

TDDC77s Datorbaserad Tentamen

2018-08-28 kl 14:00-18.00

- Det är inte tillåtet att ha telefoner, böcker, icke tomma papper eller annat hjälpmaterial till hands under tentamen.
- Det är tillåtet att ha lösa tomma papper och pennor.
- Räck upp handen för att få hjälp om något inte fungerar i systemet
- För betyg 3 krävs godkänt på en av uppgifterna. För betyg 4 krävs två godkända uppgifter. För betyg 5 krävs tre godkända uppgifter.
- Det spelar **ingen** roll i vilken ordning du löser uppgifterna. Börja med den som verkar enklast för dig. Fortsätt med nästa uppgift så fort du har skickat en lösning.
- API:et kan öppnas med chrome under:
`http://www.ida.liu.se/~TDDC77/extra/api-7/index.html`
- Dokumentera ordentligt din kod och alla metoder.

Rättning:

- Steg 1: Vi kompilar och provkör ditt program. Vi tittar på programkoden. Programmet kan bli godkänt (isf går vi till steg 2). Annars:
 - Ditt program får komplettering. E.g. vi har några testfall som programmet inte klarar, eller programmet uppfyller inte vissa krav. Du får kommentarer och möjligheten, om tiden tillåter det, att skicka ett nytt program.
 - Ditt program blir underkänt: programmet går inte att kompilera eller klarar inte de testfall som kommer med uppgiften. Du får inte möjlighet att skicka komplettering för den uppgiften. Du får fortsätta med nästa uppgift.
- Steg 2: Om uppgiften blev godkänd uppdateras strax ditt betyg.

1 Sortering

Skriv ett program som läser in sina argument från terminalen:

- Man anropar programmet med "java Sort a0 a1 ... an" där alla a0, a1, ... an är heltal. Programmet sorterar a0, a1, ... an i stigande ordning.
- Programmet ska hantera alla situationer där användaren skriver in något som inte matchar beskrivningen ovan. I dessa fall ska programmet inte krascha utan terminera (utan errors) efter att ha skrivit ut:

```
Syntax: Sort a0 a1 ... an
With : a0, a1, ... an are integers
E.g. : Sort 10 -2 5 9
```

- Du ska använda dig av följande metod i klassen Integer:
`public static int parseInt(String s) throws NumberFormatException`
- Du ska inte använda dig av en färdig sortering metod i API:et, utan implementera din egen sortering metod enligt beskrivningen nedan.
- Antar att programmet får a0 a1 ... ai ... an som argument (där i är mellan 0 och n. Programmet ska göra följande:
 - *. För varje i från 0 till n
 - *. För varje j från i+1 till n
 - *. Om ai > aj bytta ut (eng. swap) ai och aj (i.e., ai tar värdet av aj och aj värdet av ai).
 - *. Skriva ut värdena på de nya och sorterade a0 a1 ... an

```
$ java Sort 10 -2 5 9
Sorted: -2 5 9 10
```

```
$ java Sort 10
Sorted: 10
```

```
$ java Sort
Sorted:
```

```
$ java Sort 0 -1 1 0 -1 1 1
Sorted: -1 -1 0 0 1 1 1
```

```
$ java Sort 10 -2w 5 9
Syntax: Sort a0 a1 ... an
With : a0, a1, ... an are integers
E.g. : Sort 10 -2 5 9
```

2 Calculator

Nedan följer ett antal faktapunkter. Din uppgift är att skapa den specificerade hierarki. Ange så pass mycket, MEN INTE MER, av alla klasser, gränssnitt (eng. interfaces), variabler, metoddeklarationer och implementationer så att allt som är specificerat i uppgiften implementeras. Du behöver inte dokumentera din kod. Den ska respektera inkapsling och använda sig av lämpliga namn som tillämpar Java konventioner. Kopiera filen `Driver.java` som finns under `given_files/` till en ny mapp under din hemkatalog. Skapa och lägg till de Java-filer som behövs för din lösning. Man ska kunna kompilera dina filer och köra `Driver` för att få som output innehållet av `Driver.output.txt`. Skicka alla Java-filer.

Specifikationer:

- `Expression` är ett gränssnitt (eng. interface) med två metoder `int getValue()` och `String toString()`.
- `Variable` är en klass som har en privat `int` variabel och fyra offentliga (eng. public) metoder. Klassen implementerar gränssnittet `Expression` där `getValue()` returnerar värdet på den privata variabeln och `toString` returnerar variabelns värde som en `String` (du kan använda dig av API metoden `Integer.toString(int x)`). Klassen har ytterligare två metoder som tilldelar deras parameter till den privata variabeln: konstruktören `Variable(int x)` och en setter `void setValue(int x)`
- `Binary` är en abstrakt klass som implementerar `Expression` och som har tre variabler: en `String` variabel `name` och två `Expression` variabler `left` och `right`. Klassens `toString()` är offentlig (eng. public) och returnerar strängen `"(" + left + " " + name + " " + right + ")"`
- `Plus` är en klass som utökar (eng. extends) abstrakta klassen `Binary`. Klassens konstruktör (eng. constructor) tar två `Expression` objekt som argument och tilldelar de till variablerna `left` och `right`. Konstruktören tilldelar också ett sträng värde (som du ska välja själv) till variabeln `name`. Klassens `getValue()` är offentlig och returnerar summan av attributerna (i.e., `left.getValue() + right.getValue()`)
- `Minus` är en klass som utökar (eng. extends) abstrakta klassen `Binary`. Klassens konstruktör (eng. constructor) tar två `Expression` objekt som argument och tilldelar de till variablerna `left` och `right`. Konstruktören tilldelar också ett sträng värde (som du ska välja själv) till variabeln `name`. Klassens `getValue()` är offentlig och returnerar skillnaden mellan attributerna (i.e., `left.getValue() - right.getValue()`)

3 Not three in a row

Kopiera filen `given_files/NotThreeInARow.java` till en ny mapp under din hemkatalog. Programmet fyller i flera bräde med 0 och 1 tills att den hittar ett bräde där alla kolumner och rader är valida. En rad eller kolumn är valid om:

- den har lika många celler som innehåller 0 som celler som innehåller 1
- den längsta sekvens av celler med samma siffran är mindre än, men inte lik med, 3

Kompletera den givna koden enligt de anvisningar som finns i form av kommentarer märkta med "TODO: ...". Koden skall skrivas enligt Javas kodkonventioner och kommenteras väl. Koden ska kunna köras. En körning kan se ut så här:

```
$ java NotThreeInARow
1 1 0 0
0 1 0 1
0 1 0 1
1 1 1 1

0 0 0 1
0 1 1 0
0 1 0 0
0 0 0 0

0 1 0 0
1 0 1 0
0 0 0 0
1 1 0 0

...

1 0 0 0
1 1 0 0
0 0 0 0
1 1 0 0

1 0 1 0
0 1 0 1
0 1 0 1
1 0 1 0
```