

16-okt-2012/FK

ID1004 Laboration 2, 31 oktober - 1 november 2012

Labben bör göras individuellt. Beräknad tid ca 1-2 timmar.

Instruktionen antar att labben utförs i datasal, med hjälp av den integrerade utvecklingsmiljön Eclipse. Alternativt kan du använda egen dator, någon lämplig texteditor, Java JDK och kompilera och köra själv. Om du väljer editorn TextPad, tänk att ha installerat Java JDK innan du installerar TextPad, eftersom TextPad vid sin installation då konfigureras för att stödja källkod i Java.

Labben innehåller uppgifter av varierande svårighetsgrad. Välj en vid labbtillfället, men gör gärna alla.

Resurser online

Java version 6 API dokumentation

<http://download.oracle.com/javase/6/docs/api/>

Java JDK download (SE version 6)

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

TextPad editor for Windows

<http://www.textpad.com/>

Eclipse IDE (välj Classic)

<http://www.eclipse.org/downloads/>

Namngivningskonventioner i Java

Namn på klasser, metoder, variabler och annat som programmeraren väljer kallas identifierare. Identifierare i Java skiljer på gemener och versaler. Dessa identifierare är alltså alla olika: a, A, foo, Foo, FoO.

Namn på klasser i Java skrivs med stor begynnelsebokstav, t ex String.

Namn på variabler och metoder skrivs med liten begynnelsebokstav, t ex toString(), limit.

Namn på konstanter (dvs variabler vars värde inte går att ändra) skrivs med versaler, t ex MAX_INT.

Uppgift L2.1 Fyra tal

Skriv en klass som har följande fält och metoder:

- Lagringsplats för fyra tal
- En metod som låter användaren mata in fyra tal och sparar dem.
- En metod som skriver ut talens summa.
- En metod som skriver ut talens medelvärde.
- En metod som skriver ut talens produkt.
- En metod som skriver ut det minsta och högsta talet.
- En public metod som anropar de andra metoderna i tur och ordning.

Välj förklarande namn på metoderna och inled namnen med liten bokstav.

Låt metoden main skapa en instans av klassen och anropa den publika metoden.

Tips: `java.util.Scanner`, `java.util.Arrays.sort()`

Uppgift L2.2 Sortera slumpstal

Skriv en klass som har följande fält och metoder:

- En variabel för det önskade antalet slumpstal
- En array för slumpstalen
- En metod som frågar användaren hur många slumpstal som önskas, och låter användaren mata in ett tal mellan 10 och 100. Om det inmatade talet är utanför det tillåtna området ska programmet fråga om tills det blir rätt. Det inmatade talet lagras i klassen.
- En metod som slumpar fram det önskade antalet och lagrar dem i en array. Slumpstalen ska dras ur området 0.0 – 1.0.
- En metod som sorterar slumpstalen i stigande ordning.
- En metod som skriver ut talen.
- En public metod som anropar de andra metoderna i tur och ordning.

Välj förklarande namn på metoderna och inled namnen med liten bokstav.

Låt metoden main skapa en instans av klassen och anropa den publika metoden.

Tips: `java.util.Scanner`, `java.util.Random`, `java.util.Arrays.sort()`

Uppgift L2.3 Spelet Gissa Talet

Skriv en klass som implementerar spelet gissa talet. Detta går till så att spelet drar ett slumpstal mellan 0 och 100. Spelaren får 10 chanser att gissa talet. Efter varje gissning ska spelet skriva ut om gissningen var rätt (och spelet är slut), för hög, eller för låg. När spelaren använt sina 10 gissningar utan att ha gissat rätt är spelet slut. Varje gång spelet är slut, oavsett om sista gissningen var rätt, ska hemligheten och spelarens gjorda gissningar skrivas ut i den ordning de gjordes.

Skriv en klass med följande fält och metoder:

- En variabel som innehåller hemligheten, dvs det tal som spelet tänker på.
- En konstant (final int) som innehåller max antal gissningar i en omgång.
- En variabel som innehåller antalet gissningar som spelaren har gjort.
- En array som innehåller spelarens gissningar.
- En metod som nollställer spelet: antal gissningar sätts till noll, en ny hemlighet slumpas fram, spelet skriver ut att det tänker på ett tal.
- En metod som hämtar en gissning från användaren, sparar den i nästa position i arrayen och returnerar gissningen (Scanner).
- En metod som jämför spelarens gissning med hemligheten. Om gissningen är för låg eller hög skriver metoden ut vilket och returnerar `false`. Om gissningen är korrekt skriver metoden ut det och returnerar `true`.
- En metod som returnerar det antal gissningar som spelaren har kvar.
- En metod som skriver ut hemligheten och de gissningar som spelaren har gjort.
- En metod som frågar spelaren om denne vill spela igen och returnerar `true` eller `false` beroende på hur spelaren svarar (Scanner).
- En omgångsmetod som med hjälp av de andra metoderna spelar en omgång av spelet. Spelet nollställs, och gissningar efterfrågas tills dess att spelaren gissar rätt eller inte har några gissningar kvar. Besked om vinst eller förlust skrivs ut (Scanner).
- En public spelmetod som skriver en kort välkomst- och förklaringstext, kör en omgång, frågar om spelaren vill spela igen och låter spelaren spela så länge spelaren önskar. Denna metod kan med fördel skapa en Scanner och skicka med den som argument till de metoder som behöver använda den.
- Ett huvudprogram (metoden main) som skapar en instans av spelet och anropar spelmetoden.

Tips: `java.util.Scanner`, `java.util.Random`