


Programmeren 1 31 augustus 2011 Prof. T. Schrijvers



Instructies

Schrijf al je antwoorden op deze vragenbladen (op de plaats die daarvoor is voorzien). Geef ook je kladbladen af. Bij heel wat vragen moet je zelf Java-code schrijven. Hou dit *kort en bondig*. Je hoeft geen commentaarregels te schrijven en ook eventuele `import`-opdrachten zijn niet nodig.

Pas op voor subtiele verschillen in de vraagstelling t.o.v. januari.

Veel succes!

Vraag 1: Beeldverwerking (7pt)

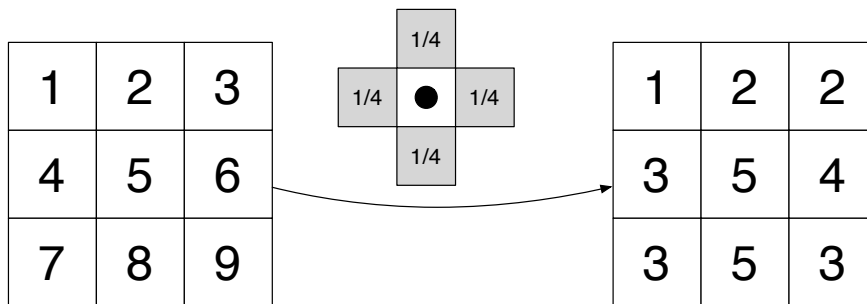
Een **Beeld** is een $n \times m$ matrix van pixels (beeldpunten). Elke pixel wordt voorgesteld door zijn kleur, een geheel getal in het bereik 0–255.

Beelden die verkregen worden via digitale fotoestellen, camera's, scanners ... zijn vaak niet zonder meer geschikt voor gebruik. Ze moeten eerst één of meer bewerkingen ondergaan om het beeld scherper of net vager te maken, om de kleuren beter uit te laten komen, ...

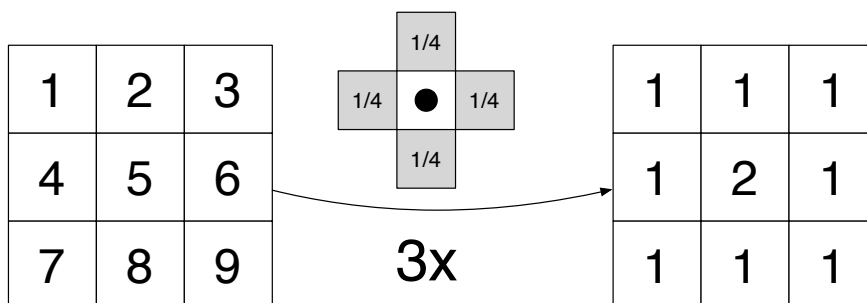
Zo'n bewerking wordt uitgevoerd door een **Transformator**. In een transformatie neemt een **Transformator** een afbeelding en produceert een nieuwe afbeelding die aangepast is ten opzichte van het origineel.

Een **StempelTransformator** is een bepaalde van soort van **Transformator** die de nieuwe waarde van elke pixel bepaalt in functie van de originele waarden van de pixels in zijn omgeving (en zichzelf). De pixels aan de rand van het beeld hebben geen volledige omgeving, daarom gaan we ervan uit dat de ontbrekende burens de waarde 0 hebben.

Een voorbeeld hiervan is de **Vervager**. Deze stelt de nieuwe waarde van een pixel gelijk aan de gemiddelde waarde van de vier burens van die pixel.



De **HerhaalTransformator** is een **Transformator** gedefinieerd in functie van een andere transformator: zijn transformatie bestaat uit l opeenvolgende oproepen van de onderliggende transformator. Hiermee kan bijvoorbeeld de vervager vier maal na elkaar opgeroepen worden.



Opgave

1. Implementeer de klasse `Beeld` en voorzie volgende methodes:
 - een constructor die een $n \times m$ beeld maakt waarvan alle pixels de waarde 0 hebben,
 - `getPixel` en `setPixel` om de waarde van een pixel op te vragen en in te stellen, en
 - `getN` en `getM` om de grootte van het beeld op te vragen.
2. Maak abstracte klasse `Transformator` met een abstracte methode `transformeer` die een origineel beeld als parameter neemt en een nieuw beeld teruggeeft.
3. Maak subklasse `HerhaalTransformator`. Voorzie een gepaste constructor. De methode `transformeer` herhaalt l maal de transformatie van een andere transformator.
4. Maak een abstracte subklasse `StempelTransformator`. Implementeer de methode `transformeer` zodat die elke pixel overloopt en telkens de hulpmethode `stempel` oproept:

```
protected abstract int stempel(int x, int y, Beeld in)
```

Deze hulpmethode moet door concrete subklassen ingevuld worden. Voorzie een bijkomende concrete hulpmethode `protected int get(int x, int y, Beeld beeld)` die 0 teruggeeft als (x,y) buiten het beeld valt en anders de pixelwaarde binnen het beeld.

