Data och Informationsteknik / Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
Erland Holmström Göteborg 16 dec 2014

Examination i

PROGRAMMERINGSTEKNIK F1/TM1 TIN212 (Dugga)

Dag: Onsdag Datum: 2014-12-17 Tid: 9.00-12.00 (OBS 3 tim) Rum: V

Ansvarig lärare:	Erland Holmström tel. 1007, mobil 0708-710 600
Resultat:	Skickas ut av Ladok.
Lösningar:	Läggs eventuellt ut på hemsidan.
Granskning:	Efter rättning finns tentorna på vår expedition.
	Tid för granskning av tentan finns på hemsidan när resultaten är publiserade.
Betygsgränser:	CTH: G=15p, max 30p
Hjälpmedel:	Bravaco, Simonson: Java Programming From the Ground Up

Observera:

- Börja med att läsa igenom hela duggan så du kan ställa frågor när jag kommer.
  Jag kommer vanligen efter ca 1 timme (men det är typ 6 salar så det kan bli senare).
- Alla svar skall motiveras när så är lämpligt.
- Skriv tydligt, rita figurer. Lösningar som är svåra att läsa rättas inte.
- ”Answer concisely and to the point. ”
- De råd och anvisningar som getts under kursen måste följas svaren.
- Program skall skrivas i Java och skall vara indenterade. Använd kommentarer när det är lämpligt osv.
- Börja nya uppgifter på ny sida..

Lycka till!

Problem 1. *Sant eller falskt? Motivera. Eller svara (kort) på frågorna.*

a) Förklara varför det räcker att testa om p är delbart med alla tal d för varje $d \leqslant \sqrt{p}$ när man undersöker om p är ett primtal.

b) Ge två anledningar till att man bör deklarera instansvariabler ”private”?

c) Vad skrivs ut av följande program och varför blir resultatet så?
(Notera att det uppenbara svaret är fel)

```
public class Sum100Times {
  public static void main(String[] args) {
    final double SMALL_NUMBER = 0.01;
    double sum = 0.0;
    for (int i=1; i<=1000; i++) {
      sum = sum + SMALL_NUMBER;
    }
    System.out.println("sum = " + sum );
  } // end main
} // end Sum100Times
```

d) En klass byggs i princip upp av tre saker som man måste ha med när man skriver klassen, dess namn (identitet) är en av dom, vilka är dom andra?

(8p)

Problem 2. *Testar: val, loopar, delbarhet*

Ett tal, n, sägs vara ”vänskapligt” om den vänstra siffran i det är delbar med 1, om de två vänstra siffrorna är delbara med 2 osv. Till exempel så är 42325 vänskapligt eftersom 4 är delbart med 1, 42 är delbart med 2, 423 är delbart med 3, 4232 är delbart med 4 och 42325 är delbart med 5.

Skriv en metod, `boolean friendly(int n)`, som avgör om ett givet heltal är vänskapligt. Det underlättar om man vet hur många siffror n innehåller, skriv en metod för det också, `int nbrOfDigits(int n)`.

(7p)

Problem 3. *Testar: enkla klasser*

I slutet finns en API sida för en tärning. Skriv klassen.

9p)

Problem 4. *Testar: användande av klasser och fält*

Skriv ett program som skapar ett fält med tärningar med olika storlek (tex 6, 5 och 7 men koden skall i princip fungera även för många tärningar tex 100) och sedan kastar tärningarna och presenterar resultatet.

(6p)

Class Dice

```
java.lang.Object
  └Dice
```

```
public class Dice
extends java.lang.Object
```

A dice with 2 <= sides <= 20. This dice keep track of the number of dice that has been created and can be identifyed by it's creation number i.e. if this dice is the second created dice then its identity is 2. It also remembers its random (rolled) value.

Constructor Summary

`Dice(int size)`	1 parameter constructor.

Method Summary

int	`getSize()` Returns this dice size i.e. its number of sides.
int	`getValue()` Returns this dice value, 1<=value<=size.
int	`getWhoAmI()` Returns this dice identity, i.e. its creation number.
int	`roll()` Rolls the dice, (remembers) and returns a random value, 1<=value<=size.
java.lang.String	`toString()` Returns a string representaion of this dice including its value and its identity.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Lösningar Dugga TIN212 2014-12-17

Uppg 1a
==========
När vi gör x*y med x<sqrt(p) så måste y>sqrt(p) om x*y=p. Dvs i ett par x,y så p
"litet" x med ett "stort" y och när vi nått upp till sqrt(p) för x så har vi hit

Uppg 1b
==========
Inkapsling och Informationsdöljande se sid 418- i boken.
Detta ger fördelen att man tex kan döpa om variabler eller ändra hur man lagrar
värde utan att klienten påverkas.
Man kan också styra vad klienten tillåts "se" och manipulera genom att bara bear
värden via getters/setters.

(De flesta ger den andra anledningen, ofta i flera varianter. Väldigt få ger den

Uppg 1c
==========

sum = 9.999999999999831 efter 1000 termer
(exakta siffror på slutet är inte väsentligt utan just att det blir < 10)

Alla tal kan inte representeras exakt i datorn, precis som 1/3 inte är exakt lik
0.33333 så kan inte 0.01 representeras exakt utan lagras som ett lite mindre tal
(precis som 0.33333 är lite mindre än 1/3).
Summan blir därför lite för liten.

Uppg 1d
==========
Tillstånd och beteende.
Metoder och instansvariabler accepterades också.

Uppg 2
==========
se separat fil

page breake
Uppg 3
====================

```
public class Dice {
    private static int nbrOfdice = 0; // Nbr of dice created
    private int dice;               // Number showing on the first die.
    private int size = 6;           // default size
    private int whoAmI = 0;         // this dice id i.e. it's creation nbr

    public Dice(int size) {
        if ( size>=2 && size<=20 ) {
            this.size = size;
            nbrOfdice = nbrOfdice + 1;
            whoAmI = nbrOfdice;
            roll(); // initial value, don't care about the return
        } else {
            throw new IllegalArgumentException("Dice: "
                + "Otillåten storlek, måste vara 2..20)");
        }
    }

    public int roll() {
        dice = (int)(Math.random()*size) + 1;
        return dice;
    }
    public int getValue() {
        return dice;
    }
    public int getSize() {
        return size;
    }
    public int getWhoAmI() {
        return whoAmI;
    }
    public String toString() {
        return "I am Dice nbr " + whoAmI + " and my value is =" + dice;
    }
} // end class Dice
```

Uppg 4
====================

```
public class MainDiceArray {
    public static void main(String[] args) {
        Dice[] df = new Dice[3]; //1
        int[] size = {6,5,7};  //2
        for (int i=0; i<df.length; i++) {
            df[i] = new Dice(size[i]);
```

```
        }
        for (int i=0; i<df.length; i++) {
            System.out.print(df[i]);
            System.out.println(" size= " + df[i].getSize());
        }
    } // end main()
} // end class MainDiceArray
```