

EN NATT PÅ MUSEET

Inlämningsuppgift Praktiskt projektarbete och sammanfattning



DEN 4 DECEMBER 2020

WIVIANNE GRAPENHOL
grwi20ql@student.ju.se

Innehåll

Navigering.....	2
Separation.....	2
Testning.....	3
Förflyttningar	3
PossibleNavigation	3
ImpossibleNavigation	3
Motivering.....	3
Konstverk	3
AddCollectionOfArtToMuseum	3
AddArtToRoom.....	3
Motivering.....	3
Reflektion.....	3

Navigering

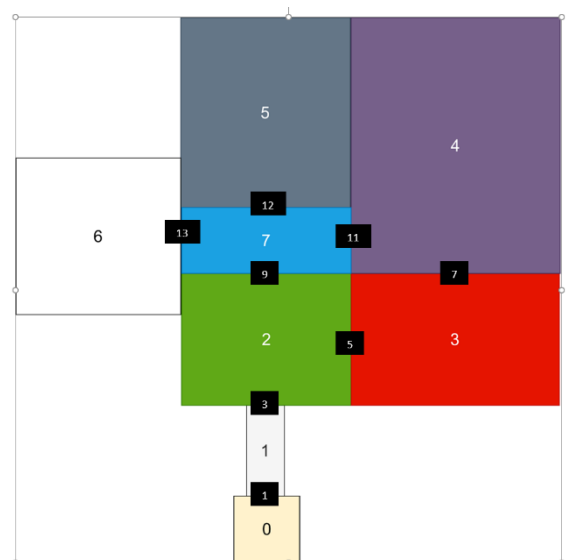
Jag har valt att numrera rum och dörrar för att kunna räkna ut vilka förflyttningar som är möjliga, se bild här bredvid. En dörr som förbinder två rum får summan av rummens roomId som doorId.

Det roomId på rummet användaren befinner sig i, valt doorId samt instansen av Museum skickas som parametrar till metoden Museum.PossibleTransfer som räknar ut vilket rum som kommer näst.

Ex:

$11 - 4 = 7$ (doorId – roomId = nästa roomId) Sedan söker metoden upp rummet i listan museumRooms sätter currentLocation till false i rum 4 och true i rum 7.

Nackdelen med min lösning är att det kan bli klurigt att få till unika doorId till dörrarna beroende på rumsuppsättning i annat museum. Man behöver ju dock inte numrera rummen i nummerordning som jag gjort här och på det viset få en större spridning av resultaten av uträkningarna och på så sätt minimera risken för identiska doorId.



Separation

Genom att jag har implementerat klassen Display har jag lyckats att hålla all output och input skild från resterande klasser. När programmet startar i Program.Main ”byggs” mitt specifika museum som sedan skickas med som parameter till metoderna Display.ShowIntro eller Display.ShowRoom. på så vis är det bara instansen av Museum med tillhörande instanser av Room, Door och Art som interagerar med Console klassen. Den input som behöver ske, val av dörr, skickas vidare som parameter till metoden Museum.PossibleTransfer.

Och som jag förstår det är därför logiken och användargränssnittet väl avgränsade från varandra.

Testning

Jag har skapat fyra enhetstester. De två första testar logiken i förflyttningar i mitt museum och de två efterföljande testar att lägga till konstverk i museum och rum.

Förflyttningar

PossibleNavigation

Här arrangerar jag ett scenario med en tillåten förflyttning och antar att `currentLocation` kommer vara `true` i nästa rum, vilket innebär att användaren har förflyttats till det rummet och därför var det en acceptabel övergång mellan rummen.

ImpossibleNavigation

Här arrangerar jag samma utgångsläge som ovan men försöker gå genom en dörr som inte finns i rummet som är `currentLocation`, därför antar jag att `currentLocation` kommer vara `false` i båda rummen som är sammankopplade med den dörren.

Motivering

Eftersom jag valt att matematiskt räkna ut förflyttningar och hålla reda på användarens position med `bool` variabeln `currentLocation` anser jag att dessa två tester är väl valda för att testa denna logik. Dessutom så skapar jag i båda testerna en instans av `Museum`, tre instanser av `Room` (lägger till dessa i `Museum.museumRooms`) som i sin tur skapar ett givet antal instanser av `Door` (lägger till dessa i `room.doors`) så testas ju på så sätt även att alla dessa funktioner implementeras korrekt.

Konstverk

AddCollectionOfArtToMuseum

Här skapar jag en instans av `Museum`, förväntar mig att 11 konstverk skall finnas i listan `Museum.museumArts`. Anropar metoden `Museum.AddArtCollection` som för över alla 11 konstverk som är listade och sparade i en array i `Art.CollectionOfArt`. Sedan kollar jag det faktiska antalet i `Museum.museumArts`. För att slutligen jämföra förväntat antal med antalet objekt i `Museum.museumArts`.

AddArtToRoom

I detta test arrangerar jag först ett utgångsläge med en instans av ett `Museum`, lägger till museets konstsamling, skapar en instans av rum, lägger till det i `Museum.museumRooms`. Antar att rummets lista `room.arts` kommer innehålla 3 objekt. Jag anropar metoden `Room.AddArtToRoom` med `Museum.museumArts` samt index för konstverken som parametrar. Sedan kollar jag det faktiska antalet i `room.arts`. För att slutligen jämföra förväntat antal med antalet objekt i `room.arts`.

Motivering

Jag tycker detta är väl valde tester för att det testar att det går att alternera både hela museets konstsamling och flytta runt konstverk mellan rum på ett relativt enkelt sätt. Även i dessa tester inkluderas implementeringen av nytt `Museum` objekt och att en instans av `Room` skapas och läggs till i museet.

Reflektion

Jag kan inte låta bli att avsluta med en liten sammanfattande reflektion över min lösning och även av min egen arbetsinsats.

Jag har funderat på hur det skulle vara att lägga till alternativa museum och flera uppsättningar av konstsamlingar. Att man skulle kunna skapa flera unika instanser av `Museum` i olika metoder i en klass som skulle kunna heta `CollectionOfMuseum`. På samma sätt skulle man kunna lösa att byta ut Museets konstsamling; om `CollectionOfArt` var en egen klass och olika instanser av klassen innehöll olika `Art` objekt inom olika metoder. Då tror jag att man enkelt kunde anropa dessa olika metoder vid behov. Jag tror att min struktur och logik enkelt går att applicera på andra museum och konstuppsättningar / museiföremål.

Jag är väldigt nöjd med min lösning utifrån de personliga förutsättningar jag haft denna vecka. Det finns säkert delar av kod som hade kunnat förbättras och förenklas. Men jag anser att min lösning uppfyller samtliga krav förutom nr 14 och det är väldigt mycket bättre än vad jag vågade hoppas på i början av veckan.