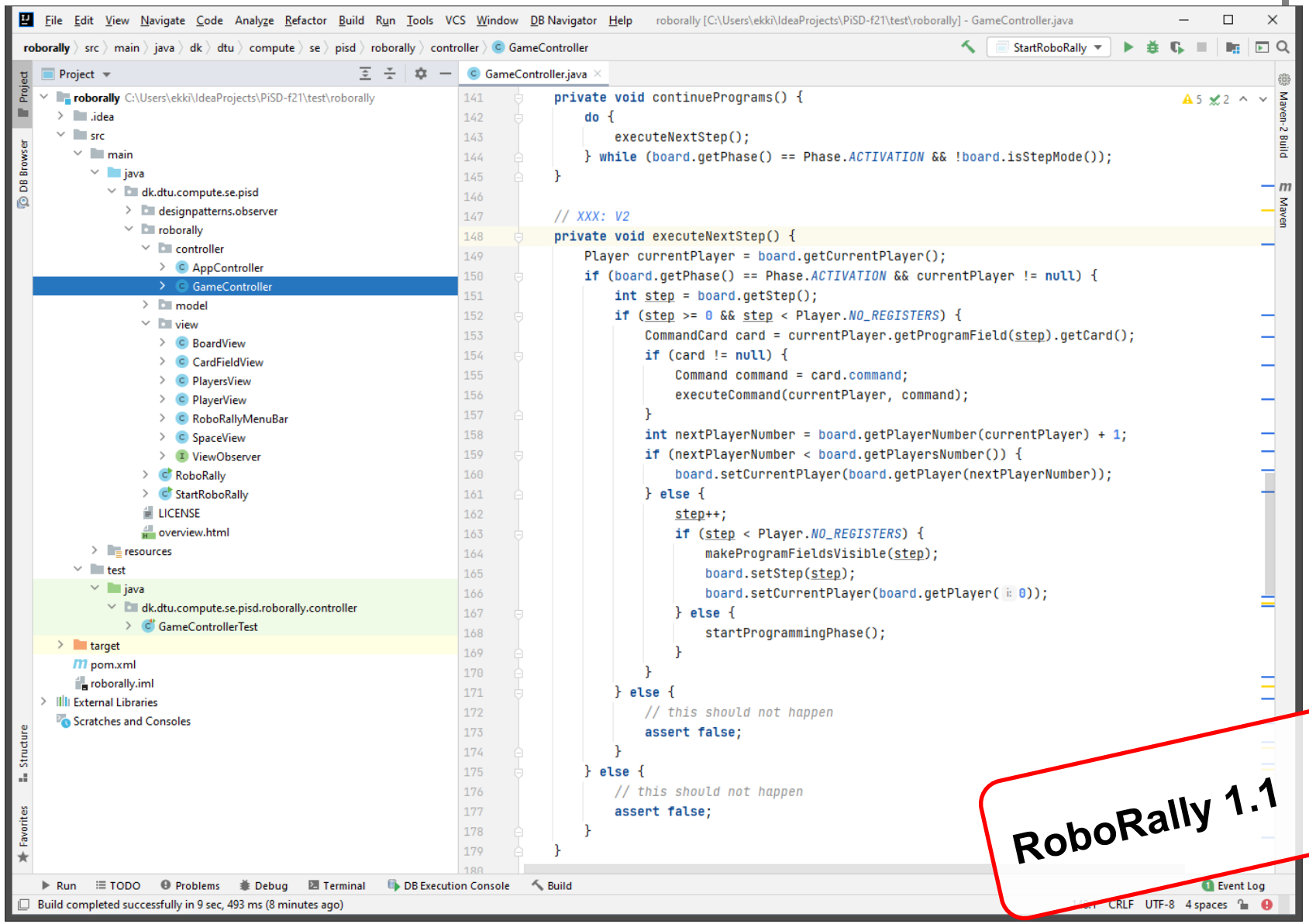
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Structure



Diskussion: Eksempel

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

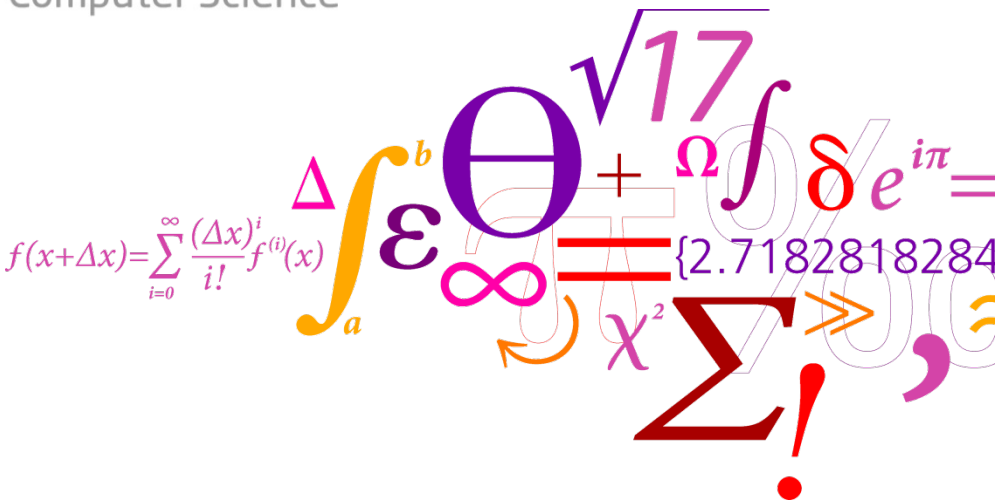


Designkapitlet (II) fortsætter vi med sener!

III. Analyse og konceptuelle modeller

Nu starter vi forfra igen med analysen eller ”hvad” softwaren skal gøre!

DTU Compute
Department of Applied Mathematics and Computer Science



Motivation (fra L01)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

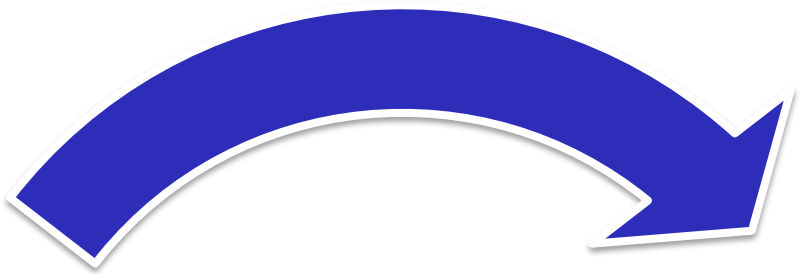
Analysen drejer sig **kun** om **HVAD** (engl. **WHAT**) softwaren skal gøre (ud fra brugerens synsvinkel)

Analysen siger **ikke noget** om **HVORDAN** (engl. **HOW**) softwaren skal realiseres og implementeres

Co-evolution (fra L01)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

"**Hvad**" skal softwaren gøre?



