

Oppgavesettet består av 2 sider

WESTERDALS OSLO ACT

PG3300 Software design

ARBEIDSKRAV

Leveranseinfo:

- Skal gjøres i grupper på 3-5 personer.
 - Leveres senest søndag 15. oktober 2017, klokka 23:59.
 - Leveres via it's learning. (Lever *før* fristen! Det kan ta tid å laste opp filer!)
 - Kode skal skrives i C#, og kompilere og kjøre i Microsoft Visual Studio Enterprise 2017.
 - UML diagrammer kan være scannede tegninger, pdf eller bildeformat (bildeformat inkluderer screenshot).
 - Tekstdokumentasjon skal være i pdf format.
-

Oppgave: UML og refactoring av SnakeMess

Se vedlagt solution/kodefil for SnakeMess.

Kodefila inneholder et (for de fleste) kjent spill. Koden kompilerer og kjører, men er ikke særlig lesbar, utvidbar, eller gjenbrukbar!

Denne oppgaven går ut på å refaktorere SnakeMess.cs til bedre kode.

"Bedre kode" inkluderer: (men er ikke nødvendigvis begrenset til)

- Tenk GRASP (få med uttrykkene i dokumentasjonen der det er relevant!)
- Se refactoringlista i forelesningsmaterialet.
- Bytt ut variabel-, klasse- og metodenavn med mer beskrivende varianter.
- Splitte koden i flere klasser og metoder. (Kanskje bytte ut en selvlagd klasse med et standardelementer fra .NET bibliotekene?)
- Implementere arv(?).

Diskuter muligheter innad i gruppen. Skriv i tekstdokumentasjonen fremgangsmåte (rekkefølge på punkter, hva dere prioriterer) og begrunn valgene deres (f.eks. hvorfor dere velger variant A fremfor B, der hvor dere kommer opp med flere muligheter). Dette skal bli en beskrivelse av prosessen som legges ved resten av innleveringen!

Merk: Refactoring av et dårlig strukturert prosjekt betyr av og til at koden vokser litt. I et prosjekt med så få kodelinjer som SnakeMess, betyr det at løsningen vokser prosentvis MYE. Dette er ok! :-)

Del a)

Sett opp et Implementation Class Diagram som viser hvordan dere tenker dere ferdig refactored versjon.

Husk at ved refactoring skal programmet oppføre seg likt for brukeren før og etter endringene! M.a.o. funksjonaliteten i programmet skal være den samme.

Del b)

Implementer det nye kodedesignet i Visual Studio. (Eller egentlig: sørg for at dere bygger og leverer en fungerende Visual Studio C# solution. Om dere vil jobbe i et annet verktøy underveis er det lov.)

Del c)

Ferdigstill tekstdokumentasjonen (pdf) der dere gjør rede for valgene og prosessen deres. Begrunn hvordan design guidelines har påvirket resultatet deres. Husk at dette er en del av vurderingen: Det er viktig at dere forklarer valgene deres.

Med andre ord: I dette tekstvedlegget, sørg for å få fram hvorfor akkurat deres løsning er så bra, dvs. vis (med ord, evt. figurer) at dere har tenkt, planlagt og resonert dere frem til gode valg underveis for løsningen deres!

Snake - slik det skal se ut når det kjører, både *før* og *etter* refactoring:

