

CDIO3

Gruppe 16

10. november 2017



Mathias
Fager
s175182



Milishia
Moradi
s175193



Nicki
Christiansen
s170208



Semi
Seitovski
s175181



Simon
Pedersen
s175195



Thyge
S. Steffensen
s175176

DTU

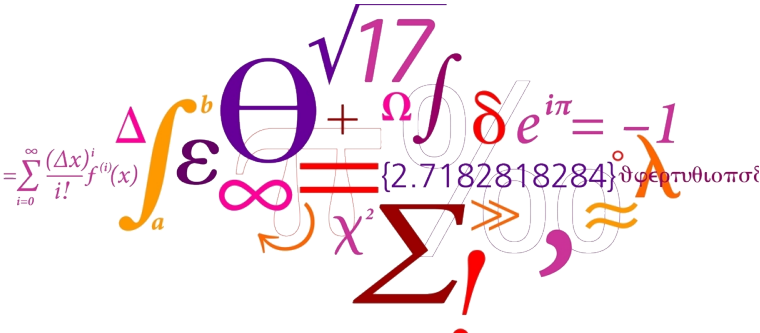


Timeregnskab

Timeregnskabet kan ses via linket [her](#) eller i billag XX.

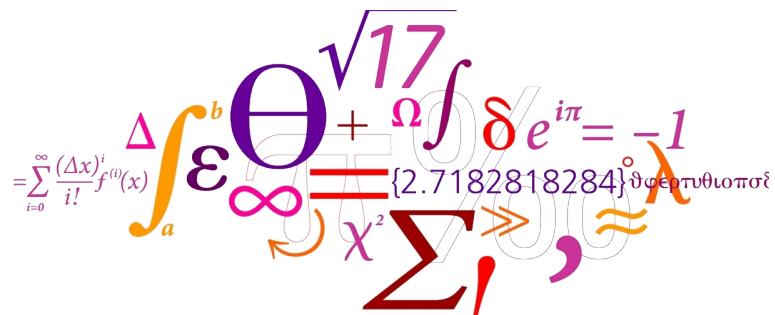
Deltager	Total timer per deltager
Mathias Fager	3
Milishia Moradi	3,5
Nicki Christiansen	10
Semi Seitovski	6
Simon Steen Pedersen	5,5
Thyge Steffensen	14,5
Total	42,5

Figur 1: Overblik over Timeregnskab den 21/11/2017



Indhold

1	Analyse	3
1.1	Kravliste	3
1.2	UseCases	4
1.3	UseCase diagram	7
2	Design	8
2.1	Klasse diagram	8
2.2	Sekvensdiagram	9
2.3	System sekvensdiagram	10
2.4	Domænemodel	11
3	Dokumentation	12
3.1	Forklar hvad arv er	12
3.2	Forklar hvad abstract betyder	12
3.3	Fortæl hvad det hedder hvis alle fieldklasserne har en landOn- Field metode der gør noget forskelligt	13
3.4	Dokumentation for test med screenshots	13
3.5	JUnit test	13
3.6	Positiv negativ test	13
3.7	Black- og Whitebox test	14
3.8	Dokumentation for overholdt GRASP	14
3.9	Vejledning til import af Git Repository til eclipse	15
3.10	Brugertests	16



$$= \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \int_a^b \varepsilon \Theta + \Omega \int \delta e^{i\pi} = -1$$