WHAT HOW

*Realiteten er mere kompliceret, men at adskille "**what**" og "**how**" i jeres hoved er et første.*

"Hvordan" er softwaren realiseret?

1. Motivation

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Inden man kan implementere et stykke software er man nødt til af finde ud af **hvad** selve software skal gøre og formulere det fra brugerens synsvinkel:

- Hvilke brugere er der?
- Hvilke begreber (engl. concepts) spiller en rolle
- Hvordan hænger begreberne sammen?
- Hvilke funktioner (use-cases/aktiviteter) er der? Hvad indebærer de og hvornår kan man gennemføre dem?

I vores projekt er det mest af alt: RoboRally's regler

Hvordan gøre man det

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Find skriftlig dokumentation om området (domænet) og skriv det ud
- Marker alle begreber som synes relevante
- Tal med nogle eksperter fra domænet
- Diskuter indbyrdes og saml informationerne op og strukturer den

Hvordan gøre man det

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- For at samle informationerne op give dem struktur:

→**Taksonomi**: liste af begreber (ord)

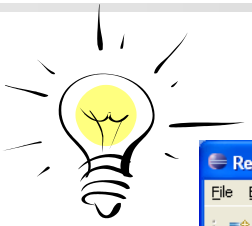
→**Glossar**: Beskrivelse af begreber deres egenskaber og hvordan de forholder sig til hinanden

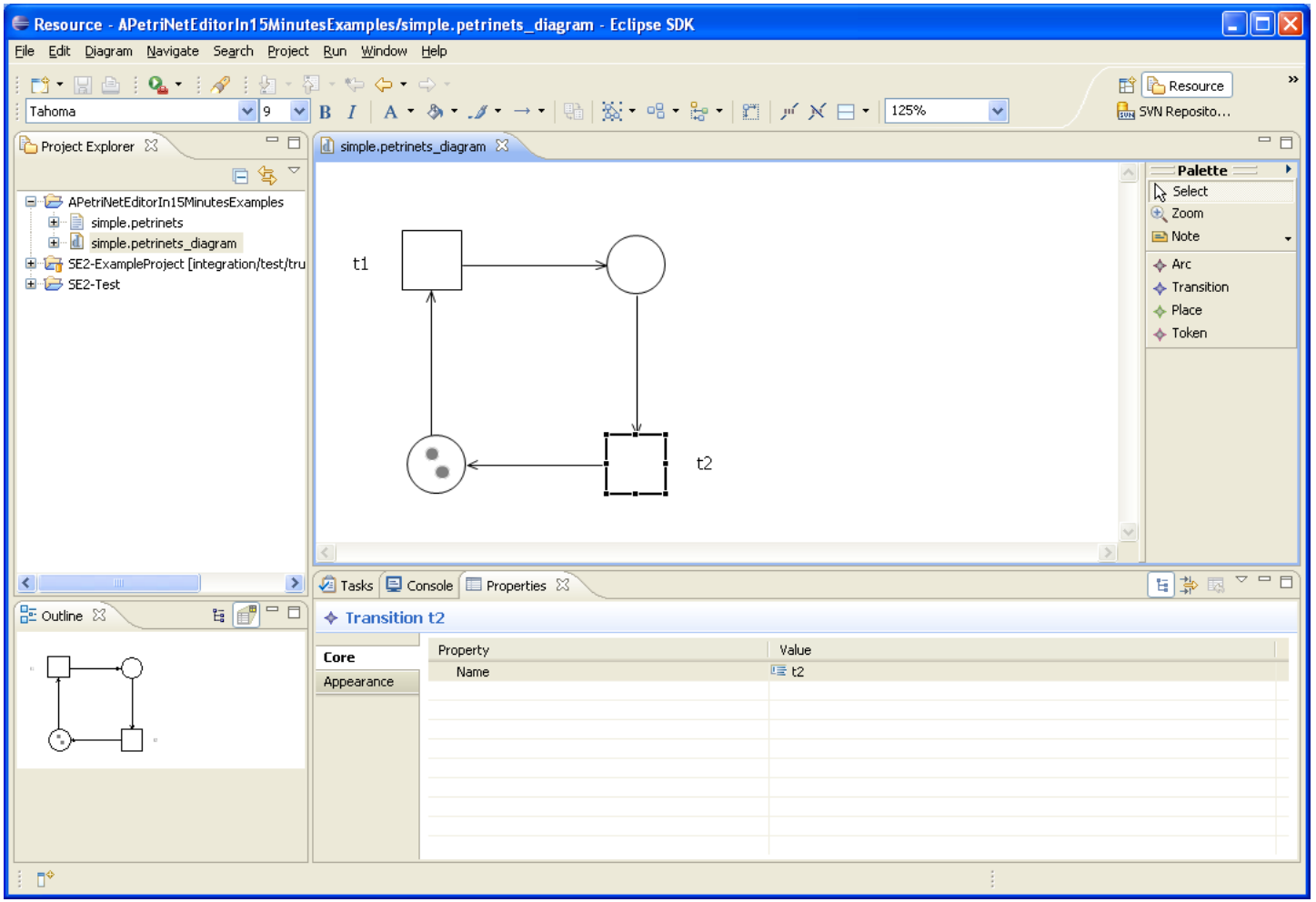
→**Domænemodel**: Begreber, deres attributter og deres relation repræsenteret som klassediagram og beskrivelse af aktiviteter og begrebernes tilstande (aktivitets- og tilstandsdiagrammer)

Disse diagram har ikke noget med programmering at gøre (endnu). Tænk kun på begreber og deres relation!

