

Inlämning: En natt på museet, 5 Yhp

Inlämningsdatum 12 dec 2021 av 23.59 **Poäng** 100

Lämnar in en länk till webbplats

Tillgänglig 6 dec 2021 kl 0:00–12 dec 2021 kl 23.59 7 dagar

Den här uppgiften låstes 12 dec 2021 kl 23.59.



Scenario

Du jobbar i det här scenariot som frilansande programmerare år 1979 och har fått ett uppdrag att göra ett modernt arbetsverktyg till ett konstmuseum, som personalen kan användas från deras splirrans ny datorterminal. Datorn har alla moderna program installerade man kan tänkas behöva - men personalen skulle behöva ett arbetsverktyg för att hålla koll på och inventera deras olika rum och konstverk! Du har pitchat en idé att programmet ska fungera precis som alla andra program via *command line arguments* - vilket personalen är väldigt positivt inställda till.

Inlämning & betygsättning:

Du ska utveckla och strukturera en konsolapplikation som uppföljer kravlistan för uppgiften. Genom att uppfylla kraven kommer du påvisa din förmåga att konstruera program på ett objektorienterat sätt (läranderesultat 4) samt strukturera din kod för god läsbarhet och att själv kunna analysera och förstå dess funktion (läranderesultat 5). Dessutom kommer du i denna uppgift att skriva enklare enhetstester för att garantera att programmet fungerar över tid (läranderesultat 6).

Påbörja arbetet genom att följa nedanstående länk till uppgiften på Github Classroom:

[Github Classroom: En natt på museet](https://classroom.github.com/a/7pg8rrRc) **[_ \(https://classroom.github.com/a/7pg8rrRc\)](https://classroom.github.com/a/7pg8rrRc)** (obs: denna gång innehåller detta repo viktig kod du behöver för att klara uppgiften)

När du är klar med uppgiften lämnar du in din lösning som en länk till det repo som skapats på Canvas. Efter rättning kommer du få feedback med korta kommentarer för varje rad i poängtabellen. För att ett krav ska vara poänggivande måste även de krav som angetts som *kriterium* vara uppfyllda! För betyget godkänt krävs minst **50 poäng** från poängtabellen, för betyget väl godkänt krävs **80 poäng**, samt att minst 20 av dessa är från de krav som är angivna med (vg) i poängtabellen. Erhålls åtminstone 25 poäng så erbjuds ett försök att komplettera din inlämning, annars ges betyget *underkänt*.

Självskattning och deadline

Läs nedanstående kravlista noggrant, när du påbörjar programmeringsuppgiften från Github Classroom så kommer du se att det finns en readme.md fil där du ska fylla i din självskattning. Självskattningen är frivillig men du bör alltid fylla i denna så att du vet att din uppgift rättas korrekt - läs **[Om inlämning och deadlines](#)** för mer information.

Kravlista

Följande kravlista är vad du ska programmera och lämna in, den fungerar som **poängtabell** för uppgiften:

NummerMax.PoängKriteriumBeskrivning			Läranderesultat
1	2p	Lösningen ska vara en vidareutveckling av det givna kodrepet från Github Classroom.	4, 5

Nummer	Max.Poäng	Kriterium	Beskrivning	Läranderesultat
2	3p	1	Programmet går att starta vid inlämning, och via den virtuella konsollen kan applikationen 'mu' användas.	4
3	2p	2	Applikationen 'mu' kan med ett lämpligt kommando lista alla rum samt konstverken som finns i rummet.	4
4	3p	2	Applikationen 'mu' kan med ett lämpligt kommando lista alla konstverken som finns i ett specifikt rum, rummet är angivet i kommandot.	4
5	2p	2	Applikationen 'mu' kan med ett lämpligt kommando lägga till ett nytt konstverk i ett rum.	4
6	3p	2	Applikationen 'mu' kan med ett lämpligt kommando radera ett specifikt konstverk från ett rum.	4
7	2p	2	Applikationen 'mu' kan med ett lämpligt kommando lägga till ett helt nytt rum i museet.	4
8	3p	2	Applikationen 'mu' kan med ett lämpligt kommando radera ett specifikt rum i museet.	4
9	4p	1	Varje konstverk representeras i kod via en titel , beskrivning och en upphovsmakare	4, 5
10	3p	1	Endast 3 konstverk kan finnas i varje rum.	4, 5
11	3p	1	Ett rum kan inte raderas om det finns konstverk i rummet.	4, 5
12	3p	2	Med applikationen 'mu' kan man skapa ett helt nytt museum/byggnad.	4, 5
13	4p	2 & 12	Via applikationen 'mu' kan man med kommandot 'select' välja vilken vald byggnad övriga kommandon utförs på, man behöver inte alltid ange vilket museum som används.	4, 5
14	3p	1	Koden är strukturerad på ett sådant sätt att Console klassen är väl separerad från programmets modellklasser (i.e. "konstverk" och "rum") - så att dessa går att återanvända i framtida applikationer	4, 5
15	5p	1	Repot ska lämnas in med minst 3 enhetstester implementerade.	6

Nummer	Max.Poäng	Kriterium	Beskrivning	Läranderesultat
16	5p	12	Enhetstesterna bevisar att flera museum/byggnader kan finnas med olika uppställningar av konstverk och rum.	4, 5, 6
17	10p	10 & 11	Enhetstesterna bevisar att reglerna i krav 10 & 11 alltid efterföljs.	4, 5, 6
18 (vg)	2p		Lösningen ska förutom kod innehålla en fil med namnet "reflections" i formatet <i>md</i> , <i>txt</i> eller <i>pdf</i> . <i>reflections</i> -filen ska under rubriken "Kommandon" innehålla	4, 5, 6
19 (vg)	12p	1	en kortfattad beskrivning av vilka <i>verb</i> som är tillgängliga i din applikation och vad som anges som ytterligare argument för varje verb - svara gärna på varför dessa är lämpliga argument till verbet. <i>reflections</i> -filen ska under rubriken	4, 5
20 (vg)	12p	14	"Seperation" beskriva hur du har seperarat användningen av <i>System.Console</i> från dina modell-klasser. (1-4 paragrafer) <i>reflections</i> -filen ska under rubriken "Testning"	4, 5
21 (vg)	14p	15	beskriva de tester du skapat och deras syfte i lösningen samt en motivering varför just dessa tester är lämpliga i denna uppgift (2-3 paragrafer)	6

Tips

Anledningen till att vi gör denna uppgift med command line arguments är för att träna på att hantera **1 begäran i taget** - vilket är likt hur webbservrar fungerar, och förbereder dig därför redan nu lite inför nästa kurs. Du kommer dock märka att detta är ganska lätt i.o.m. det färdighanterade *simulator*-projektet som skickas med när du börjar uppgiften.

Tänk på varje givet *verb* och *options* listan likt du tidigare jobbat med *Console.ReadLine* - d.v.s som användarinmatning, i denna uppgift bör du inte använda *ReadLine* förutom möjligtvis att mata in ytterligare information om rum och konstverk än vad som är möjligt med *command line arguments*.

Tänk på att tillämpa vad du lärt dig om objektorienterad programmering redan nu när du granskar kravlistan. Typiskt behöver det finnas ett sätt att koppla "rum" och "konstverk" till varandra - och även byggnader/hotell om du ämnar lösa alla krav.

Lycka till!