Programmeren 1 23 januari 2013 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 de addertjes onder het gras.
Veel succes!

Vraag 1: ASCII Art (7pt)

ASCII art is het kunstig gebruik van gedrukte karakters voor het vormen van figuren en afbeeldingen. In deze opgave ontwikkelen we een kleine bibliotheek om een eenvoudige vorm van tekstuele figuren te ondersteunen, zoals bijvoorbeeld:

```
++++----++++----++++----
++++----++++----++++----
++++----++++----++++----
----++++----++++----++++
----++++----++++----++++
----++++----++++----++++
```

BEPERKING: Nergens in je oplossing mag een array, lijst of andere collectie van karakters voorkomen.

1. Implementeer de algemene klasse `TextShape` die een rechthoekige tekstfiguur voorstelt.

   - Voorzie twee abstracte methodes

     ```
     public abstract int getWidth()

     public abstract int getHeight()
     ```

     die de breedte en hoogte van de figuur (in aantal karakters) teruggeven.
   - Voorzie de abstracte methode

     ```
     public abstract char getChar(int i, int j)
     ```

     die het karakter op kolom i, rij j teruggeeft. We beginnen te tellen bij 0 en van links boven naar rechts onder.
   - Voorzie een concrete methode

     ```
     public void print()
     ```

     die de figuur afdrukt op de terminal.

2. Implementeer de subklasse `CharBox` die een $w \times h$ figuur voorstelt, opgevuld met een karakter c. Voorzie een constructor zodat `new CharBox(4,3,'+').print()` dit effect heeft:

```
++++
++++
++++
```

3. Implementeer de subklasse `HorizontalComposer` die twee figuren horizontaal aan elkaar plakt. Je mag ervan uitgaan dat de twee figuren dezelfde hoogte hebben. Voorzie een constructor zodat

   ```
   new HorizontalComposer(new CharBox(4,3,'+'),new CharBox(5,3,'-')).print()
   ```

   dit effect heeft:

   ```
   ++++-----
   ++++-----
   ++++-----
   ```

4. Implementeer de subklasse `Transposer` die een gegeven figuur transponeert. Voorzie een constructor zodat `new Transposer(new CharBox(4,2,'-')).print()` dit effect heeft:

   ```
   --
   --
   --
   --
   ```

5. Maak een klasse `Main` met een `main`-methode die deze figuur opbouwt als een `TextShape` en dan afdrukt. Zet niet al je code op één regel en maak zo weinig mogelijk objecten aan.

   ```
   ++++----++++----++++----
   ++++----++++----++++----
   ++++----++++----++++----
   ----++++----++++----++++
   ----++++----++++----++++
   ----++++----++++----++++
   ```

```
public abstract class TextShape {

  public abstract int getWidth();

  public abstract int getHeight();

  public abstract char getChar(int x, int y);

  public void print() {
    for (int i = 0; i < getHeight(); i++) {
      for (int j = 0; j < getWidth(); j++) {
        System.out.print(getChar(i,j));
      }
      System.out.println();
    }
  }

}
```

```
public class CharBox extends TextShape {

  private final int height;
  private final int width;
  private final char character;

  public CharBox(int height, int width, char character) {
    this.height = height;
    this.width = width;
    this.character = character;
  }

  public int getHeight() {
    return height;
  }

  public int getWidth() {
    return width;
  }

  public char getChar(int x, int y) {
    return character;
  }

}
```

```
public class HorizontalComposer extends TextShape {
```

```
  private final TextShape left;
  private final TextShape right;

  public HorizontalComposer(TextShape left, TextShape right) {
    this.left = left;
    this.right = right;
  }

  public int getHeight() {
    return left.getHeight();
  }

  public int getWidth() {
    return left.getWidth() + right.getWidth();
  }

  public char getChar(int i, int j) {
    if (j < left.getWidth()) {
      return left.getChar(i,j);
    } else {
      return right.getChar(i, j  - left.getWidth());
    }
  }

}
```

```
public class Transposer extends TextShape {

  private final TextShape shape;

  public Transposer(TextShape shape) {
    this.shape = shape;
  }

  public int getHeight() {
    return shape.getWidth();
  }

  public int getWidth() {
    return shape.getHeight();
  }

  public char getChar(int x, int y) {
    return shape.getChar(y,x);
  }

}
```

```
public class Main {

  public static void main(String[] args) {
    TextShape plus   = new CharBox(4,3,'+');
    TextShape minus  = new CharBox(4,3,'-');
    TextShape pm     = new HorizontalComposer(plus,minus);
    TextShape mp     = new HorizontalComposer(minus,plus);
    TextShape tpm    = new Transposer(pm);
    TextShape tmp    = new Transposer(mp);
    TextShape block  = new HorizontalComposer(tpm,tmp);
    TextShape block2 = new HorizontalComposer(block,block);
    TextShape block3 = new HorizontalComposer(block2,block);
    block3.print();
  }

}
```

Vraag 2: Methode-oproepen **(2pt)**

```
public class A {
  public A() {}
  public int n() {
    return m() * 2;
  }
  public int m() {
    return 3;
  }
}
```

```
public abstract class B extends A {
  public B() {}
  public int n() {
    return 5;
  }
  public int m() {
    return n() * 2;
  }
}
```

```
public class C extends A {
  public C() {}
  public int m() {
    return 7;
  }
}
```

```
public class D {
  public D() {}
  public int size() {
    return new List<Object>().size();
  }
}
```

Gegeven bovenstaande klassendefinities, geef aan wat het resultaat is van onderstaande expressies:

(a)	`new A().n()`	2 * 3 = 6
(b)	`new B().n()`	compilererror (instantie van een abstracte klasse)
(c)	`new C().n()`	2 * 7 = 14
(d)	`new D().size()`	compilererror (instantie van een interface)

Vraag 3: Exceptions **(1pt)**

```
public int m(int x) throws Exception {
  if (x % 2 == 0) {
    throw new IOException("something is wrong");
  } else {
    throw new IllegalArgumentException("bad argument");
  }
}
public void n(int x) throws Exception {
  try {
    x = 3 + m(x);
  } catch (IOException e1) {
      try {
        x += 2 + m(x + 1);
      } catch (RuntimeException e2) {
          x++;
      }
  }
  System.out.println(x);
}
```

Welke waarde wordt afgedrukt bij volgende oproepen?