2. Eksempel: Petrinet

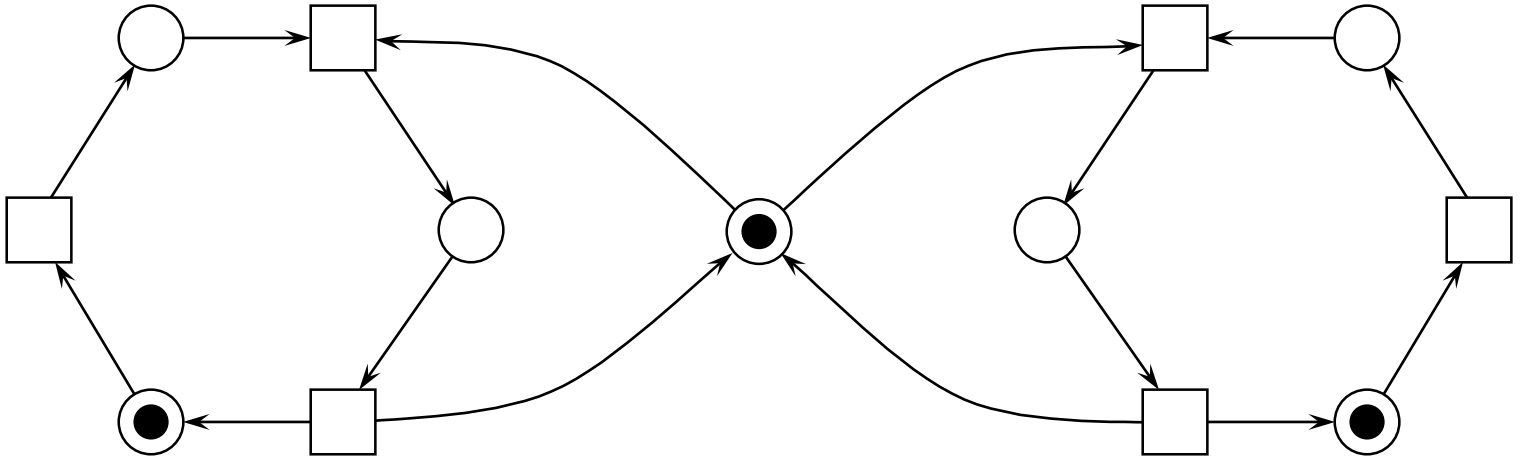
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler





Petrinet

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



Trin

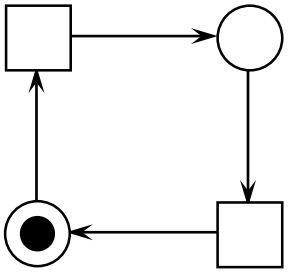
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

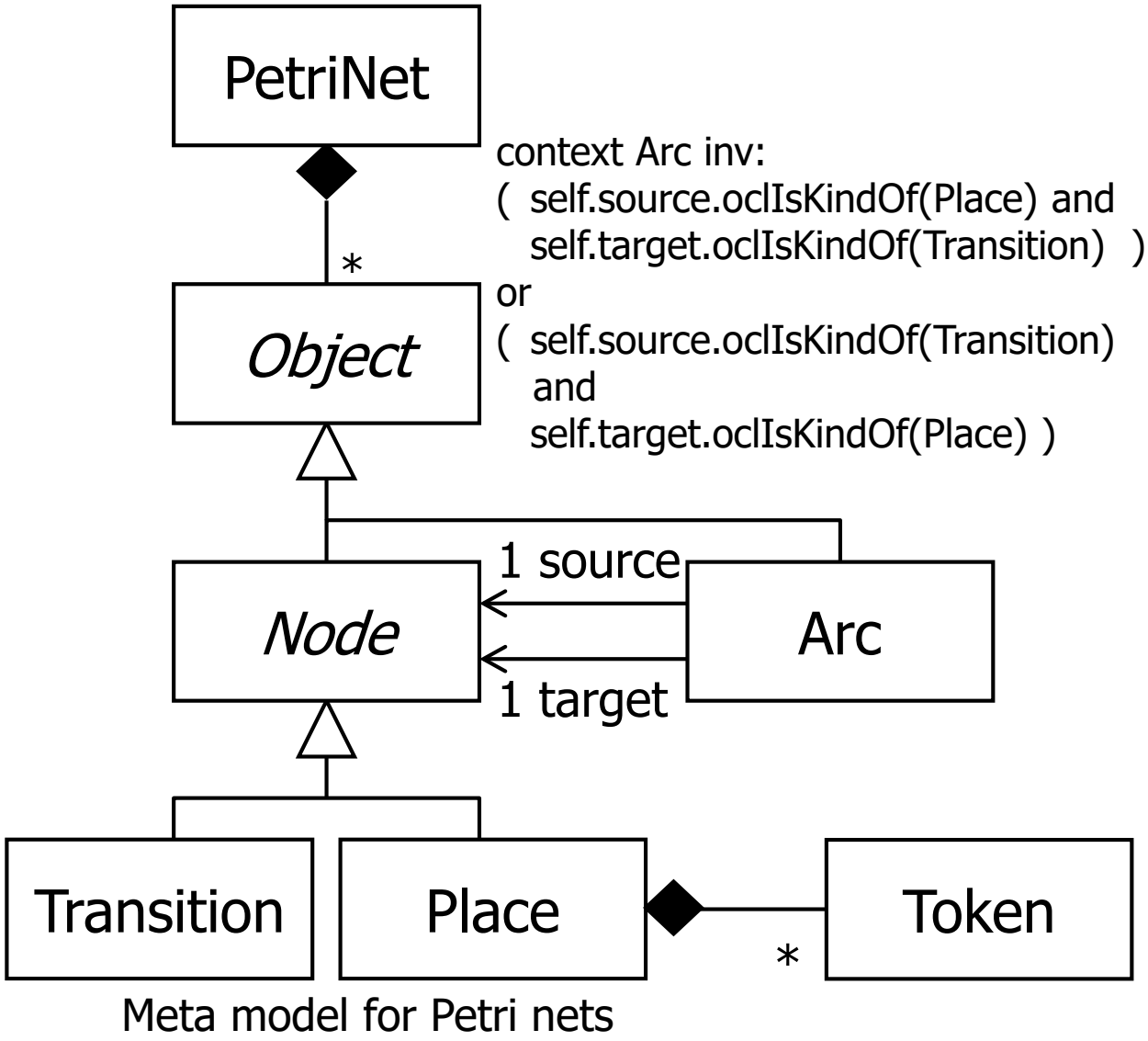
- Eksempler
- Taksonomi (på tavlen)
- Glossar
- Model (på tavlen)

Regel: I skal først starte med at lave et UML diagram, hvis I har kigget på nogle eksempler først og har en liste med hovedbegreberne (taksonomi)!

