Exercise NMCGJ: Drawing Text Patterns



We want to make a program that generates some simple text graphics (patterns) and draws them to the console and to a text editor.

Problem

Make the class TextPattern that draws simple patterns in text. In particular, it should provide the following methods to generate a triangle, square and pyramid of a given height:

```
int height = 5;
printTriangle(height);
х
ΧХ
XXX
XXXX
XXXXX
printSquare(height);
XXXXX
XXXXX
XXXXX
XXXXX
XXXXX
printPyramid(height);
    Х
   XXX
  XXXXX
 XXXXXXX
XXXXXXXXX
```

When calling such a method, the corresponding pattern should be printed both to the console and to a text editor. For the latter, the class TextFrame can be used.

The class TextFrame

The class TextFrame provides a text editor for simple text manipulation. It inherits from the class JFrame which is part of the package javax.swing. This package offers an extensive set of components for developing graphical user interfaces (GUIs), and JFrame is the component for creating a main window (frame). The following methods are available in the class TextFrame:

start()	Visualizes the text editor.
close()	Closes the text editor.
setMenuVisible(boolean visible)	Indicates whether the menu bar is visible or not.
setTextEditable(boolean editable)	Indicates whether the text field is editable or not.
setTextScrollable(boolean scrollable)	Indicates whether the text field is scrollable or not.
getText(): String	Gets the content of the text field.
setText(String text)	Sets the content of the text field.
addText(String text)	Adds content to the text field.
clearText()	Clears the text field.
loadTextFile(File file)	Loads the content from a file to the text field.
saveTextFile(File file)	Saves the content from the text field to a file.

A demo is given in the main method.

Implementation

Implementation of TextPattern (preliminary version)

We first build a simplified version of the class TextPattern that is able to print the desired text patterns to the console.

Create a class called TextPattern. Use the skeleton below (preliminary version). Only the main method is provided to test the class: it prints three text patterns (a triangle, a square, and a pyramid) to the console. Define and implement the missing methods.

Implementation of TextPrinter

Using an interface can help us

- to separate the generation of text patterns from the printing functionalities, and
- to synchronize the printing to both the console and the text editor.

The interface TextPrinter is defined below and ensures sufficient printing functionality for our purposes. This interface will help in writing the full version of TextPattern. Create two classes that implement this interface:

ConsolePrinter	Takes care of printing a given text to the console.
TextFramePrinter	Takes care of printing a given text to a certain text editor. The constructor requires as input an object of the class TextFrame.

Implementation of TextPattern (full version)

We are now ready to build a more neat and more general class TextPattern for printing the desired text patterns to different destinations. Thanks to the interface TextPrinter, we do not need to take care of the printing destination/functionality anymore and we can just work with a collection of objects satisfying the TextPrinter interface. The class TextPattern should hold an array of type TextPrinter and provide methods to add new printing destinations. Of course, the class should also provide methods to generate the text patterns and print them to all the assigned printing destinations.

Create a class called TextPattern. Use the skeleton below (full version). Only the main method is provided to test the class: it creates two printing destinations (the console and a text editor), and it prints three text patterns (a triangle, a square, and a pyramid). Define and implement the missing methods.

Notes

- To produce a new line in a String, you can use the newline character "\n".
- Do NOT modify the class TextFrame and the main method of TextPattern.

Java classes

TextPrinter.java

```
public interface TextPrinter {
    public void print(String text);
    public void println(String text);
    public void println();
}
```

TextPattern.java (preliminary version)

```
public class TextPattern {
   // Add the missing implementation to this class!
   // A small test of the TextPattern class.
   public static void main(String[] args) {
      // Print some text patterns to the console
      TextPattern pattern = new TextPattern();
      pattern.println("a triangle:");
      pattern.printTriangle(5);
      pattern.println();
      pattern.println("a square:");
      pattern.printSquare(5);
      pattern.println();
      pattern.println("a pyramid:");
      pattern.printPyramid(5);
   }
}
```

TextPattern.java (full version)

```
public class TextPattern {
   // Add the missing implementation to this class!
   // A small test of the TextPattern class.
   public static void main(String[] args) {
      // Construct the text frame
      TextFrame textFrame = new TextFrame("My Text Pattern Frame");
      textFrame.setTextEditable(false);
      textFrame.start();
      // Construct the printers
      ConsolePrinter consolePrinter = new ConsolePrinter();
      TextFramePrinter framePrinter
                               = new TextFramePrinter(textFrame);
      // Construct a text pattern generator and add its printers
      TextPattern pattern = new TextPattern();
      pattern.addPrinter(consolePrinter);
      pattern.addPrinter(framePrinter);
      // Print some text patterns
      pattern.println("a triangle:");
      pattern.printTriangle(5);
      pattern.println();
      pattern.println("a square:");
      pattern.printSquare(5);
      pattern.println();
      pattern.println("a pyramid:");
      pattern.printPyramid(5);
   }
}
```