

DTP-D10

Inlämningsuppgift 1

Villkor

Det här är en uppgift som är klok att genomföra i tid, men för att inte krångla till det så gäller de vanliga villkoren för inlämningsuppgifter:

start:	24 mars 2023, 9.00
sluttid:	c:a våren 2025, ungefär (<i>tänk dock på att prestera resultat så fort som möjligt!</i>)
inlämning:	ni lämnar in en githublänk som leder till ett publikt (!!) github-repo
godkänt:	ni genererar 10 pushes
väl godkänt:	ni genererar 15 pushes och löser minst tre (3) Exception-problem med try-catch (Uppgift 3)

D15-inlämning – refaktorering, avlusning och git

Om ni tycker att uppgifterna liknar måndagens och tisdagens uppgifter, så är det för att den liknar måndagens och tisdagens uppgifter. Om ni blir osäkra, så gå igenom dessa: övningarna, genomgångarna och slideshowerna. Läs igenom alla uppgifter innan ni startar! Om ni refaktorerar och testkör samtidigt, så sparar ni tid!

Uppgift 1: kлона, forka eller kopiera koden i [MJU23v D10 inl sveng](#), se till att du har ett eget **publikt (!!)** repo som heter detsamma, så att jag kan följa ert tänkande. Läs koden, och testkör.

Notera eventuella fel! Lägg in dem som `// FIXME`-kommentarer, åtgärda dem bara om du kan fixa dem med en enda rad.

Jag har inte gjort någon help-funktion, **om** ni vill lägga in en helpfunktion så pusha den och denna push räknas in bland pusharna som är underlag för godkännandet! Om inte, simma lugnt!

Uppgift 2 – FAS2-lik: ämnet är att **refaktorer**a koden bland annat genom att ta bort koddubletter, men i stället för att säga: "gör det, och sedan gör det", så är det bättre om ni använder ert eget omdöme, d.v.s.

- läser igenom koden,
- fattar beslut efter eget huvud och egen åsikt, och
- gör åtgärderna i den ordning ni finner mest lämpliga själva.

Vad ni skall uppnå är:

1. enbokstavsvariabler skall döpas om, **i** som for-loops-index behåller vi – det är en vedertagen konvention – gör **en enda stage/commit/push** för alla enbokstavsvariabler¹,
2. koddubletter skall bort, tag bort dem en och en, *kompilera/testa*, och efter var och en borttagning gör en *stage/commit/push*, d.v.s. om du hypotetiskt (hint hint!) refaktorerar translate-grenen i if-elseif-else-kedjan så gör du en *stage/commit/push* för exakt denna refaktorering,

¹ om du råkar göra fler än en så får du inte underkänt, men **läraren räknar** bara en push här!

3. när ni hittar fel i kompilerings/test-sekvenserna skriver ni ner dem som `// FIXME-` kommentarer, åtgärda dem bara om du kan fixa dem med en enda rad! Gör ingen git `stage/commit/push` för enbart kommentering!
4. För de olika funktionerna föreslås t.ex. följande tester:

list	list	load	load <i>file</i>	load <i>file</i>
starta	starta	starta	starta	starta
> list	> load	> load	> load <code>computing.lis</code>	> load <code>finns.ej</code>
> quit	> list	> list	> list	> list
	> quit	> quit	> quit	> quit

och

new	new	new <i>swe eng</i>	new <i>swe eng</i>
starta	starta	starta	starta
> new	> load	> load	> load
> list	> list	> list	> list
> quit	> new	> new <code>sol sun</code>	> new <code>sol</code>
	> list	> list	> list
	> quit	> quit	> quit

och

delete	delete	delete <i>swe eng</i>	delete <i>swe eng</i>
starta	starta	starta	starta
> delete	> load	> load	> load
> quit	> list	> list	> list
	> delete	> delete <code>ost</code>	> delete <code>ost cheese</code>
	> list	> list	> list
	> quit	> quit	> quit

och

```
delete swe eng
starta
> load
> list
> delete ost cleese
> list
> quit
```

Uppgift 3: Fixa tre eller flera `// FIXME-` kommentarer som genererar exceptions. När du gör en `try-catch` för att skydda från att programmet krashar när den försöker läsa en fil som inte finns, så skall du alltså fånga upp `FileNotFoundException`, inte bara `Exception`!