5. Maak een subklasse `Vervager` van `StempelTransformator` waarbij `stempel` de gemiddelde waarde van de vier burens (noord, oost, zuid, west) teruggeeft.
6. Maak een klasse `Main` met een `main` methode die het 3×3 beeld uit het voorbeeld maakt (met waarden $1, \dots, 9$) en hierop de 3-maal herhaalde `Vervager` op toepast met behulp een `HerhaalTransformator`.

...

Vraag 2: Methode-oproepen

(1,5pt)

<pre>public class A { public int m() { return 2; } public int n() { return 7 * m(); } }</pre>	<pre>public abstract class B extends A { public int m() { return super.m() * n(); } }</pre>	<pre>public class C extends B { public int m() { return 5; } }</pre>
-------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------

Gegeven bovenstaande klassedefinities, telkens met een lege constructor die niet is weergegeven, geef aan wat het resultaat is van onderstaande expressies:

(a)	new A().n()	
(b)	new B().n()	
(c)	new C().n()	

Vraag 3: Waarde

(1pt)

```
int x = 4;
try {
    x += 2;
    throw new ???("something is wrong");
} catch (RuntimeException e) {
    x++;
    throw new IOException("still wrong");
} catch (Exception e) {
    x = 11 % x;
}
```

Geef aan welke waarde zich in de variable x bevindt na het uitvoeren van deze code, als we ??? vervangen door een van onderstaande woorden:

(a)	NullPointerException	
(b)	IOException	

Vraag 4: Hamburgerwinst**(3pt)**

Schrijf een methode met signatuur

```
public int[] winst(int[] [] omzet, int[] [] marge)
```

De twee-dimensionale array `omzet` houdt bij voor elke vestiging van de hamburgerketen *Slow* hoeveel hamburgers er van elke soort verkocht zijn in een maand. Dus `omzet[i][j]` is het aantal hamburgers van soort `j` dat vestiging `i` verkocht heeft.

De winstmarge op de hamburgers hangt af van de omzet. Zo bepaalt `marge[j][k]` hoeveel de marge is per hamburger van soort `j` als er tussen `k * 50` en `k * 50 + 49` van verkocht zijn door een vestiging.

De methode `winst` geeft een array terug met de maandwinst per vestiging. Bijvoorbeeld (bij andere oproepen kunnen de arrays groter zijn!):

	H1	H2		0-49	50-99		
V1	10	70	H1	1	2	V1	290
V2	60	20	H2	3	4	V2	180
	omzet			marge			winst

Vraag 5: Arrays**(1,5pt)**Hoeveel verschillende int waarden passen in geneste array `arr` na uitvoering van de code?

(a)	<code>int[] [] arr = new int[0][2];</code>	
(b)	<code>int[] [] arr = new int[3] []; for (int i = 0; i < arr.length; i++) { arr[i] = new int[3-i]; }</code>	
(c)	<code>int[] [] arr = new int[3] []; int[] rij = new int[3]; for (int i = 0; i < arr.length; i++) { arr[i] = rij; }</code>	

Vraag 6: Herschrijven**(3pt)**

Herschrijf onderstaande methodes zodat: (a) lijsten ipv. arrays gebruikt, (b) een for-each ipv. for lus gebruikt, en (c) een array ipv. een lijst gebruikt.

```
(a) public void m1(int[] k, int[] l) {
    for (int i = 0; i < k.length; i++) {
        k[i] = l[i];
    }
}
```

```
(b) public void m2(List<String> s) {
    for (int i = 0; i < s.size(); i++) {
        System.out.println(s.get(i));
    }
}
```

```
(c) public void m3(List<String> l) {
    for (int i = 0; i < l.size(); i++) {
        l.set(i, "hello");
    }
}
```

Vraag 7: Histogram**(3pt)**

Schrijf een methode met signatuur

```
public void histogram(String tekst)
```

De string `tekst` bestaat uit een aantal woorden gescheiden door spaties. De methode drukt voor elk verschillend woord in `tekst` af hoe vaak het voorkomt.

Bijvoorbeeld, deze oproep:

```
histogram("het is frans zei frans in het frans");
```

drukt volgende af:

```
het: 2  
is: 1  
frans: 3  
zei: 1  
in: 1
```

De volgorde waarin de woorden afgedrukt worden is niet belangrijk.

Maak gebruik van een map. Tip: je kan de verzameling (set) van sleutels van een map opvragen met de methode `keySet()`.

Einde