
Instruktioner för distanstenta

Anvisningar:

Alla lösningar bör skrivas efter varandra i **en** textfil (.txt).

Filen måste börja med studentens namn och en kort fritext där studenten intygar att alla lösningar i filen är skrivna på egenhand utan hjälp från andra och utan kopiering från internet.

Varje svar måste börja med **en tydlig rubrik** i textfilen, t.ex.:

UPPGIFT 2

=====

Del A)

...

Del B)

...

UPPGIFT 3

=====

...

Man får ta kod från föreläsningpresentationerna, sina egna labblösningar och man får söka information om Javas API med Google.

Man får inte kommunicera med andra studenter eller på annat sätt be om lösningar. Man får inte söka lösningar på internet.

Lösningar bör skickas in som en textfil (dvs en fil som slutar med .txt) via Canvas Assignments innan tentan tar slut, dvs innan klockan 12:30.

Lycka till!

Kontakt:

Vid frågor går det att få tag på examinatorn Magnus Myréen med att ringa 0729744943 under tentans gång.

DAT050 Objektorienterad programmering

Tentamen, onsdag, 2021-08-25, 14:00-18:00

Vitsordsgränser: 3=24p, 4=36p, 5=48p, max 60p.

Kom ihåg att inte fastna på en uppgift. Bestäm i förväg din egen tidsgräns per uppgift. **Lycka till!**

Uppgift 1: [10 poäng totalt]

- A. Java har ett nyckelord `private`. Förklara betydelsen av det nyckelordet och hur man vanligtvis brukar använda det. [3 poäng]
- B. Inkapsling och gränssnitt är viktiga koncept i objektorientering. Förklara dessa begrepp. [5 poäng]
- C. Förklara kort vad UML diagram är. [2 poäng]

Uppgift 2: [totalt 15 poäng]

Denna uppgift handlar om att skriva en klass för enkla lottorader (utan extra/bonus siffror). I denna uppgift antar vi att lottorader består av 7 siffror där varje siffra är mellan 1 och 40. En lottorad får inte innehålla duplikat, dvs

1, 20, 32, 20, 7, 40, 2

är inte en giltig lottorad för att siffran 20 upprepas. Ordningen på siffrorna spelar ingen roll.

Definiera `LottoRow` klassen så att den har:

- A. En konstruktor som givet sju tal, som en parameter av typen `int[]`, skapar en instans av `LottoRow` som representerar den lottoraden. Denna konstruktor bör kasta ett `IllegalArgumentException` ifall det är något som går fel, t.ex. ifall det inte finns sju tal i parametern `int[]` eller de talen av någon annan anledning inte är en giltig lottorad. För fulla poäng ska `LottoRow` klassen ska vara immutable. [8 poäng]
- B. En metod `toString` som returnerar en sträng representation av lottoraden. [3 poäng]
- C. En instansmetod med följande signatur:

```
int countMatches(LottoRow other)
```

Denna ska returnera hur många siffror som stämmer samman med den andra raden, `other`. Exempel: ifall raderna inte har någon siffra gemensam ska denna metod returnera 0. Ifall alla siffror finns i den andra lottoraden då ska den returnera 7. [4 poäng]

Uppgift 3: [10 poäng totalt]

Denna uppgift handlar om att beräkna en checksum (dvs en kontrollsiffra) och att skapa en lista med den på slutet. Skriv en metod som tar som input en lista an heltal (Java typ: `List<Integer>`) och returnerar en lista av heltal med en extra siffra tillagd på slutet. Siffran som läggs till på slutet bör vara summan av alla siffror i den givna listan modulus 7001.

Obs: metoden får inte ändra i listan som var given som input.

Uppgift 4: [10 poäng totalt]

Följande interface representerar en trädstruktur. Idén är att `getValue()` returnerar `int` värdet vid roten av trädet och `getSubtrees()` returnerar en iterator med alla subträd som kan nås med ett steg från denna roten av detta träd.

```
public interface Tree {  
    public int getValue();  
    public Iter<Tree> getSubtrees();  
}
```

Skriv en metod med följande signatur som beräknar max värdet som är sparad i ett träd.

```
public static int maxValOf(Tree t) { ... }
```

Obs: man får anta att det inte finns `null` någonstans och att trädet är ändligt.

Uppgift 5: [15 poäng totalt]

Skriv ett litet grafisk program som mäter tiden mellan musklickar. Idén är att programmet har ett tomt fönster. När man klickar i fönstret skriver programmet ut med `System.out.println` tiden i millisekunder sedan man förra gången klickade i fönstret. Första gången man klickar ska den visa hur många millisekunder det har gått sedan programmet startade.

Tips: använd `System.currentTimeMillis()` för att få tiden i millisekunder sedan 1 januari 1970. Denna metod returnerar en `long`, vilket går att använda på precis samma sätt som `int`.