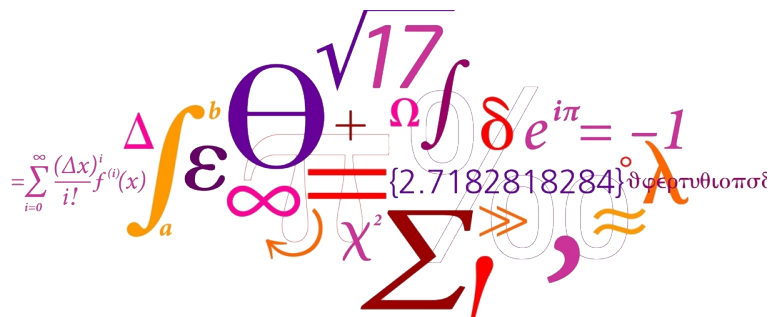
1.2 UseCases

UseCase Section: Opsætning af spil	Comment
Scope	Monopoly spil af IOOuterActive
Level	User-goal
Primær Aktør	Spillerne
Stakeholder og interesser	Spillerne er interesserede i at kunne starte spillet ved at vælge antal spillere og deres brikker
Forudsætninger	Spillet bliver kørt, og spillerne har nu mulighed for at vælge antal spillere og ønskede brikker
Success garanti	Der er blevet valgt antallet af spillere, og hver spiller har valgt sin brik, herefter er spillet klar til at blive spillet

$$\begin{aligned} & \Delta \int_a^b \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} = -1 \\ & = \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \\ & \quad \epsilon \infty = \{2.7182818284\} \quad \text{οφείνται οπισθί} \\ & \quad \chi^2 \sum \gg \approx \lambda \end{aligned}$$

Fully dressed UseCase:

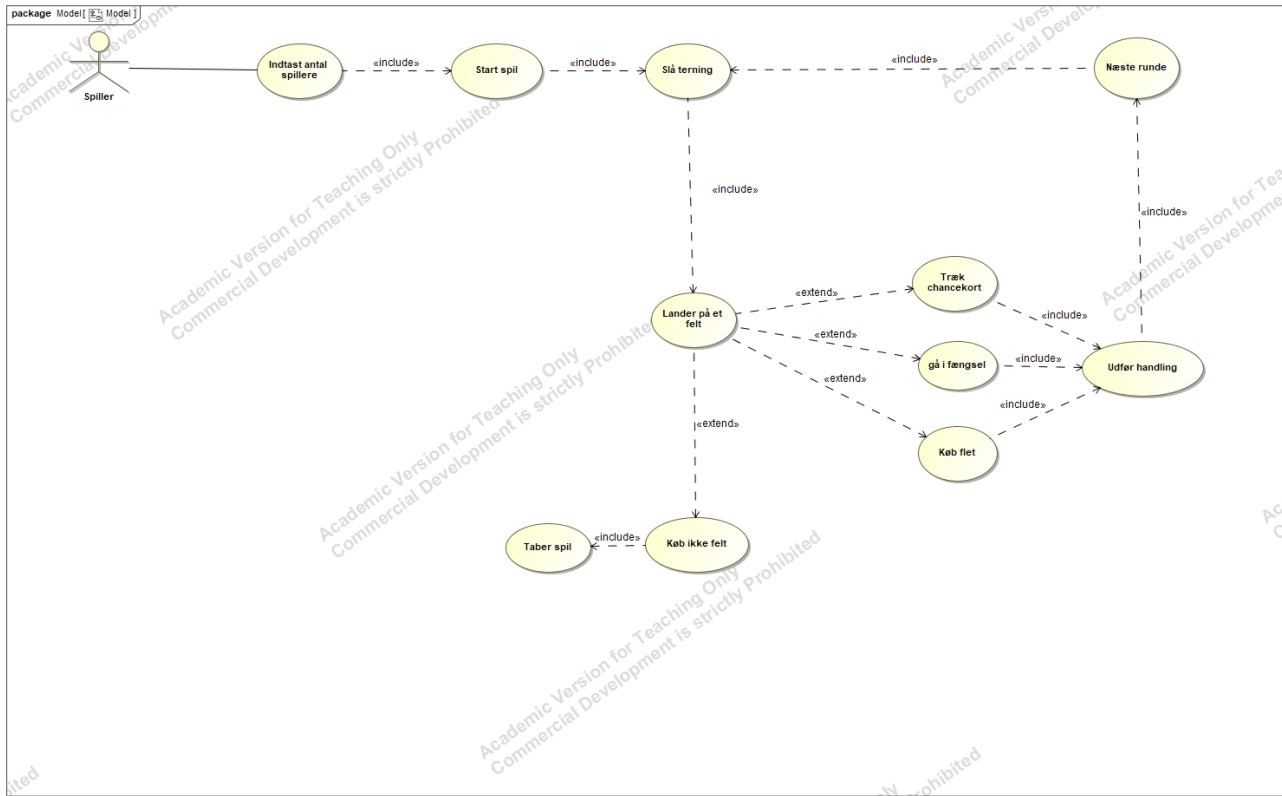
UseCase Section: Spillerne slår med terningerne	Comment
Scope	Monopoly spil af IOOuterActive
Level	User-goal
Primær Aktør	Spillerne
Stakeholder og interesser	Spillerne er interesseret i at kunne trykke på en knap, og få et billede af to terninger med tilfældige værdier
Forudsætninger	Spillet er startet op, og spillerne har valgt antallet spillere og deres ønskede brikker
Success garanti	Der er blevet valgt antallet af spillere, og hver spiller har valgt sit navn, herefter er spillet klar til at blive spillet
Hoved succes scenarie	Spillerne får udgivet en værdi af to terninger, og lander derefter på et felt
Alternative udfald	Negative udfald: - IOOuterActive har opdateret spillet, og derved opstår der en fejl når spillerne slå med terningerne, der kan ende i at der ikke bliver slået to terninger - Systemet blokerer for en spillers tur - En spiller hopper fra/på, og derved skal spillet startes om
Specielle krav	- Enheden som spillet kører på skal være kompatibel med Java - Spillerne skal kunne interagere med GUI'en ved brug af mus eller touch - Der skal være plads på enheden til at kunne hente spillet
Hyppighed	Hver tur bliver der slået med terninger



