Yatzy Programbibliotek

Brukerveiledning for klienter til programbiblioteket

Det henvises til kodedokumentasjon på selve kildekoden for utførlig dokumentasjon.

Målgruppe

3. parts-utviklere som skal benytte grensesnittet for poengberegning i sine Yatzy-applikasjoner.

Bruk

Her følger brukerveiledning for Yatzy programbibliotek. Funksjoner i biblioteket er tilgjengelig via interfacet IYatzyGrensesnitt og dets implementasjon YatzyGrensesnitt. IYatzyGrensesnitt kan brukes for å mocke ut selve grensesnitt i scenarier med automatisk testing. Slik starter man:

```
IYatzyGrensesnitt yatzy = new YatzyGrensesnitt();
```

Videre er det to metoder som kan brukes for poengberegning som er tilgjengelig.

BeregnPoengSum

Metoden BeregnPoengSum beregner poengsum for et angitt kast og en kategori. Poengsum kan derfor bli 0 poeng dersom terningkombinasjonen ikke er gyldig for angitt kategori. Et kast angis som en streng med 5 tegn, der hvert tegn må være tallene 1-6. Eksempel på kast med tre enere og to toere, som er gyldige for kategori fullt hus:

```
yatzy.BeregnPoengSum("11122", Kategori.FulltHus)
```

Dette kastet vil gi poengsum 7.

```
yatzy.BeregnPoengSum("23442", Kategori.Femmere)
```

Dette kastet vil gi poengsum 0, da det ikke er noen femmere i kastet..

BeregnHøyesteMuligePoengSumForKast

Metoden BeregnHøyesteMuligePoengSumForKast beregner høyeste mulige poengsum og tilhørende kategori som er mulig å oppnå med et kast. Et kast angis som en streng med 5 tegn, der hvert tegn må være tallene 1-6. Eksempel på kast med tre enere, en treer og en toer:

```
yatzy.BeregnHøyesteMuligePoengSumForKast("13121")
```

Dette kastet vil finne at høyeste mulige poengsum er 3, for kategorien “Enere”. Kategori er angitt med en enum “Kategori”.

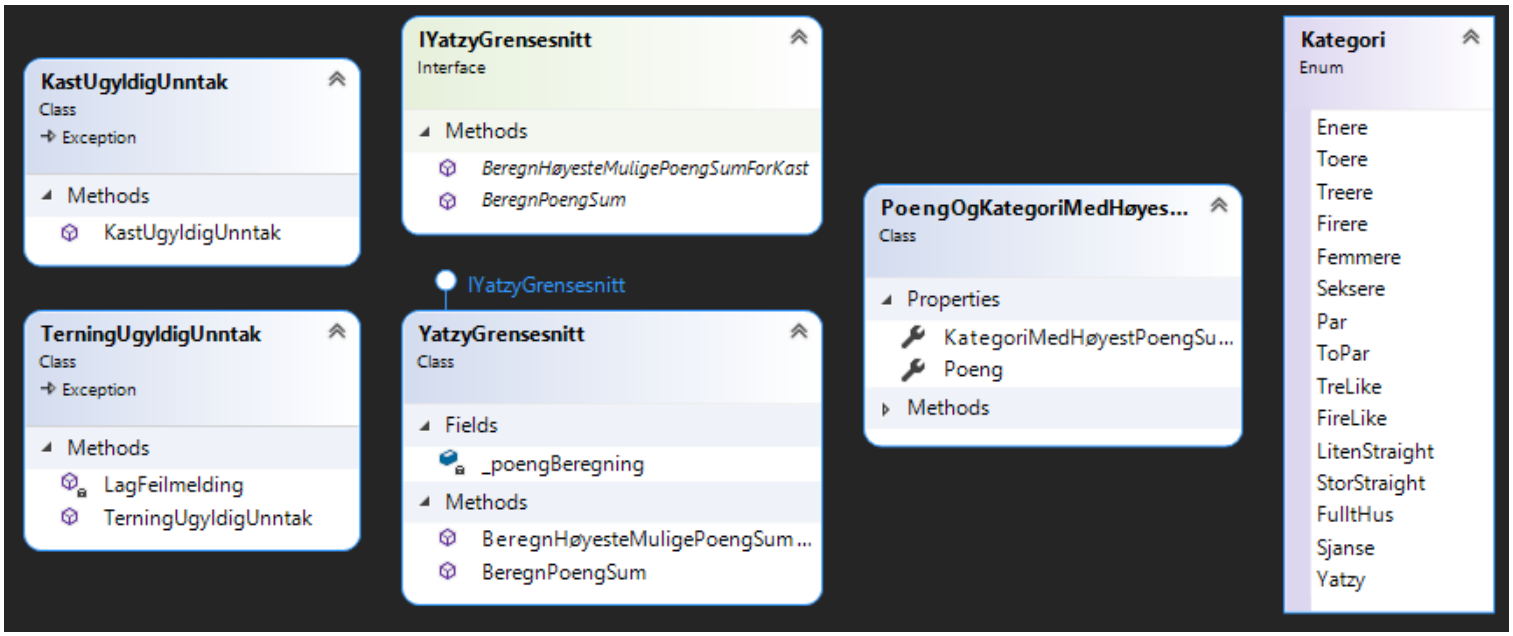
Dokumentasjon av intern oppbygning for kildekode

Det henvises generelt til kodedokumentasjon i selve kildekoden for detaljer på absolutt alle elementer.

