

Funktionell Programmering- Övningsuppgifter Kotlin

Innehåll

Utbildning <i>JAVA22, JAVAD22</i>	1
Övningsuppgifter	2
Minilabar (Efter ”Vi börjar med Kotlin”-filmen)	2
Uppgift 1a– Videobandspelaren	2
Uppgift 1b– Videobandspelaren, forts	3
Uppgift 2a– For-loopar	3
Uppgift 2b– For-loopar	3
Uppgift 2c– For-loopar	3
Uppgift 3a - Gissa talet	3
Uppgift 3b - Gissa talet, med exceptions	3
Uppgift 4a– Klassen Djur	4
Uppgift 4b– Listor med Djur	4
Uppgift 4c– Set med Djur	4
Uppgift 4d – Map med Djur	4
Uppgift 4e – When	4
Uppgift 5a – Extension functions	4
Uppgift 5b – Extension functions	5
Uppgift 5c – Extension functions	5
Uppgift 6a – Lambdas	5
Uppgift 6b – Lambdas	5
Uppgift 6c – Lambdas	5
Uppgift 6d – Lambdas	5
Uppgift 6e – Lambdas	5
Uppgift 6f – Lambdas	5
Uppgift 6g – Lambdas	5
Uppgift 6h – Lambdas	6
Uppgift 7a – Rekursion, ränta på ränta	6
Uppgift 7b – Rekursion, summering	6
Uppgift 7c – Rekursion, max-värde	6
Uppgift 7d – Rekursion, max-värde	6
Uppgift 7e – Rekursion, balanserade parenteser	6
Uppgift 7f – Rekursion, fibonacci	6
Uppgift 8 – Högre Ordningens Funktioner – Text Distortion	7
Uppgift 9 – Högre Ordningens Funktioner – Generiska funktioner	7
Uppgift 10a – Mera lambdas, zip	7
Uppgift 10b – Mera lambdas, zip	7
Uppgift 11a – Mera lambdas, gruppering av mappar	7
Uppgift 11b – Mera lambdas, traversering av mappar	7
Uppgift 11c – Mera lambdas, filtrering av mappar	8
Uppgift 11d – Mera lambdas, mera mappar	8
Uppgift 11e – Mera lambdas, fold	8
Advent of Code 1a+1b (2020)	8
Advent of Code 2a + 2b (2020)	8

NACKADEMIN

Advent of Code 3a (+3b).....	9
Advent of Code 5a+5b (2020).....	9
Advent of Code 6a (2020).....	9
Advent of Code 8a+8b (2020).....	9
Advent of Code 10a+10b (2020).....	9
För er som vill ha mer	9
Facit.....	10

Övningsuppgifter

Minilabar (Efter "Vi börjar med Kotlin"-filmen)

Skapa upp ett Kotlin-projekt med ett Worksheet-dokument.

Se till att du kan skriva ut "Hello World", både som ren utskrift och från en funktion.

Skriv en funktion som returnerar valfri int, använd =

Skriv samma funktion, fast med krullparenteser, notera skillnaden.

Skriv ut en variabel, inifrån en sträng, mha \$variabelnamn.

Skapa en area-uträknings-funktion som tar 2 int som tar höjd och bredd som inparametrar och räknar ut och returnerar arean. Gör även en omkrets-uträkningsfunktion som returnerar omkretsen för en rektangel. Skriv bägge dessa funktioner på både kort och lång-notation (med = och med { })

Skapa variabler som betecknar höjd och bredd för en rektangel. Testa att göra olika variabler som är både implicit och explicit typade, och som både är var och val. Använd variablerna när ni anropar era funktioner för area och omkrets.

Explicit typ är t.ex: var i : Int = 5

Uppgift 1a- Videobandspelaren

Förr när man fortfarande spelade in film på videoband var det ibland svårt att veta om ett kommande tv-program skulle få plats på bandet.

Skriv ett program som hjälper oss räkna ut om ett band kommer att räcka eller inte.