UseCase Section: Spillet afsluttes	Comment
Scope	Monopoly spil af IOOuterActive
Level	User-goal
Primær Aktør	IOOuterActive
Stakeholder og interessenter	IOOuterActive er interesseret i at programmet viser en vinder og afsluttes
Forudsætninger	Alle spillere undtagen en, har fået en balance på 0
Success garanti	Spillet viser en vinder og kan derefter afsluttes

$$= \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \int_a^b \epsilon \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} = -1$$

1.3 UseCase diagram



Figur 2: UseCase diagram tegnet i MagicDraw

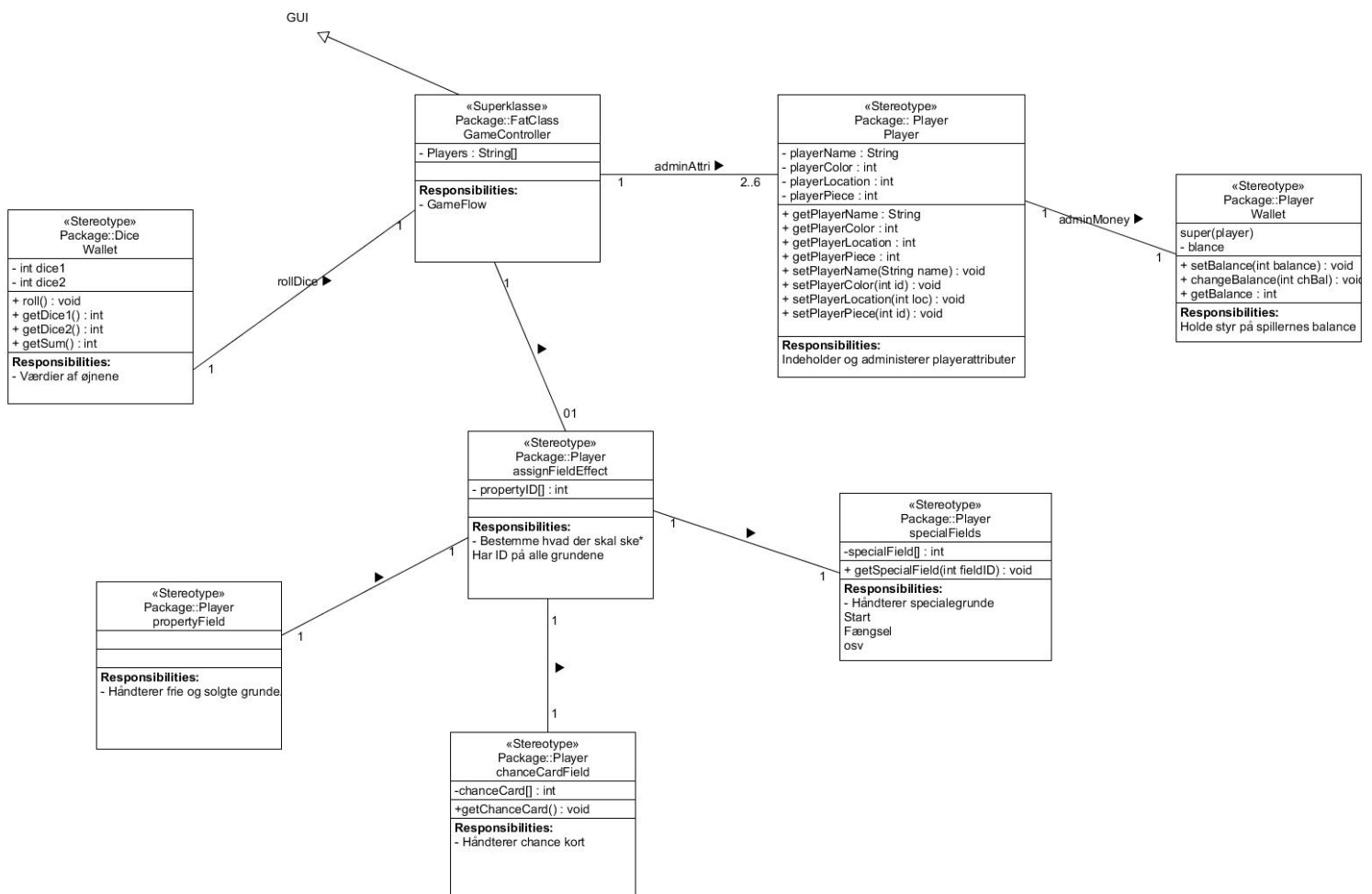
$$\sum_{i=0}^{\infty}\frac{(\Delta x)^i}{i!}f^{(i)}(x) \int_a^b \epsilon \Theta + \Omega \int \delta e^{i\pi} = -1$$
$$=\sum_{i=0}^{\infty}\frac{(\Delta x)^i}{i!}f^{(i)}(x)$$
$$\int_a^b \epsilon \Theta + \Omega \int \delta e^{i\pi} = -1$$
$$=\{2.7182818284\}$$
$$\chi^2$$
$$\Sigma$$
$$>$$
$$\approx$$
$$\lambda$$

© 2016 Pearson Education, Inc. or its affiliate(s). All rights reserved.

2 Design

Herunder ses en række 'design steps', som skal hjælpe os med at lave Monopoly Junior spillet.

