

Tölvunarfræði 1

Heimadæmi 9

Í tíundu viku námskeiðsins (26-30. október.) verður fjallað um sýnidæmi um hvernig hægt er að skrifa stærri forrit í Java. Sýnidæmið er Monte Carlo hermun á seytli (*percolation*). Þetta er efni úr [kafla 2.4](#) í kennslubókinni.

Hér fyrir neðan eru heimadæmi 9, sem eru að mestu um endurkvæmni úr kafla 2.3 í kennslubókinni. Þið eigið að skila þessum dæmum í Gradescope fyrir kl. 14:00 föstudaginn 30. október. Ekki verður tekið við lausnum sem koma eftir það.

Heimadæmi

Hvert dæmi gefur 2 punkta.

0.1 Áður en þið byrjið að forrita dæmin skuluð þið búa til klasana í dæmunum. Þeir eru:

`Fibonacci`, `Lesafylki`, `Hornafoll`, `Leit` og `Reglustika`.

0.2 Fyrir hvert dæmi skuluð þið

skilgreina mynsturfar og skrifa tómmt fall með return setningu þar sem það á við.

0.3 Þýðið alla klasana í IntelliJ

0.4 Ef klasarnir eru villulausir úr þýðingu þá skuluð þið hlaða þeim inn í Gradescope. Ef það kemur villa úr Gradescope er það vegna þess að klasanöfn eða mynsturför eru röng.

0.5 Hefjist nú handa við að forrita dæmin.

1. Í heimadæmi 2 forrituðu þið einfalt main-fall til að prenta fibonacci orð. Núna eigið þið að forrita klasann `Fibonacci` sem inniheldur endurkvæma fallið `fibonacciOrd`. Fallið tekur inn (argument) tvo bókstafi (notið `char` tagið) og heiltölu `n`. Fallið skilar (return) `n`-ta fibonacci orðinu sem streng. Skilgreina mynsturfar (signature) á `fibonacciOrd` og skjalið (document) með JavaDoc.

Þið getið notað fallið `Character.toString` til að búa til streng úr `char` breytu.

Skrifaðu main-fall í `Fibonacci` til að prófa `fibonacciOrd` fallið með `n` sem 0,1,2, 3 og 4. Í main-fallinu lesið bókstafina af staðalinntaki með því að nota `StdIn.readChar()`. Skrifðu niðurstöðuna úr `fibonacciOrd` á staðalúttak.

2. [Prófdæmi 2019] Skrifðu klasann `Lesafylki` sem hefur fallið `lesa`. Byrjið á að skilgreina mynsturfar á `lesa` og skjalið með JavaDoc.

Fallið `les` inn af staðalinntaki tvívítt heiltölufylki (notið `StdArrayIO`) sem hefur jafnmargar línur og dálka, reiknar út hornasummu fylkisins frá efra hægra horni og til

neðra vinstra horns og skilar summunni sem heiltölu. Ef innlesna fylkið hefur ekki jafnmargar línur og dálka þá skilar fallið 0.

Skrifið main-fall til að prófa fallið **lesa**.

3. Skriðið klasann **Hornafo11** með fallinu **sin** sem tekur inn kommutölu og heiltölu í þessari röð. Kommatalan er x í formúlunni hér á eftir og heiltalan er n í formúlunni.

Byrjið á að skilgreina mynsturfar (signature) **sin** og skjalið með JavaDoc.

Fallið útfærir á endurkvæman hátt sinus fallið sem er reiknað út svona:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

$$\sum_{n=1}^{\infty} (-1)^{(n-1)} \frac{x^{(2n-1)}}{(2n-1)!}$$

Í útfærslunni skuluð þið nota [Factorial](#) klasann úr bókinni og þið þurfið ekki að hlaða honum inn í Gradescope.

Skrifið main-fall í **Hornafo11** klasann sem prófar fallið **sin** með x gildi af eigin vali og $n=1$ og 9.

4. Skriðið endurkvæmt leitarfall **finna** í klasanum **Leit** sem leitar að streng, **nafn**, í raðaðri símaskrá (**simaSkra**) í sætum **fra** til og með **til**. Fallið skilar sætinu sem **nafn** er í en annars -1.

Byrjið á að skilgreina mynsturfarið svona og skrifa JavaDoc lýsingu.

```
public static int finna(String[] simaSkra, String nafn, int fra, int til)
```

Stökin í **simaSkra** fylkinu gætu t.d. verið "Ari" "Bjarni" "Hlín" "Nanna" "Rögnvaldur" **finna** (**simaSkra**, "Nanna", 0, 4) skilar 3

Pseudókóðinn fyrir **finna** aðferðina er eftirfarandi:

```
public static int finna(String[] simaSkra, String nafn, int fra, int til)
```

```
    ef til < fra þá  
        skila svarinu -1
```

```
    midja = (fra+til)/2
```

```
    if nafn er sama og miðjustakið í simaSkra þá  
        skila svarinu midja
```

```
    annars ef nafn < miðjustakið í simaSkra þá  
        skila svarinu finna (simaSkra, nafn, frá, midja-1)
```

```
    annars  
        skila svarinu finna (simaSkra, nafn, midja+1, til)
```

Þið getið notað aðferðina `compareTo` í klasanum `String` til að bera saman 2 strengi.
<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/String.html>

Forritið `main` fall sem prófar fallið `finna`. Veljið ykkur aðferð að eigin vali til að lesa inn eða útbúa símaskrána og nafnið sem á að finna.

5. Skriðið klasann `Reglustika` með **endurkvæmu** aðferðinni `stika` og `main`-falli. Aðferðin `stika` skrifar út á staðalúttak streng sem sýnir hæð strika í `reglustika`. Aðferðin `stika` tekur inn heiltölu `n` sem gefur hæstu tölu á `reglustikunni`. Aðferðin skilar engu. Ef kallað er á `stika (4)` þá prentast út eftirfarandi á staðalúttak: 121312141213121. Ef kallað er á `stika (3)` þá prentast út eftirfarandi á staðalúttak: 1213121. Ef kallað er á `stika (0)` á ekkert að prentast.

Í `main` fallinu skuluð þið prófa að kalla á `stika` með `n=0`, `n=1` og `n` eitthvert gildi að eigin vali.