Målgruppe

Utviklere som skal vedlikeholde og utvide koden videre.

Eksternt synlige klasser



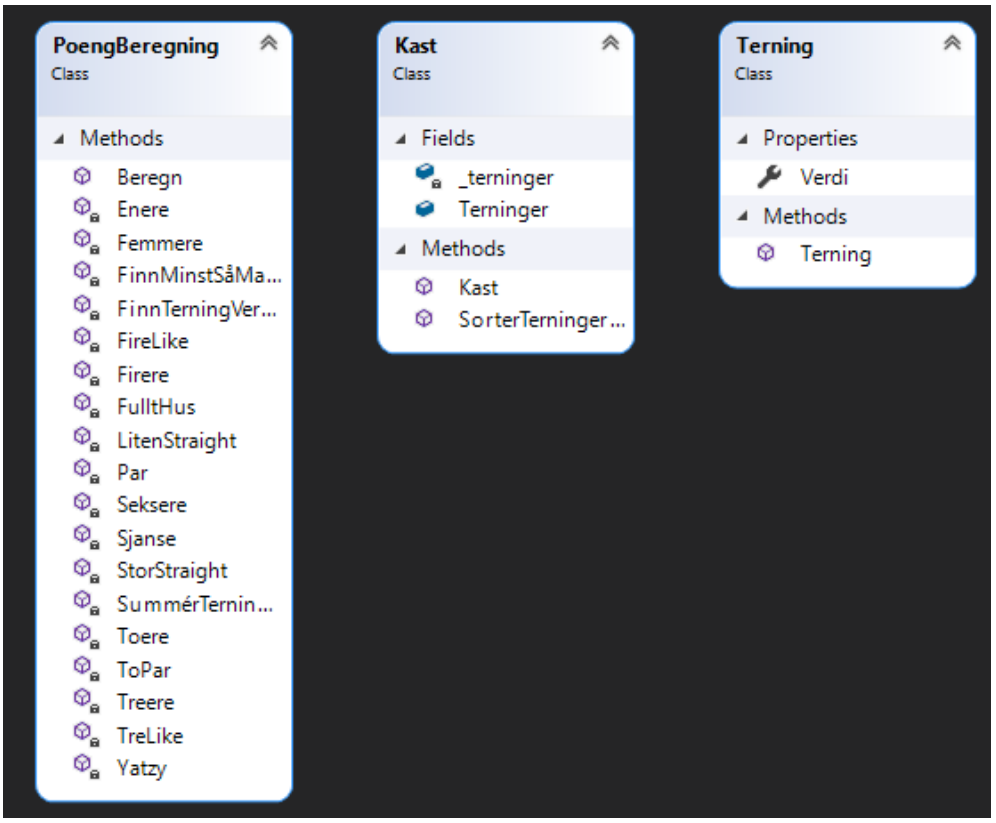
Inngangspunktet i koden er klassen *YatzyGrensesnitt,* som inneholder to metoder for poengberening. Den implementerer interfacet *IYatzyGrensesnitt* for tillate 3. parts klienter av programbiblioteket å enkelt kunne mocke ut avhengigheten.

Det er to unntak (Exceptions) som kan kastes ut av biblioteket, disse kastes ved ugydlige verdier i inputstrenger til metodene i *YatzyGrensesnitt*. *TerningUgyldigUnntak* kastes dersom det forsøkes å angi en terning i et kast med en ugyldig verdi. Feilmelding i unntaket angir hva som var ugyldig med terningverdien, og på hvilket sted i strengen for terningkast verdien var angitt. *KastUgyldigUnntak* kastes dersom en streng for et kast har ugyldig lengde.

PoengOgKategoriMedHøyestPoengSum benyttes som returverdi for metoden *YatzyGrensesnitt.BeregnHøyesteMuligePoengSumForKast*, siden denne skal returnere to verdier.

Kategori brukes for å angi hvilken av Yatzys mange kategorier et kast tilhører.

Interne klasser/domenemodell



Yatzy er et ganske enkelt spill, som her er modellert internt med klassene *Kast*, *Terning* og *Poengberegning*.