Model (Meta Model)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



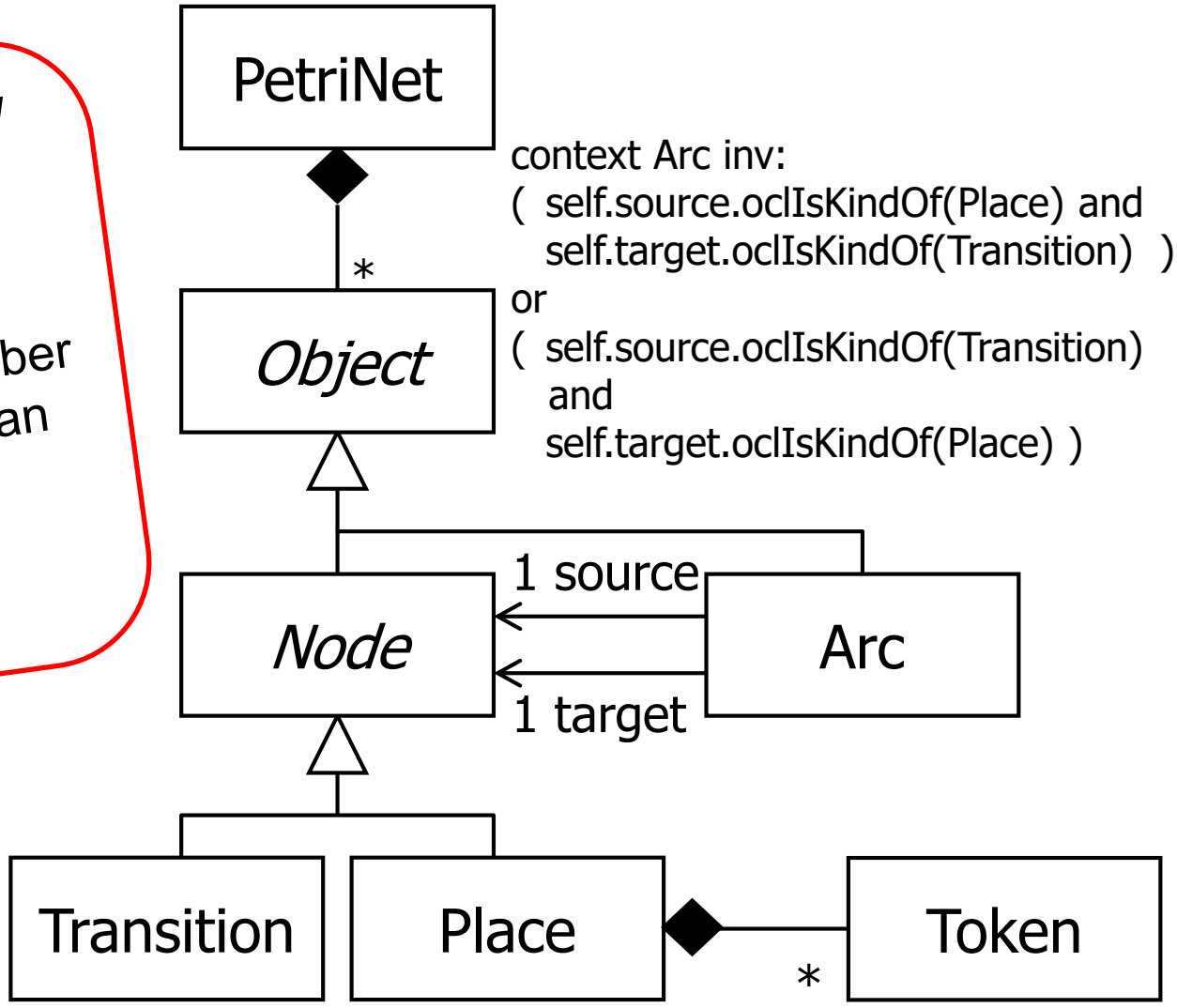


Meta model for Petri nets

Model (Meta Model)

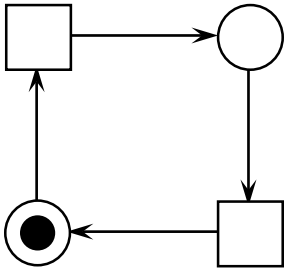
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Regel: Under analysen / konceptuel modellering, tænk ikke på programmering eller Java! Det er kun begreber fra domænet og hvordan de relater til hinanden (Petrinet i vores eksempel)!



Syntaks (abstrakt and konkret)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

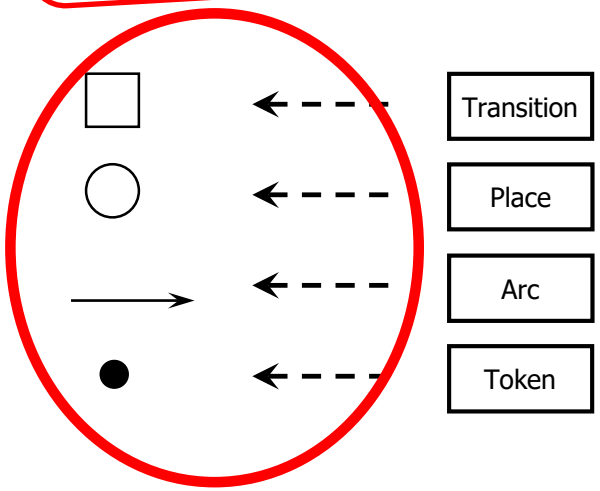


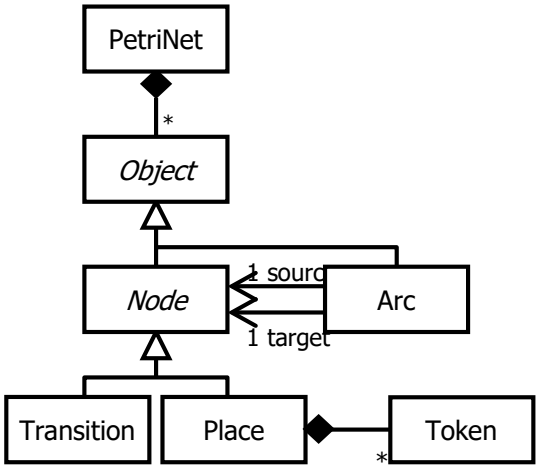
graphical /
concrete syntax

abstract syntax
(as an UML object diagram)