Programmet ska ha följande 6 värden från start:

- videobandets längd (i minuter),
- hittills använd tid av bandet,
- den tidpunkt då tv-programmet börjar (ange först antal timmar, sedan minuter)
- den tidpunkt då det slutar (ange först antal timmar, sedan minuter).

Vi kan utgå från att programmen alltid börjar och slutar på samma dag.

Gör flera små, separata metoder, en för varje typ av uträkning. Det är helt ok att hitta din gamla Java-lösning för denna uppgift och titta på den när du skriver din Kotlin-kod.

NACKADEMIN

Uppgift 1b- Videobandspelaren, forts

Betrakta din lösning av uppgift 1a, liknar den kod du brukade skriva i Java? Låt oss nu närma oss en mer funktionell stil att programmera.

- Om du har funktioner som är skrivna med krullparenteser, kan du skriva om dem med kortnotation (=)?
- Använder du **var** där du hade kunnat ha **val**? Använd alltid **val** där det funkar!
- Står dina funktioner uppradade ovanför main? Släng in dem i main! Functions are first class citizens.
- Finns det saker som hade kunnat göras på en rad, men som nu görs på flera rader? (Det är inte alltid rätt med allt på en rad, men testa att slå ihop funktionaliteten där det är rimligt och se hur det känns.)

Uppgift 2a- For-loopar

Skriv en funktion som listar alla jämna tal mellan 1 och 20, baklänges. Använd modulu

Uppgift 2b- For-loopar

Skriv en funktion som listar godtycklig del av godtycklig multiplikationstabell

Inparametrar ska vara:

- En int x som talar om vilken multiplikationstabell som ska listas
- Det lägsta tal som x ska multipliceras med
- Det högsta tal som x ska multipliceras med

Uppgift 2c- For-loopar

Skriv en funktion som räknar ut om ett givet tal är ett primtal. Returnera true eller false.

Uppgift 3a - Gissa talet

Gör ett litet program där du väljer ut ett random värde mellan 1 och 10.

Låt användaren gissa talet

Skriv ut "Högre" eller "Lägre" tills användaren har gissat rätt.

Fråga sen användaren om hen vill spela igen

Uppgift 3b - Gissa talet, med exceptions

När ni anropar `readLine()!!.toInt()`, vid inläsning av tal, finns risk för att `NumberFormatException` kommer att kastas. Detta sker om det som skrivs in inte är en siffra.

Lägg till ett try-catch-block som fångar `NumberFormatException` och säger åt användaren att skriva in en siffra när detta fel har inträffat

Kolla att ditt spel funkar fint och att användaren får sitt felmeddelande och kan fortsätta spela även om `NumberFormatException` har inträffat

NACKADEMIN

Uppgift 4a– Klassen Djur

Skapa en klass ”Djur” som har en

- String ”Sort”
- Int ”antalBen”
- Skapa getters och setters för alla variabler

Gör metoder för att skriva ut

- Djurets sort
- Antal ben
- All info om ett djur

I ditt huvudprogram

- Skapa upp ett antal djur
- Skriv ut vilka djur du har och antal ben för varje djur

Uppgift 4b– Listor med Djur

Modifiera ditt huvudprogram så att du:

- Skapa upp ett antal djur **och lägg i en lista**
- Skriv ut vilka djur du har och antal ben för varje djur **genom att iterera över listan**

Uppgift 4c– Set med Djur

I ditt djur-program, skapa upp två olika Set.

- Ett Set som innehåller alla djur, flera gånger om
- Ett Set som innehåller ett ex av varje djur, men i annan ordning än i det första Set:et
- Skriv ut alla djur i Set1
- Notera att varje djur bara finns med en gång i din utskrift
- Skriv ut alla djur i Set2
- Skriv ut resultatet av Set1.equals(Set2)
- Är de lika eller inte?

