

## Kontrollskrivning 2, Programmeringsteknik för D/C

**2014–12–01, 8.00–11.00**

*Anvisningar:* fyll i omslaget fullständigt, även lösta uppgifter och antalet inlämnade blad. Skriv bara på en sida av varje papper. Lämna bara in dina lösningar och omslaget, inte skrivningen och inte några kladdpapper. Tillåtet hjälpmedel: Java-snabbreferens.

*Preliminärt* ger uppgifterna  $22 + 12 + 6 = 40$  poäng.

Jag meddelar på kurshemsidan när rättningen är klar och sätter upp en lista med poäng på anslagstavlan. Senare blir det visning av skrivningen.

---

1. En gles vektor är en vektor där de flesta elementen har värdet 0. Exempel på en gles vektor med 16 element varav 4 är skilda från 0:

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
0	0	13	0	0	3	7	0	0	0	2	0	0	0	0	0

Om man använder en vanlig vektor av typen `int []` för att lagra elementen så tar alla elementen plats, även nollorna. Om vektorn är stor och antalet nollor är stort så slösar man med minnesutrymme. Man kan i stället lagra bara de värden som är skilda från 0 i en lista, tillsammans med värdenas index:



Figur 1: Lista med alla vektorelement vars värde är skilt från 0.

Ett element i en sådan lista beskrivs av den färdigskrivna klassen `Element`:

```
/** Skapar ett element med givet index och värde */
Element(int index, int value);

/** Returnerar elementets värde */
int getValue();

/** Sätter elementets värde till value */
void setValue(int value);

/** Returnerar elementets index */
int getIndex();
```

Klassen `SparseVector` beskriver en gles vektor som kan innehålla naturliga heltal, alltså tal från 0 och uppåt.

```
/** Skapar en gles vektor med storleken size där alla elementen är 0 */
SparseVector(int size);

/** Returnerar vektorns storlek */
int getSize();

/** Returnerar värdet med index i. Om i ligger utanför det tillåtna
    intervallet [0, size-1] returneras -1 */
int get(int i);

/** Sätter in värdet value med index i. Om i ligger utanför det tillåtna
    intervallet ska ingenting sättas in. Returnerar true om insättningen
    gick bra (dvs. i har ett tillåtet värde), annars false */
boolean set(int i, int value);
```

Implementera klassen `SparseVector`. Anvisningar:

- Använd en `ArrayList` för att lagra de element vars värden är skilda från 0 (se figur 1).
- Listan med element behöver inte vara sorterad på något sätt.

2. En tärning beskrivs av följande färdigskrivna klass (för den vanligaste sortens tärning är `sides = 6`):

```

/* Skapar en tärning med sides sidor, sides >= 1 */
public Die(int sides);

/** Kastar tärningen, returnerar antalet prickar (1..sides) */
public int roll();

/** Returnerar antalet sidor */
public int getSides();

```

- a) Skriv en metod som kastar *två* tärningar  $n$  gånger och returnerar en vektor med antalet gånger som summan av tärningarnas prickar blev 2, 3, ... Om två sexsidiga tärningar kastas 1000 gånger kan resultatet till exempel bli en vektor med elementen 29 54 69 115 112 170 150 129 72 69 31 (summan blev 2 29 gånger, 3 54 gånger, ..., 12 31 gånger). Metoden ska ha följande rubrik:

```
private static int[] countRolls(Die d1, Die d2, int n);
```

Metoden ska fungera för tärningar med godtyckligt antal sidor.

- b) Skriv en main-metod som kastar två sexsidiga tärningar 1000 gånger och på en rad skriver ut antalet gånger som summan blev 2, 3, ..., 12. Du ska naturligtvis utnyttja metoden `countRolls`.

3. Betrakta följande klass:

```

public class CharProgram {
    public static void swapChars1(char[] a, int i, int j) {
        char temp = a[i];
        a[i] = a[j];
        a[j] = temp;
    }

    public static void swapChars2(char c1, char c2) {
        char temp = c1;
        c1 = c2;
        c2 = temp;
    }

    public static void main(String[] args) {
        char[] letters = new char[26];
        for (int i = 0; i < letters.length; i++) {
            letters[i] = (char) ('a' + i);
        }
        swapChars1(letters, 0, 1);
        System.out.println(letters[0] + " " + letters[1]);
        swapChars2(letters[2], letters[3]);
        System.out.println(letters[2] + " " + letters[3]);
    }
}

```

Vad skrivs ut när programmet exekveras? Förklara kortfattat vad som händer.