

Extrauppgift

Detta är en extrauppgift där vi ska skapa ett program som skall kunna göra lite olika saker. När det startar upp skall det finnas en meny som låter användaren välja vad denne vill göra.

Exempelkörning

```
Vad vill du göra?  
1. Addera två tal  
2. Räkna bokstäver i en sträng  
3. Spegelvänd en sträng  
4. Summera alla tal i en sträng
```

Uppgifterna nedan går ut på att implementera menysystemet och de olika funktionerna som programmet skall klara av att göra.

1.

Det första du skall göra är att skapa själva menysystemet för programmet. Detta skall fungera på följande sätt:

Först välkomnar det användaren och visar en lista på vad för olika saker man kan göra i programmet. Det skall sedan låta användaren mata in vad den vill göra och sedan köra rätt kod beroende på valet.

Eftersom vi än inte har skrivit någon kod för de olika funktionerna i programmet kan det vara bra att göra en enkel utskrift som säger vilket menyalternativ användaren valde, för att se testa om menysystemet faktiskt fungerar.

Exempelkörning

```
Vad vill du göra?  
1. Addera två tal  
2. Räkna bokstäver i en sträng  
3. Spegelvänd en sträng  
4. Summera alla tal i en sträng  
1  
Här adderar vi två tal
```

(Det gröna i bilden är användarinput)

2.

Vi skall nu skriva koden som sköter det första menyalternativet, alltså som låter användaren addera två stycken tal.

Det första som skall hända är att programmet frågar användaren efter de två talen och låter denne mata in dem. Efter detta skall programmet sedan summera talen och skriva ut resultatet.

Exempelkörning

```
Vad vill du göra?  
1. Addera två tal  
2. Räkna bokstäver i en sträng  
3. Spegelvänd en sträng  
4. Summera alla tal i en sträng  
1  
Första talet:5  
Andra talet:3  
Summan är 8
```

3.

Nästa steg är att skriva koden för det andra menyalternativet. Här skall programmet räkna hur många gånger en given bokstav finns i en given sträng.

Först frågar programmet efter en sträng och låter användaren mata in en. Sedan frågar det efter den bokstaven som användaren vill räkna förekomsten av. Det skall nu kolla hur många gånger den här bokstaven finns i strängen och sedan skriva ut resultatet.

Det finns inget smidigt sätt att bara hämta ett tecken med hjälp av Scanner i Java. Ett sätt att göra detta är på följande sätt:

```
char c = sc.nextLine().charAt(0);
```

Här antar vi att vi redan har en Scanner som heter sc. Vi sparar sedan den första karaktären i textsträngen användaren matar in i en character-variabel.

Exempelkörning

```
Vad vill du göra?  
1. Addera två tal  
2. Räkna bokstäver i en sträng  
3. Spegelvänd en sträng  
4. Summera alla tal i en sträng  
2  
Ange ordet du vill leta i: Hallå  
Ange bokstaven du vill leta efter: l  
Det finns 2 l i ordet Hallå
```

4.

Du skall nu refaktorera ditt program så att all kod som gäller för en viss funktion ligger i sin egen metod. Det skall alltså finnas en egen metod för varje menyalternativ.

5.

Slutligen skall vi se till att programmet inte avslutas efter att användaren har gjort något. Istället skall det igen fråga vad man vill göra. För att avsluta programmet skall användaren mata in ett 'e' i huvudmenyn. Programmet skall alltså köra så länge inmatningen i menyn är skilt från 'e'.

```
Vad vill du göra?  
1. Addera två tal  
2. Räkna bokstäver i en sträng  
3. Spegelvänd en sträng  
4. Summera alla tal i en sträng  
e. Avsluta  
1  
Första talet:3  
Andra talet:2  
Summan är 5  
  
Vad vill du göra?  
1. Addera två tal  
2. Räkna bokstäver i en sträng  
3. Spegelvänd en sträng  
4. Summera alla tal i en sträng  
e. Avsluta  
e  
Hejdå
```

6.

Det tredje menyalternativet skall låta användaren spegelvända en sträng. Första skall användaren mata in strängen den vill spegelvända, sedan skall programmet helt enkelt skriva ut hur strängen ser ut spegelvänd.

Körningsexempel

```
Vad vill du göra?  
1. Addera två tal  
2. Räkna bokstäver i en sträng  
3. Spegelvänd en sträng  
4. Summera alla tal i en sträng  
e. Avsluta  
3  
Ge mig en sträng: Hallå kompis  
sipmok ållaH
```

7.

Den sista funktionen skall låta användaren skriva in en sträng av siffror och sedan addera alla siffror i strängen.

Först skall den fråga användaren efter strängen och låta den mata in siffrorna. Här är det viktigt att programmet faktiskt hämtar en textsträng från användaren och inte ett tal.

Den skall sedan summera alla siffror i den här textsträngen.

Alla characters i Java lagras egentligen som ett heltal. Detta heltal representerar vilken sifferkod som stämmer ihop med karaktären i fråga. Om man konverterar från char till int genom en typkonvertering kommer man därför att få vilken sifferkod som charen hade. Detta kan vi utnyttja när vi vill få reda på siffervärdet på charen. Sifferkoden för '0' är 48, '1' är 49 osv. Därför kan man få fram siffervärdet på en sifferchar genom att subtrahera 48 från den efter att man har konverterat den till en int.

Om strängen skulle innehålla ett tecken som inte är en siffra skall ett felmeddelande skrivas ut och ingen summa skall beräknas. För att underlätta den här checken skall en hjälpmetod

`public static boolean isDigit(char c)` skrivas. Den skall returnera true om den givna karaktären är en siffra och false om den inte är det.