2.1 Klasse diagram

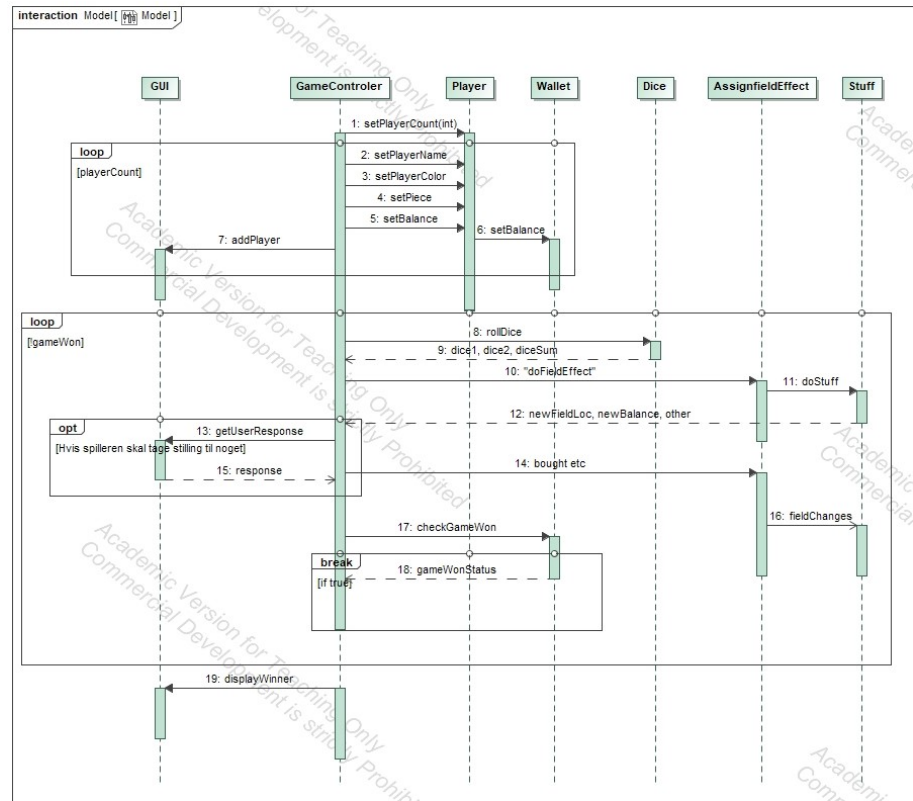


Figur 3: Klasse diagram tegnet i UMLet

Klasse diagrammet bygger på vores umiddelbare overvejelser, såvel som vores use case's. Dette er for at illustrere sammenspillet mellem vores klasser og deres associationer. Dog skal det siges, at dette er en skitse og den aktuelle programmering kan variere heraf.

$$\begin{aligned}
 &= \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \\
 &\int_a^b \epsilon \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} = -1 \\
 &= \{2.7182818284\} \approx \chi^2 \sum_{i=1}^{\infty} \frac{1}{i^2} \approx \frac{6}{\pi^2}
 \end{aligned}$$

2.2 Sekvensdiagram



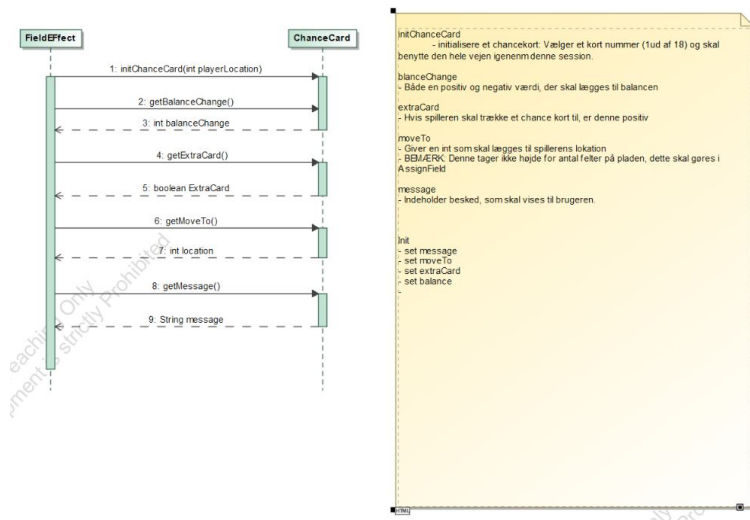
Figur 4: Sekvensdiagram tegnet i MagicDraw

Vi har her lavet et sekvensdiagram, der skal skabe et overblik over hvordan aktøren, her spilleren, kommunikerer med spillet.