(a)	`n(42)`	43
(b)	`n(43)`	niets (niet opgevangen IllegalArgumentException)

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

Schrijf een methode met signatuur

```
public int[][] slang(int n, int m)
```

De methode geeft een geneste array terug die een `n`×`m` matrix voorstelt. De matrix is opgevuld in een slangpatroon met de getallen van 1 tot `n`·`m`. **Afspraak:** de rijen van de matrix zijn de binnenste arrays van de geneste array.

Hieronder zie je het resultaat voor $n = 5$ en $m = 4$.

$$\begin{bmatrix} 1 \to 2 \to 3 \to 4 \\ \qquad\qquad\quad \downarrow \\ 8 \leftarrow 7 \leftarrow 6 \leftarrow 5 \\ \downarrow \\ 9 \to 10 \to 11 \to 12 \\ \qquad\qquad\quad \downarrow \\ 16 \leftarrow 15 \leftarrow 14 \leftarrow 13 \\ \downarrow \\ 17 \to 18 \to 19 \to 20 \end{bmatrix}$$

```
public int[][] slang(int n, int m) {
  int[][] result = new int[n][m];

  for (int i = 0; i < n; i++) {
    if (i % 2 == 0) {
      for (int j = 0; j < m; j++) {
        result[i][j] = (i * m) + (j + 1);
      }
    } else {
      for (int j = 0; j < m; j++) {
        result[i][j] = (i * m) + (m - j);
      }
    }
  }

  return result;
}
```

Vraag 5: Arrays **(1pt)**

Hoeveel verschillende `int` waarden passen in geneste array `arr` na uitvoering van de code?

(a)	`int[][] arr = new int[2][3];`	$2 \times 3 = 6$

(b)	`int[][] arr = new int[5][];`
	`for (int i = 0; i < arr.length; i++) {`
	`  arr[i] = new int[i * (5 - i)];`
	`}`
		$\sum_{i=0}^{4} i \times (5 - i) = 20$

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

Herschrijf onderstaande methodes zodat: (a) lijsten i.p.v. arrays gebruikt, (b) een iterator i.p.v. een for-each lus gebruikt, en (c) een while-lus met index i.p.v. for-each lus gebruikt.

(a)
```
public void m1(int[] k, int[] l) {
  for (int i = k.length; i > 0; i--) {
    k[k.length - i] = l[i-1];
  }
}
```

```
public void m1(List<Integer> k, List<Integer> l) {
  for (int i = k.size(); i > 0; i--) {
    k.set(k.size()- i, l.get(i-1));
  }
}
```

(b)
```
public void m2(Set<String> set) {
  for (String el: set) {
    System.out.println(el);
  }
}
```

```
public void m2(Set<String> set) {
  Iterator<String> it = set.iterator();
  while (it.hasNext()) {
    System.out.println(it.next());
  }
}
```

(c)
```
public int m3(int[] a) {
  int sum = 0;
  for (int x : a) {
    sum += x;
  }
  return sum;
}
```

```
public int m3(int[] a) {
  int sum = 0;
  int index = 0;
  while (index < a.length) {
    sum += a[index];
    index++;
  }
  return sum;
}
```

Vraag 7: Skyfall 2 **(3pt)**

Om je geheime berichten te versleutelen verwissel je de letters van plaats volgens een gegeven permutatieschema. Bijvoorbeeld, met het permutatieschema:

$$\{0 \mapsto 8, 1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 3, 4 \mapsto 9, 5 \mapsto 5, 6 \mapsto 0, 7 \mapsto 7, 8 \mapsto 6, 9 \mapsto 4\}$$

wordt de boodschap `"NAMED JOBS"` omgezet in `"JAMES BOND"`.

Schrijf een methode met de volgende signatuur om deze ontcijfering uit te voeren en weg te schrijven naar het bestand met gegeven naam.

```
public void permuteer(String boodschap, int[] schema, String bestand)
```

Hierbij geeft `schema[i]` aan op welke plaats letter `i` moet komen. Bovenstaande voorbeeld wordt bekomen met de oproep

```
permuteer("NAMED JOBS",new int[]{8, 1, 2, 3, 9, 5, 0, 7, 6, 4},"forurisonly.txt")
```

```
public void permuteer(String boodschap, int[] schema, String bestand) {
  char[] permutatie = new char[schema.length];
  for (int i = 0; i < schema.length; i++) {
    permutatie[schema[i]] = boodschap.charAt(i);
  }
  try {
    PrintWriter pw = new PrintWriter(new BufferedWriter(new FileWriter(bestand)));
    pw.print(new String(permutatie));
    pw.close();
  } catch (IOException e) {
    System.out.println("Fout bij het schrijven van de boodschap!");
  }
}
```

Einde