Tentamen Programmeringsteknik I 2011-03-17

Skrivtid: 1400-1700
Hjälpmedel: Java-bok

Tänk på följande

- Skriv läsligt! Använd *inte* rödpenna!
- Skriv bara på framsidan av varje papper.
- Börja alltid ny uppgift på nytt papper.
- Lägg uppgifterna i ordning. Skriv uppgiftsnummer och pin-kod (eller namn om du saknar sådan) på alla papper. Skriv inte längst upp i vänstra hörnet - det går inte att läsa där efter sammanhäftning.
- Fyll i försättssidan fullständigt.
- Det är principer och idéer som är viktiga. Skriv så att du övertygar examinatorn om att du har förstått dessa även om detaljer kan vara felaktiga.
- Programkod skall vara läslig dvs den skall vara vettigt strukturerad och indenterad. Namn på variabler, metoder, klasser etc skall vara beskrivande men kan ändå hållas ganska korta.
- Betygsgränser: 18 ger säkert 3, 27 ger säkert 4, 36 ger säkert 5.

Lycka till!

Uppgifter

1. Skriv lösningen på denna uppgift (1 a och b) direkt på detta ark, riv ut det och häfta ihop det med övriga papper du lämnar in!

   a) Nedan ser du 6 par med kodsatser (i - vi). Under koden finns en figur med sex linjer som skall fyllas i med kod. På varje linje skall det stå en sats. Från varje par ska du välja en sats (A eller B). Placera satserna du väljer i rätt ordning så att du får javakoden till en metod som går att kompilera utan fel. Metoden ska räkna ut arean för en triangel där bas och höjd är givna parametrar. Metoden skall kontrollera att båda parametrarna är icke-negativa.

   i. A. `if (bas  < 0  ||  höjd  < 0)`
      B. `if (bas  < 0 &&  höjd < 0)`

   ii. A. `double area = -1;`
      B. `int area = -1;`

   iii. A. `public double triangel(double bas, double höjd)`
      B. `public int triangel (double bas, double höjd)`

   iv. A. `return area;`
      B. `System.out.println("Arean av trianglen är " + area);`

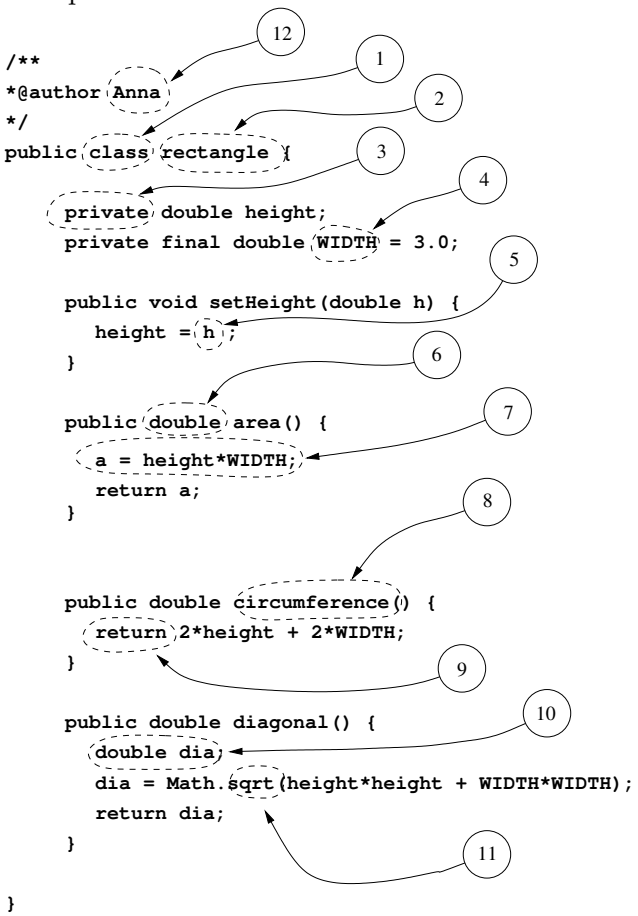
   v. A. `return;`
      B. `System.out.println("Ogiltiga mått på triangeln");`

   vi. A. `area = bas*höjd/2;`
      B. `(bas*höjd)/2 = area;`

```
_____ {
    _____
    _____ {
        _____
    } else {
        _____
    }
    _____
}
```

(3p)

b) I koden är en del konstruktioner inringade och markerade med ett tal. Under koden finns en tabell med en beskrivning eller en term per rad. Du ska för varje rad i tabellen bestämma vilken av de markerade konstruktionerna i koden som bäst svarar mot beskrivningen i tabellen. Svara med endast ett tal per rad. Observera att alla talen inte kommer med i tabellen. Ett svar är redan ifyllt som ett exempel.



Programmerarens namn	12
Klassmetod	
Avvikelse från namnkonventionerna	
Objektmetod	
Deklaration av lokal variabel	
Parameter	
Konstant	
Returtyp	
Ger kompileringsfel	

(4p)

2. Vi vill skapa ett enkelt gissningsspel där datorn tänker på ett tal, du gissar sedan och datorn talar om ifall du gissar rätt eller om ditt tal är för stort eller för litet. Du fortsätter att gissa tills du gissat rätt. Varje gång du skapar ett nytt objekt av typen `Game` tänker datorn på ett nytt tal. Du gissar genom att skriva in tal på tangentbordet.