$$= \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \quad \int_a^b \epsilon \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} = -1$$

$$\infty = \{2.7182818284\} \quad \chi^2 \sum \gg \approx \lambda$$

2.3 System sekvensdiagram



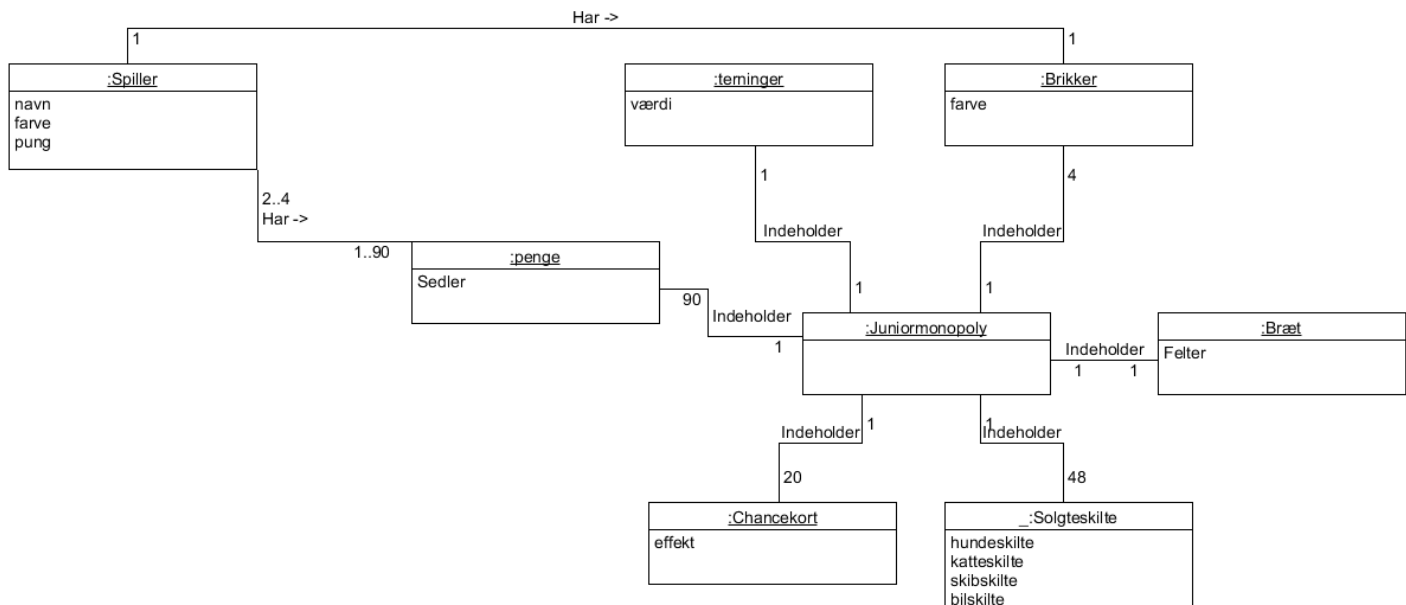
Figur 5: Systemsekvensdiagram tegnet i MagicDraw

Vi har her lavet et systemsekvensdiagram, for at forhøje gennemsigtigheden ved bruge af 'chanceCard' klassen.

$$= \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \int_a^b \epsilon \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} = -1$$

$$\{2.7182818284\} \approx \chi^2 \sum_{i=1}^{\infty} \frac{1}{i!}$$

2.4 Domænemodel



Figur 6: Domænemodel tegnet i UMLet

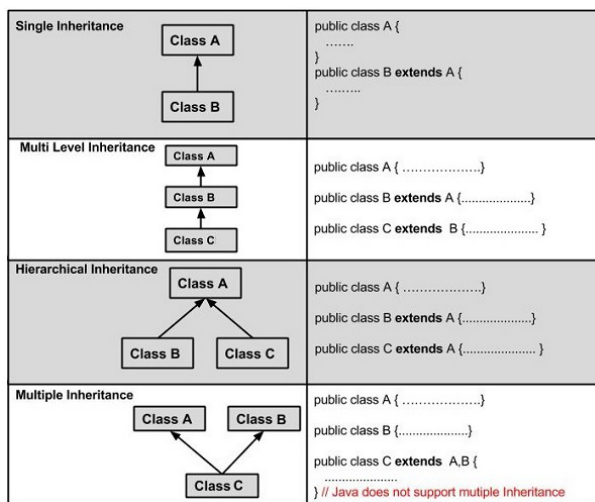
Ved hjælp af domænemodellen vil vi trække paralleller mellem den virkelige verden og programmeringen. Domænemodellen er en visuel repræsentation af konceptklasser og 'objekter fra den virkelige verden'. Ved hjælp af denne kan vi også oplyse kunden om, hvad vi vil lave.