Kast er et domeneobjekt for å representere et kast i Yatzy. Implementanetasjon består hovedsaklig av å validere en streng som skal representere et kast med fem terninger.

Terning representerer en kastet terning i spillet, det er stort sett validering i konstruktør som er implementert.

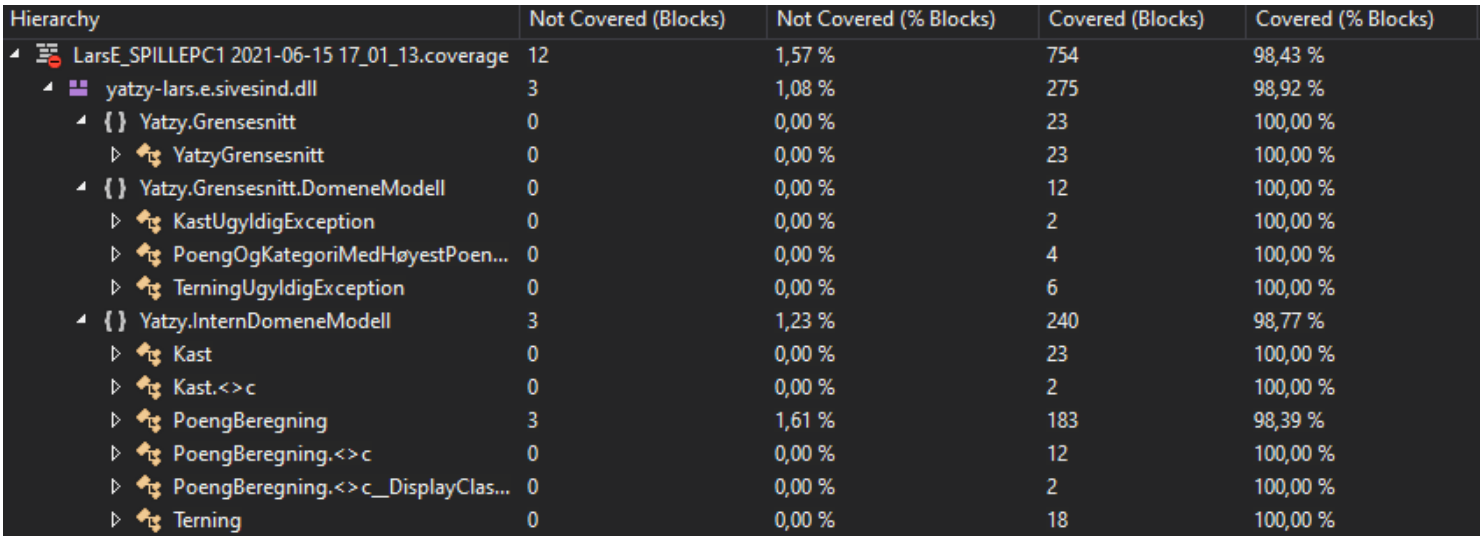
PoengBeregning inneholder mest logikk i koden, der ligger en bereningsmetode per kategori.

Automatiske tester

Automatiske tester finnes i testprosjektet Yatzy.AutomatiskeTester og er delt i integrasjonstester og enhetstester. Enhetstester tester kun én klasse, integrasjonstester berører flere. Det er gjort et utvalg av tester da det er mange kombinasjoner i Yatzy. En videreutvikling av programkoden må ivareta disse.

Dekningsgrad for automatiske tester

Dekningsgrad angir hvor mye av koden som berøres av tester. Den angir ikke alene hvor godt koden er testet, det avhenger også av gode sjekker (asserts) i teskoden. Det er likevel



Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
LarsE_SPILLEPC1 2021-06-15 17_01_13.coverage	12	1,57 %	754	98,43 %
yatzy-lars.e.sivesind.dll	3	1,08 %	275	98,92 %
Yatzy.Grensesnitt	0	0,00 %	23	100,00 %
YatzyGrensesnitt	0	0,00 %	23	100,00 %
Yatzy.Grensesnitt.DomeneModell	0	0,00 %	12	100,00 %
KastUgyldigException	0	0,00 %	2	100,00 %
PoengOgKategoriMedHøyestPoen...	0	0,00 %	4	100,00 %
TerningUgyldigException	0	0,00 %	6	100,00 %
Yatzy.InternDomeneModell	3	1,23 %	240	98,77 %
Kast	0	0,00 %	23	100,00 %
Kast.<>c	0	0,00 %	2	100,00 %
PoengBeregning	3	1,61 %	183	98,39 %
PoengBeregning.<>c	0	0,00 %	12	100,00 %
PoengBeregning.<>c__DisplayClas...	0	0,00 %	2	100,00 %
Terning	0	0,00 %	18	100,00 %

viktig å opprettholde dekningsgrad, for å være sikker på at ny kode testes.