   Det kan implementeras så här:

```
import java.util.*;        // för Scanner
public class Game {
  private int talet;        // det tal datorn tänker på

  /**
   * Konstruerar ett objekt med ett pseudoslumptal i intervallet [1,9]
   */
  public Game() {
    talet = (int) (Math.random() * 9 + 1);
  }


  /**
   * Metod som används för att jämföra din gissning med datorns val.
   * @parameter  myGuess  Tal som skall jämföras med datorns valda tal
   * @return     -1, 0 eller 1 beroende på om parametern är mindre än, lika med eller större än
   *              datorn val
   */
  public int compare(int myGuess) {
    ...
  }


  public static void main(String [] args) {
    //    Gör följande:
    //       Skapa ett objekt av typen Game med den parameterlösa konstruktorn,
    //       det betyder att datorn väljer ett tal i intervallet 1-9.
    //
    //       Upprepa tills du gissat rätt:
    //           Fråga efter en gissning genom inmating från tangentbordet,
    //           dvs med hjälp av ett Scannerobjekt.
    //           Anropa metoden compare ovan för att testa gissningen och skriv ut
    //           enligt körexemplet.
    //           Öka antalet gissningar med ett.
    //       Skriv ut antalet gissningar som behövdes för att hitta rätt tal.
  }
}
```

   Exempelvis kan det se ut så här när du kör programmet:

```
java Game
Gissa ett tal i intervallet [1,9]
Din gissning: 3
För liten
Din gissning: 5
För liten
Din gissning: 9
För stor
Din gissning: 7
För liten
Din gissning: 8
*Rätt*, 5 gissningar krävdes
```

   Dina uppgifter:

   a) Skriv metoden `compare` enligt beskrivningen ovan. (4p)

   b) Skriv metoden `main` som fungerar enligt beskrivningen ovan. (4p)

3. Antag att du har följande klass:

```
public class Tentamen {
  private int maxpoäng;     // maxpoäng på tentan
  private int[] resultat;   // array med poängfrekvens
                            // dvs i element 0 finns antal personer med noll poäng
                            // i element 1 antal personer med 1 poäng osv upp
                            // till maxpoäng.
                            // Arrayen ska alltså innehålla maxpoäng+1 element.

  public Tentamen(int max) {
     // Konstruktor där man anger önskad maxpoäng som parameter
     // Instansvariablerna, dvs maxpoäng och resultat ska initieras
     ...
  }

  public lagraResultat(int poäng) {
     // Parameter; poäng anger poängtal för en person

     // Öka lämplig plats i arrayen med 1 men kolla rimlighet av poängen först.
     // Om exempelvis poäng har värdet 5 så ska element nummer 5 i arrayen ökas med 1.
     ...
  }

  public void statistik(int godkänd) {
     // Parameter: godkänd anger lägsta poäng för godkänd
     // tenta, dvs poängtal som åtmindstone ger betyget 3

     // Metoden beräknar och skriver ut (på skärmen) antal personer som blev
     // underkända, dvs antalet personer med lägre poäng än parameterns värde
     ...
  }

  public static void main(String [] args) {
     // Skapa ett objekt av klassen Tentamen med
     // maxpoäng 40 med hjälp av konstruktorn i klassen
     //
     // Läs in 100 tentamensresultat (hela poäng) från tangentbordet, dvs med ett Scannerobjekt.
     // Lagra dessa ett i taget genom att anropa metoden lagraResultat i ditt Tentamensobjekt
     //
     // Anropa metoden statistik i ditt Tentamensobjekt för få antalet underkända utskrivet
  }
```

Din uppgift är att skriva den konstruktor och de metoder (inkl `main`) som är beskrivna ovan. Kommentarerna beskriver vad varje del ska göra.

Förklarande figur:

12	9	7	15	22	12	6	1

Exempel: En tentamen med maxpoäng 7. Vi ser att det finns 12 studenter med 0 poäng, 9 studenter med 1 poäng osv. Om vi anger gränsen för godkänt till 4 poäng ser vi att 12+9+7+15 = 43 studenter blir underkända.

(10p)

4. I en bilaga finns javakod som implementerar klassen `Bank`. En bank består av en array (`customers`) av kunder, en räknare (`numberOfCustomers`) som håller reda på aktuellt antal kunder och en räknare (`globalCustomerNumber`) som räknas upp för varje ny kund och som ser till att nya kunder får nya nummer (borttagna kunders nummer återanvänds ej).

   Om en kund tas bort flyttas de som ligger på högre index i `customers` så att befintliga kunder alltid finns på index `0` till `numberOfCustomers-1` i arrayen.

   Vissa delar av koden är utelämnade.

   I bilagan finns också dokumentation av publika delar av klasserna `Customer` och `Person` som används av klassen `Bank`. Klassen `Account` är helt utelämnad eftersom den inte är av intresse för denna uppgift.

   Dessutom finns ett körexempel.

   a) Programmet avslutas med alternativet 0 i huvudmenyn varefter man får en kontrollfråga som skall besvaras med ”ja” om man verkligen vill sluta. Som synes i slutet av körexemplet så fungerar inte detta — programmet avbryts inte. Varför inte? Hur skall man göra? (2p)

   b) Skriv klar koden i metoden `listCustomers` så att den fungerar i enlighet med körexemplet. (5p)

   c) Skriv klar koden i metoden `createCustomer` som lägger till en ny kund. Om arrayen med kunder är för liten skall den utökas med en faktor 2 (dvs vid första utökningen skall storleken ökas till 6 eftersom den är 3 initialt). (8p)

   Du får inte ändra eller lägga till något i klasserna `Customer` eller `Person`. Du får inte heller ändra i några andra metoder i klassen `Bank` förutom att rätta felet enligt deluppgift a)