$$\begin{aligned}
 &= \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \\
 &\int_a^b \epsilon \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} = -1 \\
 &= \{2.7182818284\} \text{ } \partial \phi \epsilon \tau \theta \iota \omega \pi \sigma \xi \\
 &\chi^2 \sum \gg \approx \lambda
 \end{aligned}$$

3 Dokumentation

3.1 Forklar hvad arv er

Et af de vigtigste elementer i objektorienteret programmering er nedarvning, men hvad betyder nedarvning? At arve betyder i bund og grund at noget med en materiel værdi overdrages til noget eller nogle efterkommere. I programmerings verden ville man hellere definere arv som en eller flere egenskaber der overføres fra en generation til en anden generation, også sagt på en anden måde, at en klasse der får overført egenskaber fra en anden klasse. Den klasse der nedarves fra kaldes for en Superklasse (parent), hvor klassen der nedarver eller får egenskaber fra en Superklasse, kaldes for en Subklasse (child). Ved benyttelse af denne proces vil oplysningerne om de forskellige klasser ende op i en hierarkisk rækkefølge. På figur 7 ses forskellige eksempler på typer af nedarvning der findes, men dog skal man være opmærksom på, at Java ikke understøtter flere nedarvninger. I Java kan en klasse kun have én Superklasse, dvs. at hver klasse kun kan nedarve fra én klasse.¹



Figur 7: eksempel på typer af arv

2

3.2 Forklar hvad abstract betyder

Den abstrakte klasse kører ingen metoder, og indeholder kun attributter. Klassen bruges som en slags "over"-klasse, hvor andre klasser nedarver attri-

¹ Java Programmering - En bog for begyndere af Henrik Kressner

²https://www.tutorialspoint.com/java/java_inheritance.htm

$$= \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \quad \Delta \int_a^b \epsilon \Theta + \Omega \int \delta e^{i\pi} = -1$$

$$\infty = \{2.7182818284\} \quad \chi^2 \quad \sum \quad \approx \quad \lambda$$

$$\sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \int_a^b \epsilon \Theta + \Omega \int \delta e^{i\pi} = -1$$

Finished after 0.041 seconds

Runs: 2/2 Errors: 0 Failures: 1

test.TestDice [Runner: JUnit 4] (0.023 s)

testRollFrequencyTerning1 (0.019 s)

testGetDiceValue1 (0.004 s)

Failure Trace

java.lang.AssertionError: Hyppigheden af de forskellige forekommer ikke lige ofte
at test.TestDice.testRollFrequencyTerning1(TestDice.java:65)

```

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
Dice.java
TestDice.java
/**
 * Tester om terningen ligger inde for spektret,
 */
@Test
public void testGetDiceValue1() {
    for (int i = 0; i < 100000; i++) { // Ruller terning 1 100000 gange
        dice.roll();
        if (dice.getDiceValue1() < 1 || dice.getDiceValue1() > 6) {
            fail("Roll-returværdi er udenfor spektret 1-6");
            break;
        }
    }
}
/**
 * Tester om hyppigheden for hvert slag er lige stor, for terning 1.
 */
@Test
public void testRollFrequencyTerning1() {
    //Definere alle øjnene
    int count1d1=0, count2d1=0, count3d1=0, count4d1=0, count5d1=0, count6d1=0;

    //Tester for 100000 kast og ser om hyppigheden for hvert slag er lige stor for terning 1
    for (int i = 0; i < 100000; i++) {
        dice.roll();
        switch(dice.getDiceValue1()) {
            case 1: count1d1++;
            break;
            case 2: count2d1++;
            break;
            case 3: count3d1++;
            break;
            case 4: count4d1++;
            break;
            case 5: count5d1++;
            break;
            case 6: count6d1++;
            break;
            default: fail("Virker ikke");
            break;
        }
    }
}