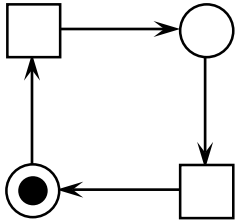
GMF

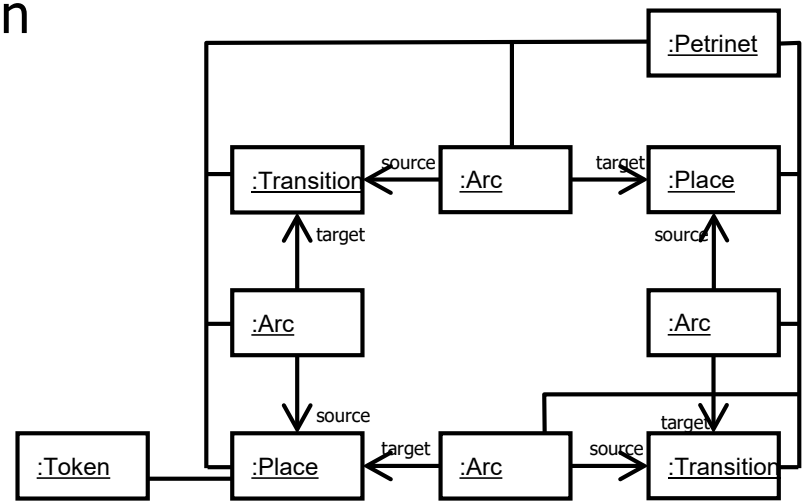
GMF = Teknologi til automatisk generering af editorer (men ikke relevant i dette kursus)!





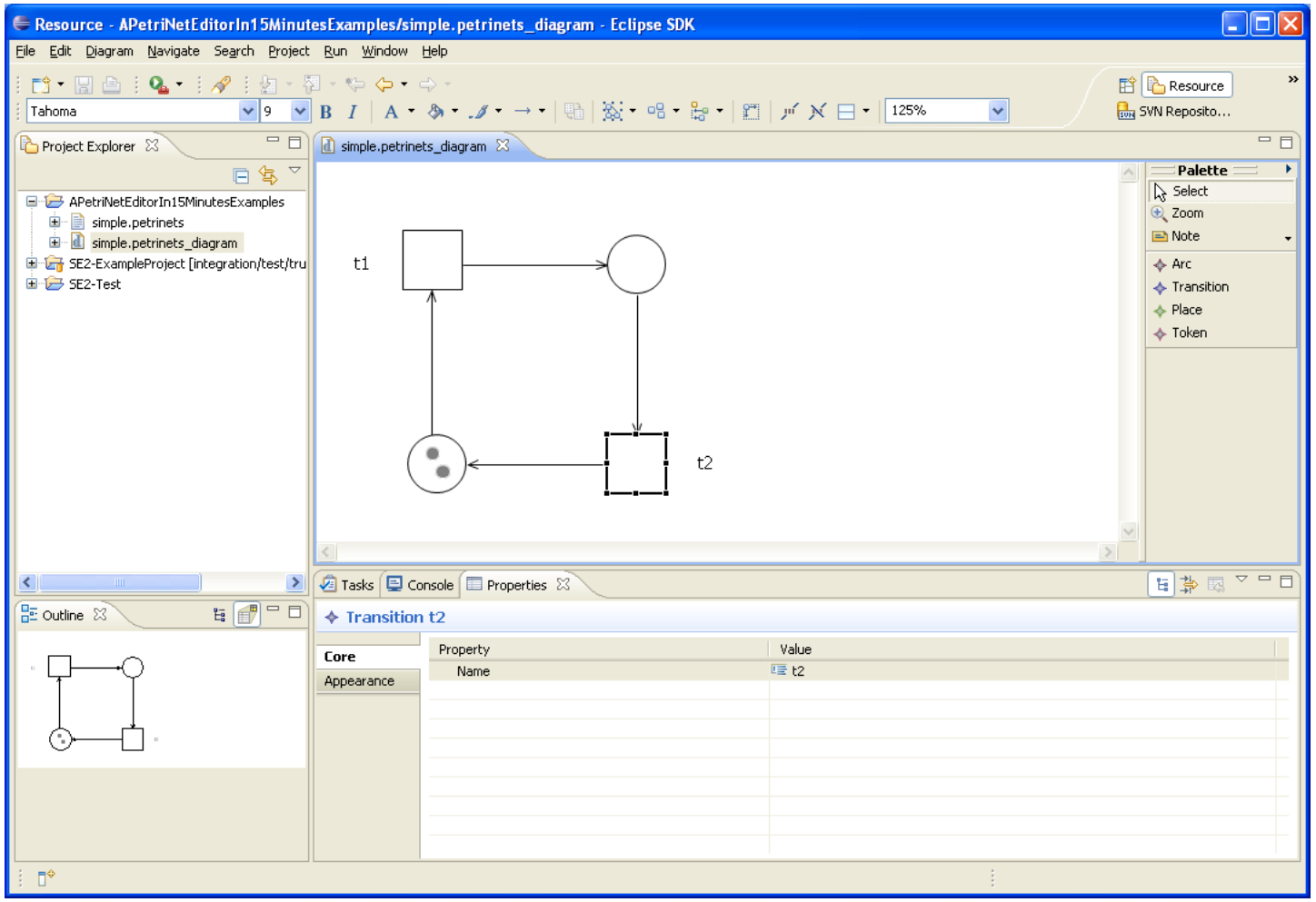
generate an editor





Resultat

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



3. Eksempel: RoboRally

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Vi gennemgår processen med hjælp af udtræk af vores RoboRally-projekt