Uppgift 4d – Map med Djur

I ditt djur-program, skapa upp en map över dina djur

- Låt djurets namn vara nyckel (du hittar på namn till dem)
- Låt själva djurobjektet vara värdet i mappen
- Skriv ut alla djur i din mapp

Uppgift 4e – When

Skapa en funktion som tar ett Djur som inparameter. Beroende på Djurets sort ska du skriva ut djurets läte. Använd en When-sats (Kotlins motsvarighet till Switch)

Uppgift 5a – Extension functions

Skapa klassen Rectangle som har följande parametrar

- Bredd
- Höjd

NACKADEMIN

Skapa extension functions till Rectangle som beräknar

- Area
- Omkrets

Gör ett huvudprogram där du

- Skapar en Rectangle
- Skriver ut area och omkrets som du beräknar i dina funktioner

Uppgift 5b – Extension functions

Skapa följande extension functions för String:

DoubleUp() - Skriver ut en String två gånger

Uppgift 5c – Extension functions

Skapa följande extension functions för String

PrintRepeat(i : Int) - Skriver ut en String i antal gånger

Uppgift 6a – Lambdas

Bygg ut djurklasserna i övningsuppgift 4:

Gör om utskriften av listan så att du använder en lambda för att skriva ut djuren

Testa minst 2 olika sätt att skriva ut.

Uppgift 6b – Lambdas

Skriv en funktion som tar en lista av olika djur och gör om alla djuren till grodor. Använd lambdas för detta

Uppgift 6c – Lambdas

Skriv en funktion som tar en lista av djur och sorterar bort alla spindlar (Eller någon annan sorts djur som du har i din lista)

Uppgift 6d – Lambdas

Skriv en funktion som kollar om det finns några katter i din lista. Testa både med en sorts djur som finns i listan och en sort som inte finns

Uppgift 6e – Lambdas

Skriv en funktion som hittar det högsta antalet ben som ett djur i listan har (viss googling kan behövas för funktionen som hittar max-värdet)

Uppgift 6f – Lambdas

Skriv en funktion som summerar antal ben djuren har (viss googling kan behövas för funktionen som hittar max-värdet)

Uppgift 6g – Lambdas

Skriv en funktion som berättar hur många insekter som finns i listan (det är djur med fler än 4 ben)

NACKADEMIN

Uppgift 6h – Lambdas

Skriv en funktion som fördubblar antalet djur i den lista den returnerar. (Kräver ingen lambda)

Uppgift 7a – Rekursion, ränta på ränta

Skriv en rekursiv funktion beräknar hur mycket ränta du får efter ett viss antal år.

Inparametrar till funktionen är

- Initialt belopp
- Antal år
- Ränta

Lite inspo till hur den rekursiva metoden kan byggas upp:

- Du sätter in 100 kr med 10% ränta
- Efter ett år har du $100 * 1,1 = 110$ kr
- Nästa år har du $(100 * 1,1) * 1,1 = 121$
- Efter 3 år har du $((100 * 1,1) * 1,1) * 1,1 = 133,10$
- Efter 5 år har du $((((100 * 1,1) * 1,1) * 1,1) * 1,1) * 1,1 = 161,05$

(Det går utmärkt att skriva en ränteuträknings-metod som inte är rekursiv, men nu är det rekursivitet som är övningen)

Uppgift 7b – Rekursion, summering

Skriv en rekursiv funktion som tar en lista av ints och returnerar summan av talen i listan

Uppgift 7c – Rekursion, max-värde

Skriv en rekursiv funktion som tar en lista av ints och returnerar maxvärdet av talen i listan

Uppgift 7d – Rekursion, max-värde

Skriv en rekursiv funktion som skriver ut en sträng baklänges

Uppgift 7e – Rekursion, balanserade parenteser

Skriv en rekursiv funktion som tar en sträng och kollar om antalet parenteser är balanserade (att det finns lika många höger som vänsterparenteser). Returnera true eller false.

Exempel:

- `:-)` inte balanserad
- `))((` balanserad

Uppgift 7f – Rekursion, fibonacci

Betrakta Fibonacci-demon i facit. Fibonacci-talen är en sekvens där varje nytt tal räknas ut genom att addera de två föregående talen i sekvensen. De första talen är 1 och 2

Sekvensen blir: 1, 2, 3, 5, 8, 13, 21 ...