```

Problems Console Progress Git Staging Git Repositories History Coverage

Element Coverage Covered Instructions Missed Instructions Total Instructions

Figur 8: JUnit test udført i Eclipse den 23/11-2017

3.7 Black- og Whitebox test

Blackbox test: Her har man intet kendskab til koden, ud over hvad denne skal kunne gøre. Det er derfor lige til højrebænet, at teste det faktiske output mod det forventede output og at teste ækvivalensklasserne. **Whitebox test:** Her opstiller man tests på baggrund af koden og dens opbygning. Man vil med disse tests teste hele koden, altså alle instruktioner, forgreninger og stier, som koden kan blive udført i. Dette er ikke altid muligt med *BlackBox testen*, da man her, ikke har et indblik i koden.

3.8 Dokumentation for overholdt GRASP

GRASP står for *General Responsibility Assignment Software Patterns*. GRASP bruges til at give det rigtige ansvar til de forskellige klasser, der bliver

$$\begin{aligned}
 & \Delta \int_a^b \epsilon \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} = -1 \\
 & = \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \\
 & \quad \chi^2 \sum_{i=1}^{\infty} \frac{1}{i!} \approx \lambda
 \end{aligned}$$

1. Creator
2. Information expert
3. Low coupling
4. Controller
5. High cohesion
6. Indirection
7. Polymorphism
8. Protected variations
9. Pure fabrication

3.9 Vejledning til import af Git Repository til eclipse

$$\sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \int_a^b \epsilon \Theta + \sqrt{17} \int \delta e^{i\pi} = -1$$

Test person	Hvad man vil teste	Trin	Forventet resultat	Testresultat
Spillere	Antallet af spillere	Vælg antallet af spillere	Man kan være mellem 2 - 4 spillere	Pass
Spillere	Går turen videre	Spillerne slår med terningerne	En spiller slår med terningerne og turen går videre efter	Pass
Spillere	Udskrives der tekst når en spiller lander på et felt	Når en spiller lander på et felt	Der bliver printet en tekst der fortæller om feltet	Pass
Spillere	Køber spilleren feltet når han/hun lander på det	Når spillet er i gang	Feltet bliver opkøbt	Pass
Spillere	Spilleren er tvunget til at betale ejeren af feltet han lander på	Når spillet er i gang	Spilleren betaler ham/hende der ejer feltet	Pass

$$\sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) \int_a^b \epsilon \Theta + \sqrt{17} \int \delta e^{i\pi} = -1$$

Spillere	Får spilleren 2M når de passere start feltet	Når spillet er i gang	Spillerens balance stiger med 2M	Pass
Spillere	Ryger spilleren i fængsel og mister der 2M	Når en spiller lander på "FÆNGSEL"	Spilleren modtager ikke 2M	Pass
Spillere	Skal spilleren give sit omkostningskort eller betale 1M når han/hun er i fængsel	Når en spiller er på "FÆNGSEL"	Spillerens mister sit omkostningskort eller betaler 1M hvis kortet er brugt	Fail
Spillere	Virker chancekort	Når en spiller trækker et chancekort	Spilleren trækker et chancekort og får en effekt	Fail
Spillere	Virker besøg feltet	Når en spiller lander på "Besøg" feltet	Der sker intet og turen går videre	Pass
Spillere	Kan spillerne gå rundt om pladen	Når spillerne har været en runde rundt på pladen	Spillerne kan gå flere omgange rundt på pladen	Pass

$$\begin{aligned}
& \Delta \int_a^b \epsilon \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} = -1 \\
& = \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x) = \{2.7182818284\} \approx \chi^2 \sum_{i=0}^{\infty} \frac{1}{i!}
\end{aligned}$$