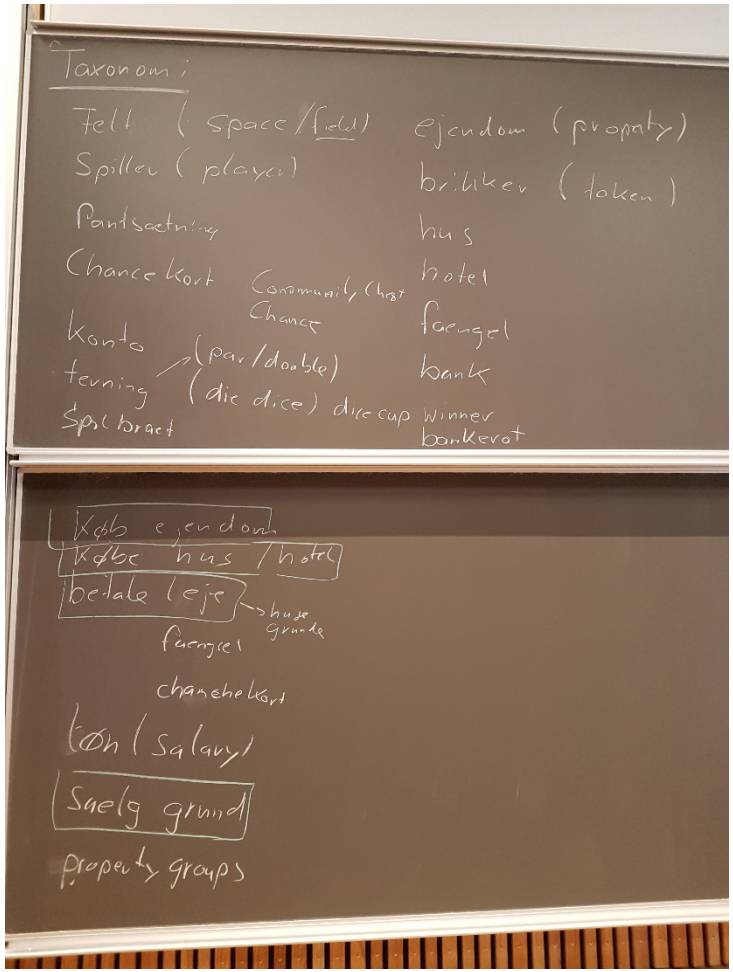
1. Find beskrivelsen af RoboRally frem (se projektslides og især links til RoboRally's regler)
2. Taksonomi
3. Glossar
4. Modeller (klassediagrammer, tilstandsdiagrammer, aktivitetsdiagrammer)

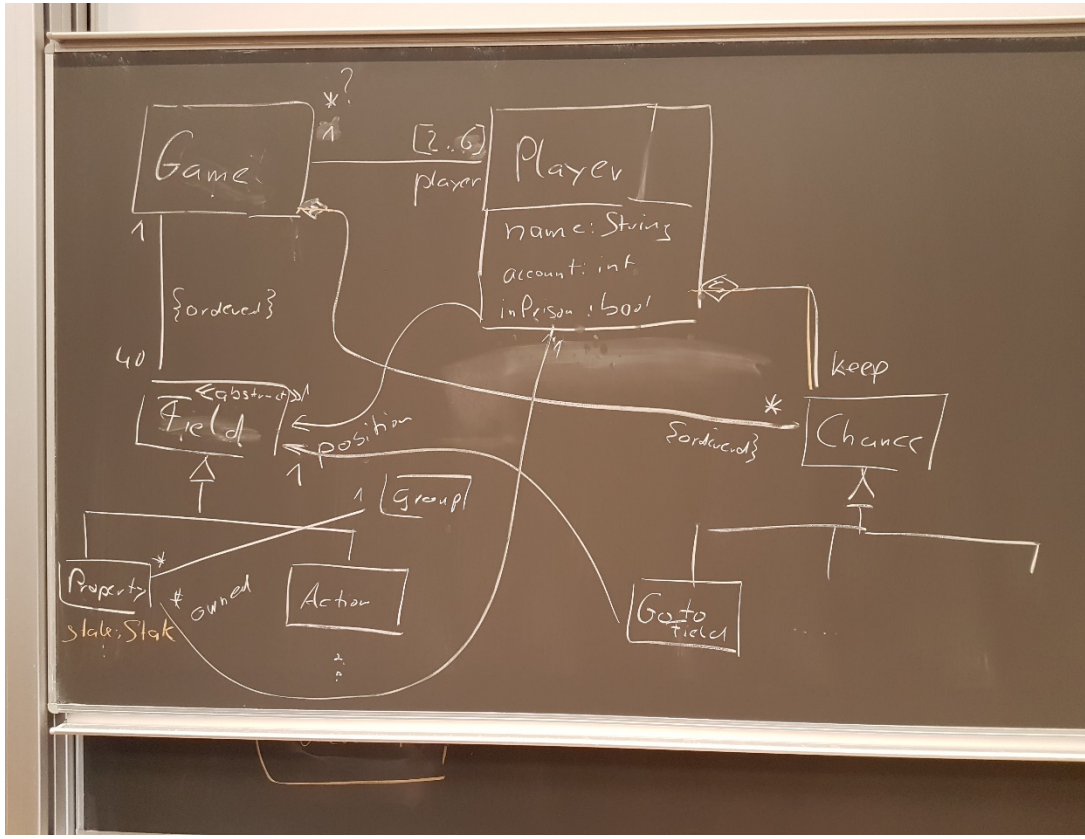
Det springer vi over her, men det kan gerne være en del af jeres rapport!

Diskuter i jeres i gruppe!

Senere under live-delen : diskussion med hele klassen

Fra de tideligere år: Monopoly/Matador





Livscyklus af udvalgte objekter

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Vi starter med livscyklus af udvalgte objekter, da de er typisk nemmere og tætter på klassediagrammet:

- spil
- spiller (== robot ?)
- felt
- kort
- …

Diskussion i grupper og på tavlen; vi bruger tilstandsdiagrammer (UML state machines) til det

Faktisk kunne nogle af de der tilstande være et attribut af klasser i klassediagrammet (eller de kan afledes af nogle af objektets andre attributter.

Aktiviteter

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

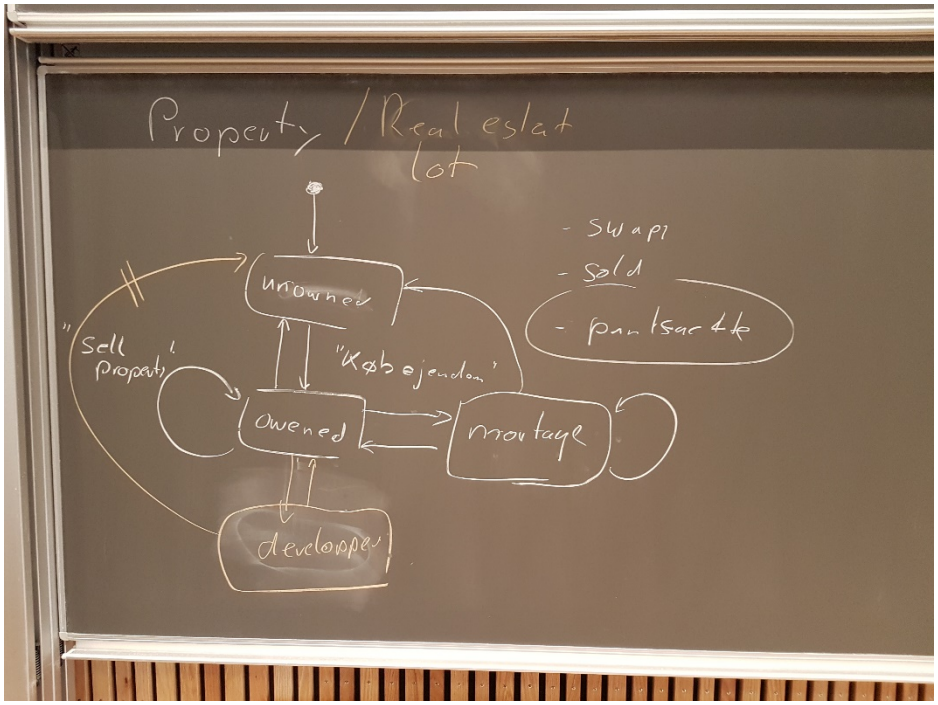
Vi diskuterer udtræk af nogle udvalgte aktiviteter:

