

Øving 10: Prosjekt del 2

Læringsmål

Du skal lære hvordan å bygge videre på et program du har laget

Godkjenning

Du skal vise koden din, både på Github og at den kjører, til studentassistentene.

Deloppgaver

- a) Denne oppgaven bygger direkte på øving 9. Du kan velge om du gjør den i samme mappe og repository som øving 9 eller ikke. Hvis ikke må du kopiere over alt innholdet fra øving 9
- b) Skriv en enhetstest for klassen Sporsmaal, som tester i alle fall metodene sjekk_svar og korrekt_svar_tekst.
- c) Lag en klasse Spiller, som har egenskapene navn og poengsum. Poengsum skal ha 0 som default verdi.
- d) Lag en funksjon som lager ei liste med spillere (objekter av klassen Spiller). Funksjonen skal spørre brukeren om antall spillere. Funksjonen skal spørre brukeren om navn til hver spiller. Alle spillerne starter med 0 poeng
- e) Skriv om if `__name__ == "__main__"` blokken fra øving 9 på følgende vis. Før den starter spillet skal den bruke funksjonen fra forrige deloppgave til å lage ei liste med spillere. Etter å ha skrevet ut spørsmålet skal den spørre hver spiller med navn om hvilket svaralternativ vedkommende tror er korrekt. Deretter skal den skrive ut korrekt svaralternativ. Deretter skal den for hver spiller sjekke om spilleren gjettet riktig eller feil, skrive ut dette, og gi hver spiller som gjettet riktig ett poeng. Til slutt skal den sjekke hvilken spiller som får flest poeng, og skrive ut navn og poengsum til denne spilleren. Den obligatoriske delen krever ikke at du håndterer uavgjort
- f) **Frivillig:** Håndter uavgjort ved å skrive ut navnene til alle spillerne som har samme høyeste poengsum
- g) **Frivillig:** Gi spørsmålene en ekstra egenskap «kategori». Gi spillerne separate poeng for hver kategori. Dette kan du gjøre gjennom å bruke et dict i stedet for et tall som verdi for poengsum-egenskapen til spillerne, hvor dict-et har kategorien (en streng) som nøkkel og poengsummen som verdi. På slutten skal du sjekke en vinner for hver kategori.
- h) **Frivillig:** Lag en klasse «tallsvar». Denne skal ha de samme metodene som Sporsmaal klassen, men skal ikke ha ei liste med svaralternativer. Lag ei liste med vanlige spørsmål og tallsvar og kjør programmet ditt med denne lista. Er tallsvar korrekt skrevet skal dette virke. Dette er et eksempel på konseptet «polymorfisme» som går litt utover pensum i DAT120.

Eksempelutskrift

Her er et eksempel på utskrift, de to første spørsmålene:

Skriv inn navn på spillerne. Avslutt med tom linje.

Navnet til spiller 1: Erling

Navnet til spiller 2: Frida

Navnet til spiller 3: Georg

Navnet til spiller 4: Hanne

Navnet til spiller 5:

Den delen av en datamaskin som kjører programmet kalles?

Svaralternativer:

0: RAM

1: Harddisk

2: CPU

3: Sekundærlager

Velg et svaralternativ for spiller Erling: 0

Velg et svaralternativ for spiller Frida: 1

Velg et svaralternativ for spiller Georg: 2

Velg et svaralternativ for spiller Hanne: 3

Korrekt svar: CPU

Spiller Erling: Feil

Spiller Frida: Feil

Spiller Georg: Korrekt

Spiller Hanne: Feil

En ___ er nok minne til å lagre en bokstav eller et lite heltall.

Svaralternativer:

0: bit
1: byte
2: bryter
3: transistor

Velg et svaralternativ for spiller Erling: 0

Velg et svaralternativ for spiller Frida: 1

Velg et svaralternativ for spiller Georg: 2

Velg et svaralternativ for spiller Hanne: 3

Korrekt svar: byte

Spiller Erling: Feil

Spiller Frida: Korrekt

Spiller Georg: Feil

Spiller Hanne: Feil