

# Inlämning: Multifabriken, 3 Yhp

**Inlämningsdatum** 19 nov 2021 av 23.59      **Poäng** 100

**Lämnar in** en länk till webbplats

**Tillgänglig** 8 nov 2021 kl 0:00–19 nov 2021 kl 23.59 12 dagar

Den här uppgiften låstes 19 nov 2021 kl 23.59.



## Scenario

Du ska utveckla en konsolapplikation för bolaget **Multifabriken AB**. De vill skapa en konsolapplikation för sina kunder där man kan välja att beställa nya produkter från deras fabrik. Fabriken kan producera bilar, godis, snören eller tofu. Via inmatning av beställning ska nya produkter skapas av den valda typen och samtliga produkter man beställt ska kunna listas av användaren.

## Inlämning & betygsättning

Du ska utveckla och strukturera en konsolapplikation som uppföljer nedanstående

kravlista *från* Multifabriken. Dit mål är att uppvisa färdighet i att lösa följande koduppgift på ett objektorienterat sätt (läranderesultat 4) samt att strukturera din kod för god läsbarhet och tydlighet och analysera dess funktion (läranderesultat 5).

Läs igenom nedanstående kravlista/poängtabell och några korta tips på uppgiften nedanför, starta sedan arbetet genom att acceptera uppgiften på Github Classroom:

### Github Classroom: Multifabriken [\(https://classroom.github.com/a/pb4J\\_vmt\)](https://classroom.github.com/a/pb4J_vmt)

När du är klar med uppgiften lämnar du in din lösning som en länk till det repo som skapats på Canvas. Efter rättnings 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 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.

### Reflektion och kodgranskning

Efter denna uppgift kommer vi i nästa modul genomföra ett seminarium i grupp där du granskar några av dina klasskamraters lösningar på samma uppgift - tänk då på att den reflektion och självskattning du gör i denna uppgift även kommer vara synlig för andra klasskamrater!

Alla dela med sig av det upplevda resultatet av ett projekt är ett vanligt och viktigt inslag i yrkeslivet - och vi kommer i seminariumet att träna på detta, oftast är förmågan att utvärdera och beskriva både programkod och projektresultat viktigare än att faktiskt ha skrivit den bästa möjliga lösningen.

## Kravlista

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

Nummer	Max.Poäng	Kriterium	Beskrivning	Läranderesultat
1	2p		Lösningen ska bestå av en konsolapplikation skriven i C# som är körbart med .NET Core	4, 5
2	10p	1	När man startar programmet ska man mötas av en meny samt en kort introduktion till programmet	4, 5
3	6p	2	Man ska kunna göra ett val i menyn med vilken typ av produkt man vill beställa, om man vill lista alla produkter man beställt, eller om man vill avsluta programmet	4, 5
4	5p	2	Produkterna man kan välja mellan är <b>bilar</b> , <b>godis</b> , <b>snöre</b> samt <b>tofu</b>	4, 5
5	5p		När man väljer att skapa ett <i>snöre</i> så omges man att mata in "färg" samt "längd"	4, 5
6	5p		När man väljer att skapa en <i>bil</i> ska man omges att mata in ett "registreringsnummer", en "färg", och sedan ett "bilmärke"	4, 5
7	5p		När man väljer att skapa en <i>godis</i> omges man att mata in "smak" följt av "antal"	4, 5
8	5p		När man väljer att skapa <i>tofu</i> omges man att mata in "volym" i liter samt en "kryddning"	4, 5
9	10p	2	Om man väljer att lista alla produkter ska alla produkter man beställt hittills skrivas ut med den informationen man angott	4, 5
10	2p	2	Om man väljer att avsluta programmet ska programmet helt enkelt sluta köras	4, 5
11	5p	1	Lösningen ska bestå av en "Program.cs" fil, en ".csproj" fil samt en fil för varje extra klass du skapar	4, 5
12 (vg)	10p		Lösningen ska förutom kod innehålla en fil med namnet "reflections" i formatet <i>md</i> , <i>txt</i> eller <i>pdf</i>	4, 5
13 (vg)	10p	12	<i>reflections</i> -filen ska en ska innehålla en kort beskrivning av kodens funktioner och struktur	4, 5
14 (vg)	20p	12	Filen <i>reflections</i> ska också innehålla en egen analys och reflektion över hur du strukturerat din kod genom användandet av metoder och klasser.	4, 5

## Tips

Tänk på att utgå från formuleringarna i det ovanstående *scenariot* samt kravlistan när du väljer vilka klasser du ska skapa! Efter att information har matas in av användaren så borde exempelvis ett objekt av korrekt klass skapas och läggas i en lista - det bör alltså finnas en klass per typ av produkt, samt en lista för varje klass. Ingen information behöver sparas mellan körningar av programmet.

När du skriver din analys och reflektion, motivera vilka klasser du skapat och hur det hjälper programmet att bli överskådligt för andra programmerare än dig själv; om dom skulle jobba vidare i samma projekt. Ge gärna ett exempel på hur du gjort tidigare, eller ett annat tankesätt du känner till, och jämför vad som var som var bättre och sämre med denna struktur.

### Lycka till!

Så här kan exempelvis programmet se ut och fungera när det är klart

Introduktion:

Här kan du beställa olika produkter från vår fabrik.  
Välj alternativ genom att skriva en siffra och tryck sedan på enterknappen.  
Du kan inte ta bort produkter efter att du beställt dem, så var försiktig!

1. Beställ bil.
2. Beställ godis.
3. Beställ snöre.
4. Beställ tofu.