Kopiera exemplet och skriv om

Man ska kunna ange en siffra som betecknar ett index i fibonnacci-sekvensen som inparameter till din funktion

Ditt program ska sedan skriva ut vilket fibonacci-tal som är på den plats som motsvarar det tal du skickade in

NACKADEMIN

Uppgift 8 – Högre Ordningens Funktioner – Text Distortion

Skriv en metod: TextDistortion, som tar in en funktion som i sin tur tar in en String och lämnar ifrån sig en String. I TextDistorsion anropas funktionen som skickas in som inparameter.

Skriv följande metoder som kan användas som inparametrar till TextDistorsion

- ToUpperCase(String) : String
- RemoveAllDigits(String) : String

Uppgift 9 – Högre Ordningens Funktioner – Generiska funktioner

- 9a Skriv en egen version av funktionen "filter"
- 9b Skriv en egen version av funktionen "map"
 - Använder inte predicate utan function (T) -> (T)
- I ditt mainprogram, skapa upp en lista och se att det funkar att göra filter eller map på listan
- Om generics känns svårt, börja med att definiera upp map och filter för listor som bara tar t.ex Int eller String
 - Nästa steg är att kunna hantera alla sorters objekt

Uppgift 10a – Mera lambdas, zip

- Skapa en lista av Strängar, som är olika namn på djur
- Skapa en annan lista med integers som är olika antal ben som djur kan ha
- Låt listorna vara lika långa
- Skriv en metod som zippar ihop de två listorna så att en ny lista skapas
 - Denna lista är en List<Djur>
 - Varje djurnamn och antal Ben i de gamla listorna är ihopparade

Uppgift 10b – Mera lambdas, zip

- Skapa upp en lista med integers,
 - [2, 3, 8, 45, 123, 256, 2345]
- Genom att använda zipWithNext, skapa en lista som innehåller skillnaden mellan varje tal
 - Alltså [1, 5, 37, 78, 133, 2089]

Uppgift 11a – Mera lambdas, gruppering av mappar

- Skapa en lista av olika sorters djur
- Skapa en map som är grupperad på sort av djur
 - Nyckel: sort, Värde: Djur
- Skriv en utskriftsfunktion som traverserar den grupperade mappen och skriver ut varje nyckel, följt av antal djur som ligger i nyckelns värde

Uppgift 11b – Mera lambdas, traversering av mappar

- Skapa en lista av olika sorters djur
- Skapa en map som är grupperad på antal ben
 - Nyckel: antalBen, Värde: Djur
- Skriv en utskriftsfunktion som traverserar den grupperade mappen och skriver ut varje nyckel, följt av de sorter av djur som är sorterade under varje nyckel
 - Helst inga dubblerade värden på djursorter

NACKADEMIN

Uppgift 11c – Mera lambdas, filtrering av mappar

- Utifrån de mappar du skapade i uppgift 11a-b
- I mappen som har djurens sorter som nyckel
 - Filtrera mappen så att den bara visar de djur som börjar på bokstaven K
 - Filtrera mappen så att den bara visar de djur som det finns ett ex av

Uppgift 11d – Mera lambdas, mera mappar

- Skriv en funktion som kollar om du har en katt i din map
- Skriv en funktion som kollar om alla djur i mappen är katter

Uppgift 11e – Mera lambdas, fold

Gör en funktion där du adderar fram summan av dina djurs benantal

- Använd Fold
- (Varför är fold lämpligare än reduce i detta fall?)
- (lättare att jobba utifrån den ursprungliga listan än från de mappar som skapats)

Advent of Code 1a+1b (2020)

Gå till Advent of Code och regga konto på <https://adventofcode.com/>

Läs instruktionerna till uppgift 1a

När uppgift 1a är löst visas uppgift 1b

Denna uppgift passar bra efter att vi gått igenom **for-loopar och listor** i Kotlin.