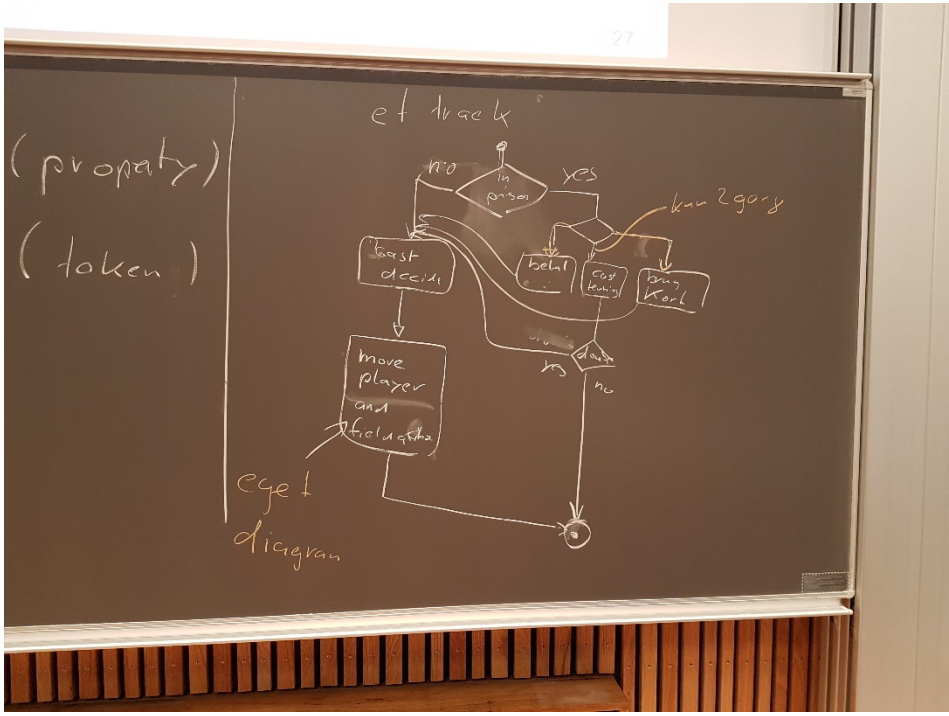
- lav et træk???
- …

Diskussion i grupper og på tavlen; vi bruger aktivitetsdiagrammer (UML activity diagrams) til det

Bemærk at nogle aktiviteter bliver udført som del af nogle andre aktiviteter; og aktiviteter kan køre også sideløbende med andre aktiviteter

Fra de tideligere år:
Monopoly/Matador





Analysen

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Ud over taksonomien og diagrammerne, skal analysen også indeholde

- Tekst som beskriver diagrammerne
- Diskussion af krav, som ikke er beskrevet igennem taksonomien og diagrammerne (ikke-funktionelle krav):
  - brugbarhed
  - performance
  - vedligeholdbarhed
  - …
- GUI

I vores projekt kommer hoveddelen af GUIen fra mig og det er ikke hovedfokus! Derfor kan dens beskrivelse være meget kort (udover nogle særlige features fra jer).

Opgave A1 (til 22./23. 2.)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

http://www2.compute.dtu.dk/courses/02362/f21/opgaver/V02/

Bemærk at opgave A1 beskrives sammen med opgave V2!

Detaljerede analyse/konceptuelle modeller til RoboRally (**afleveringsopgave**):

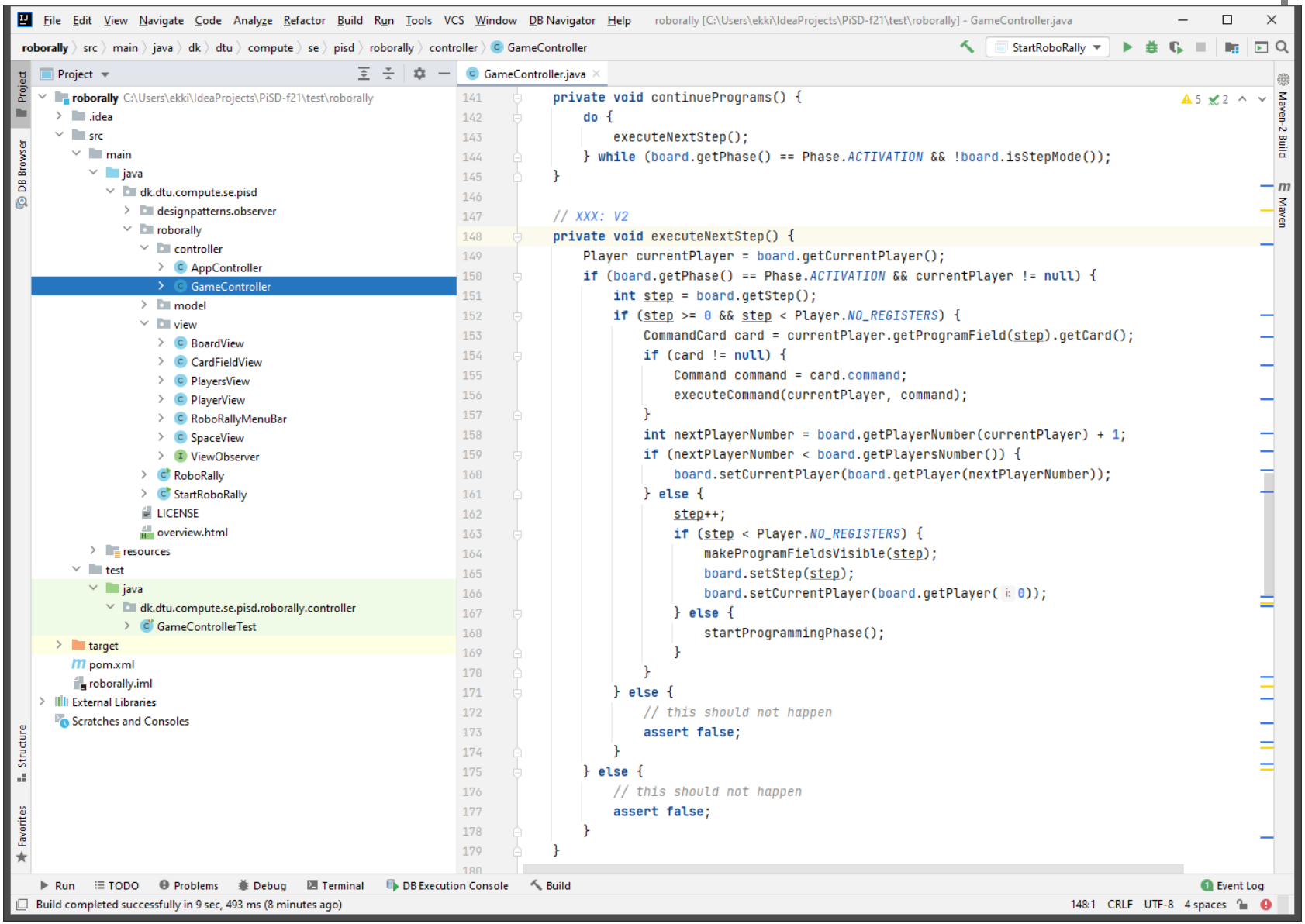
- Taksonomi (tilstandsbegreber / aktionsbegreber)
- Klassediagrammer

