

Exercice 29 - Structures répétitives imbriquées

Créez la classe `Rectangle` avec les attributs entiers positifs `width` (largeur) et `height` (hauteur) et définissez un constructeur permettant d'initialiser les attributs (prenez la valeur absolue des données fournies).

- a) Définissez la méthode `void draw()` qui dessine un rectangle à l'aide d'étoiles.

Exemple pour `width=10` et `height=4` :

```
*****
*****
*****
*****
```

- b) Définissez la méthode `void drawBorder()` qui dessine un rectangle en dessinant uniquement le bord à l'aide d'étoiles.

Exemple pour `width=10` et `height=4` :

```
*****
*           *
*           *
*****
```

- c) Définissez la méthode `void drawNumbers()` qui "dessine" un rectangle à l'aide de nombres.

Exemple pour `width=10` et `height=4` :

```
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 9 9 10
```

- d1) Définissez la méthode `void drawShifted1()` qui imprime des nombres décalés.

Exemple pour `width=5` et `height=4` :

```
1 2 3 4 5
  1 2 3 4 5
    1 2 3 4 5
      1 2 3 4 5
```

Aide : il faut 3 boucles...

- d2) Définissez la méthode `void drawShifted2()` qui imprime des nombres décalés et coupés.

Exemple pour `width=5` et `height=4` :

```
1 2 3 4 5
  1 2 3 4
    1 2 3
      1 2
```

Aide : il faut 3 boucles...

- d3) Définissez la méthode `void drawShifted()` qui imprime des nombres décalés totalement.

Exemple pour `width=5` et `height=4` :

```
1 2 3 4 5
5 1 2 3 4
4 5 1 2 3
3 4 5 1 2
```

Aide : il faut 3 boucles...

Rectangle	
-	<code>width : int</code>
-	<code>height : int</code>
+	<code>Rectangle(pWidth : int, pHeight : int)</code>
+	<code>draw() : void</code>
+	<code>drawBorder() : void</code>
+	<code>drawNumbers() : void</code>
+	<code>drawShifted1() : void</code>
+	<code>drawShifted2() : void</code>
+	<code>drawShifted() : void</code>
+	<code>drawIncrease() : void</code>
+	<code>drawDecrease() : void</code>
+	<code>drawSlope() : void</code>

e) Définissez la méthode `void drawIncrease()` qui imprime des nombres de plus en plus.

Exemple pour `height = 4` (**width** n'est pas utilisé) :

```
1
1 2
1 2 3
1 2 3 4
```

f) Définissez la méthode `void drawDecrease()` qui imprime des nombres de moins en moins.

Exemple pour `height = 4` (**width** n'est pas utilisé) :

```
4
4 3
4 3 2
4 3 2 1
```

g) Définissez la méthode `void drawSlope()` qui imprime des nombres en pente.

Exemple pour `height = 4` (**width** n'est pas utilisé) :

```
1
2 1
3 2 1
4 3 2 1
```

Notions requises : classe, objet, méthode, paramètre, type, attribut, void, constructeur
Structures requises : if, (for || while) imbrication de boucles

Exercice 30 : Figures d'étoiles

Écrivez la classe `stars` qui permet de dessiner différentes figures en employant le caractère '*'. (P.ex. en 'dessinant' 4 lignes contenant chacune 4 étoiles, on obtient un carré de côté 4.). On suppose que toutes les données fournies comme paramètres sont positives.

Stars
+ drawRectangle(pWidth : int, pHeight : int) : void
+ drawSquare(pSideLength : int) : void
+ drawPyramide(pHeight : int) : void
+ drawReversePyramide(pHeight : int) : void
+ drawDiamond(pHeight : int) : void

Exemple de *pyramide* pour **pHeight** = 3

```
*
***
*****
```

Exemple de *reversePyramide* pour **pHeight** = 3

```
*****
***
*
```

Exemple de *diamond* pour **pHeight** = 3

```
*
***
*****
***
*
```

Notions requises : classe, objet, méthode, paramètre, type, attribut, void, constructeur
Structures requises : if, (for || while) imbrication de boucles