För att läsa in alla integers i en lista:

- Kopiera input:en från AoC och lägg i en fil i ert IntelliJ-projekt
 - För att läsa in data till er Kotlin-klass och skapa en lista av Int:
 - Anpassa pathen nedan till er path
 - Listor kan hanteras som i Java, med `get(i)`, `add(e)` och liknande
- ```
val list2 =
 File("src/AoC/Dec1/input.txt").readLines().map{it.toInt() }
```

## Advent of Code 2a + 2b (2020)

Lös Aoc 2 a+b. Detta passar bra efter att vi gått igenom lambdas. Sista delen (se nedan) av uppgiften passar bra efter att vi gått igenom Högre Ordningens Funktioner.

Ett tips för att underlätta parsningen är att använda Split. För ha flera tecken att splitta på: `split(" ", ":", "-")`

För att läsa in inputen till en lista av String:

- `val list2 = File("src/AoC/Dec2/input.txt").readLines()`

Du kommer att märka att a och b-uppgifternas lösningar är nästan identiska

- Först sker en parsning som kommer att se likadan ut i både a och b
- Bara bearbetningen på slutet är olika

Dessa 2 uppgifter passar perfekt att slå ihop



# NACKADEMIN

- Lagra "bearbetningsfunktionerna" i 2 olika variabler
- Passa funktionerna som inparametrar till en metod som först genomför parsningen, sedan kör den metod som passades som inparameter

## Advent of Code 3a (+3b)

Denna uppgift passar bra efter att vi gått igenom **for-loopar och listor** i Kotlin.

För att läsa in alla rader från input-filen i en lista:

- Kopiera input:en från AoC och lägg i en fil i ert IntelliJ-projekt
  - För att läsa in data till er Kotlin-klass och skapa en lista av String:
    - Anpassa pathen nedan till er path
    - Listor kan hanteras som I Java, med `get(i)`, `add(e)` och liknande
- ```
val list2 = File("src/AoC/Dec3/input.txt").readLines()
```

Advent of Code 5a+5b (2020)

Tricket med denna uppgift är att inse att om man byter ut B och R mot 1 och F och L mot 0 bildas ett binärt tal, detta tal är ert svar

För att parsea en String till ett binärt tal: `String.toInt(2)`

Advent of Code 6a (2020)

Denna uppgift passar bra efter att vi gått igenom **Set** i Kotlin.

Advent of Code 8a+8b (2020)

Lös Advent of Code 8a och 8b. Ett tips är att använda `When`

För att parsea raderna: `val fields = i.split(" ")`

Delarna kommer man åt genom: `fields[0]`, `fields[1].toInt()`

Advent of Code 10a+10b (2020)

En inte helt lättbegriplig uppgift!

Vad ni behöver göra är att skapa en lista på alla volt-tal i input:en, samt lägga till ett element med värdet 0 och ett med värdet "det högsta volttalet + 3"

I b-uppgiften måste ni göra en lista med skillnaderna mellan alla närliggande värden i listan
`zipWithNext`

När en 3a dyker upp finns ingen ersättning. När flera ettor står i rad, räkna ettorna och klura ut hur många varianter de kan göras om till. För varje grupp ettor, multiplicera antalet varianter för att få antalet permutationer.

Lämplig att göra efter att vi gått igenom `zipWithNext`

För er som vill ha mer

Gör de officiella Kotlin-koans som finns på nätet.

NACKADEMIN

Hitta tutorials om lambdas och funktionell programmering på nätet.

Signa upp på en kurs i diskret matematik för att förbättra förståelsen för lambdas, logik, kombinatorik och mängdlära. Denna matematik är INTE som matten i skolan, utan mera av hjärngympa-karaktär.

T.ex Coursera har kurser i diskret matematik för programmerare:

<https://www.coursera.org/specializations/discrete-mathematics>

Notera att det går att ta kurserna gratis genom att man "Audit":ar (även om de försöker gömma att detta är möjligt). Då får man tillgång till allt material och föreläsningar, men ingen hjälp med feedback, rättningar av uppgifter och man får heller inget diplom eller betyg.

Gå in på <https://codesignal.com> och fighta loss.

Facit

Sigruns lösningar: https://github.com/sigrunolafsdottir/FunctionalProgramming_Kotlin