De er først påkrævet med aflevering A2: Projektdefinition

- Tilstandsdiagrammer til relevante klasser (frivilligt)
- Aktivitetsdiagrammer til relevante aktiviteter (frivilligt)

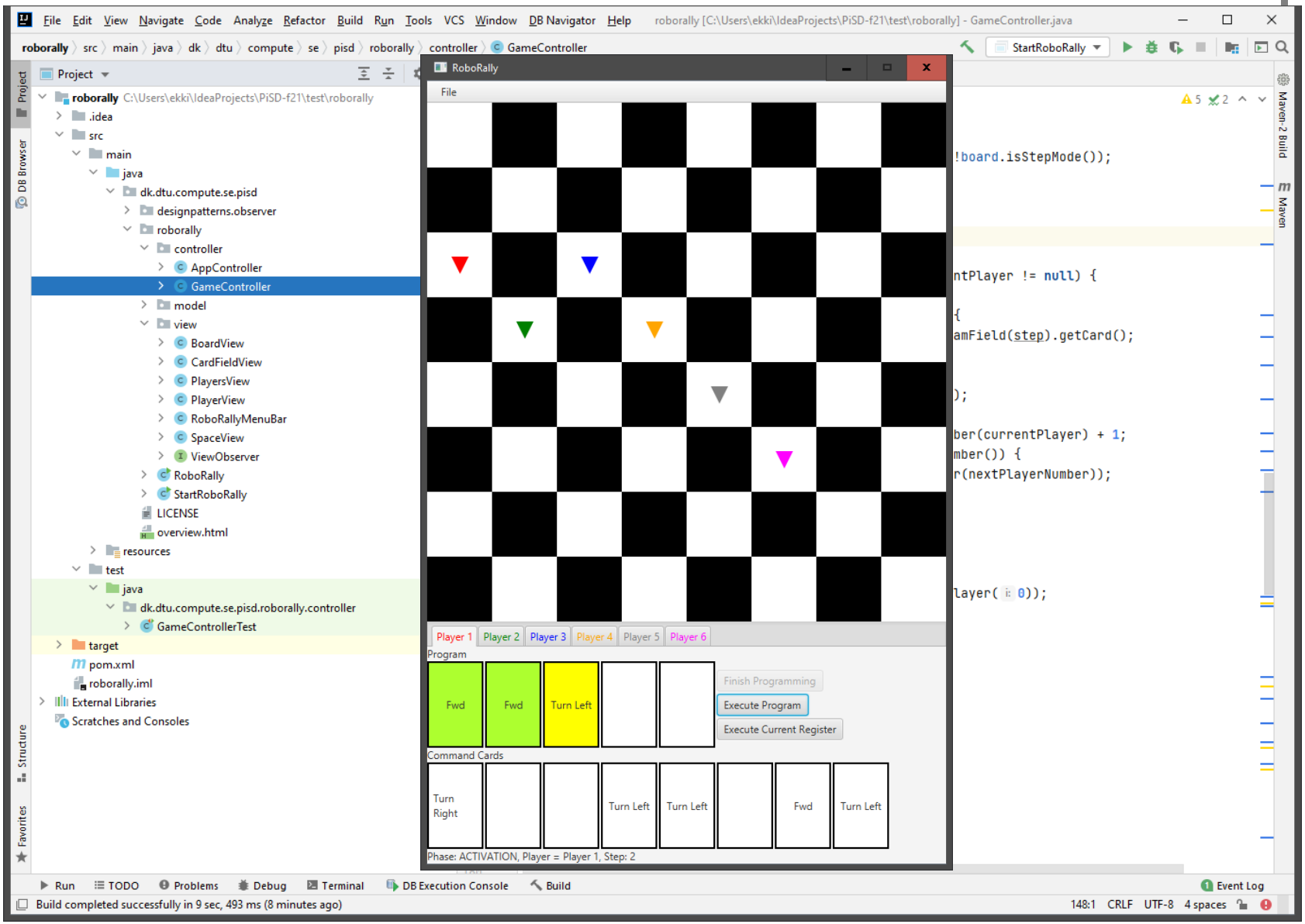
RoboRally 1.1.0 → ???

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



RoboRally 1.1.0 → ???

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



Opgave V2 (til 15./16. 2.)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

http://www2.compute.dtu.dk/courses/02362/f21/opgaver/V02/

I får udleveret en enkel implementering af RoboRally-spillet (udvidet fra Opgave V1):

- Tilknyt knapperne (i viewer) til de rigtige metoder i GameController
- Implementer robottens kommandoer i GameController
- Tilføj test til de ny-implementerede metoder i GameController

Flere detaljer om opgaven og løsningen i en separat præsentation