Veckoövning V7: Kontaktlista

Under veckans föreläsning har vi tittat närmare på generiska klasser och samlingar, så som `List` och `Map` ("avbildningar" i Skansholms bok). I den här övningen kommer du att använda dem till att bygga en kontaktlista, som du kan lägga till och slå upp namn i.

1. Förberedelser

Gå igenom veckans föreläsning och läsanvisningar. Om du inte känner dig säker på relationen mellan primitiva typer och referenstyper, eller hur `==` och `equals` skiljer sig kan det vara en god idé att gå tillbaka och fräscha upp minnet om detta.

Det kan även vara klokt att ha dokumentationen för Map och List nära till hands när du gör övningen.

2. Modellera dina kontakter

Till att börja med behöver vi modellera kontakterna vi kommer att fylla vår kontaktlista med. Hitta på en lämplig klass, med namn `Entry`, `Contact`, `Person`, eller något annat passande. Klassen bör kunna innehålla åtminstone följande instansvariabler och metoder:

- Minst ett namn.
- Noll eller flera telefonnummer per kontakt.
- Noll eller flera emailadresser per kontakt.
- En toString-metod, som returnerar en textrepresentation av kontakten. Denna representation kan exempelvis se ut så här: `Namn: Rainbow Dash Webbsida: example.com Email: 2fast4u@example.com Telefon: 031-1234567, 070-7654321` Tips: använd `String.format` för att snygga till utskriften, men lägg inte för mycket tid på att få den perfekt.

Använd listor (*inte* arrayer) för att modellera saker (t ex telefonnummer) som det kan finnas flera av.

3. Modellera kontaktlistan

Nu ska vi implementera själva kontaktlistan. Tanken är att vi ska kunna söka efter våra kontakter baserat på namn, så använd en `Map<String, Contact>` (om din klass från steg 2 hette `Contact`) för att representera kopplingen mellan namn (`String`) och poster i kontaktlistan.

Skapa en klass `Contacts` som har följande publika metoder:

- `void add(Contact contact)`: lägger till den givna kontakten i kontaktlistan, med sitt namn som nyckel. Om en kontakt med samma namn redan finns skrivs den över med den nya kontakten.

- `Optional<Contact> lookup(String name)`: returnerar kontakten med det givna namnet, insvept i en Optional. Om ingen sådan kontakt finns returneras `Optional.empty()`.
- `void remove(Contact contact)`: tar bort den givna kontakten ur kontaktlistan.

Testa din kontaktlista genom att ladda in `Contacts` och `Contact` i jshell, skapa en kontaktlista, och lägga till/ta bort/slå upp några kontakter i den. Alternativt kan du göra detta i en `main`-metod.

4. Kontakter med samma namn

Det är inte ovanligt att folk har samma namn, men vår kontaktlista klarar ännu inte av detta. Ändra kopplingen mellan namn och poster så den istället har typen `Map<String, List<Contact>>` - dvs. varje namn kopplas till en lista av poster istället för en enda post.

Ändra sedan klassens metoder på följande vis:

- Se till att samtliga metoder fungerar med den nya kopplingen.
- `add` ska inte skriva över tidigare kontakter med samma namn. Istället ska den nya kontakten läggas till i samma lista som de tidigare kontakterna med samma namn.
- Låt `lookup` returnera en `List<Contact>` istället för en `Optional`. Om inga kontakter med det givna namnet finns returneras en tom lista.

5. Användargränssnitt

Skriv en klass `ContactProgram` som innehåller en `main`-metod (plus eventuella hjälpmetoder du tycker behövs). När programmet körs visar det ett textbaserat gränssnitt för användaren, som låter hen lägga till, ta bort och slå upp kontakter.

En sådan interaktion kan t ex se ut så här:

```
Super Contact List 1.0
---
Use one of the verbs "add", "find" or "remove" followed by the contact's name.
---

What do you want to do?
> add Rainbow Dash
OK! What is Rainbow Dash's phone number (comma-separated list)?
> 031-1234567, 070-7654321
OK! What is Rainbow Dash's email (comma-separated list)?
> 2fast4u@example.com
OK! What is Rainbow Dash's website?
> example.com
OK, adding Rainbow Dash to contact list!
```

```
What do you want to do?
> find Rainbow Dash
Name:    Rainbow Dash
Website: example.com
Email:   2fast4u@example.com
Phone:   031-1234567, 070-7654321


What do you want to do?
> remove Rainbow Dash
OK!

What do you want to do?
> find Rainbow Dash
No such contact!
```

Tips: använd `Scanner.next()` för att läsa det första ordet på en rad, och därefter `Scanner.nextLine()` för att läsa resten av raden. Du kan använda `String.trim()` för att ta bort oönskad whitespace i början och slutet av en sträng.