Cours 420-201 RE Introduction à la programmation Automne 2020 Cégep Limoilou Département d'Informatique	Laboratoire 3 Boucles, méthodes et conditions Programmer plusieurs méthodes et faire de la validation (12% de la session)

Objectifs:

- Créer des algorithmes impliquant des conditions et des boucles
- Programmer des méthodes java avec boucles et conditions
- Analyser du code Objet
- Développer son autonomie en cherchant de l'information sur le Web

Contexte:

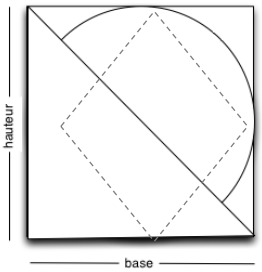
- En équipe de 2 (maximum)
- Vous avez 3 semaines pour compléter ce laboratoire.
- Pour commencer votre travail, prendre le projet disponible sur Léa dans le dossier du TP3.

À remettre:

- Vous remettez votre **dossier de projet** au complet en format compressé (.zip) dans un fichier du nom « vos noms – Tp 3.zip ».

Contexte

Dans le package « dessins_de_formes » vous allez compléter la classe DessinGeometrique qui permet de créer plusieurs formes géométriques (ligne, rectangle, carré, triangle, losange et *peut être le cercle*) sous forme de chaînes de caractères. La classe possède 2 attributs: base et hauteur. Ces 2 attributs constituent un **cadre** qui gère l'affichage de chaque forme. Toutes les formes vont avoir une hauteur (nombre de lignes) correspondante à la variable d'instance *hauteur* (propriété d'objet) et une largeur (nombre de colonnes) correspondante à la variable d'instance *base*.



Attention, un carré ou un cercle doit avoir une hauteur égale à sa base. Si deux valeurs différentes sont fournies pour ces attributs, le programme sélectionnera automatiquement la plus petite des 2 valeurs, pour la base et la hauteur, **au moment de créer les formes**.

La classe DessinGeometrique vous est fournie. Plusieurs méthodes sont déjà complétées. Vous devez compléter les autres méthodes en suivant les étapes ci-dessous.

La classe Application contient les méthodes main et demanderCadre. Elle crée et utilise des instances de DessinGeometrique. La classe Application a déjà été programmée, vous devez la faire exécuter pour constater votre avancement et déboguer vos algorithmes. **Si vous exécutez cette classe avec le code initial reçu, le programme fonctionnera, mais aucune forme ne sera affichée (dessinée).** Lorsque vous aurez complété le code de la classe **DessinGeometrique**, les différentes formes devraient apparaître dans la console (voir affichage à la fin du document).

Étape 1 : programmer et tester une méthode de validation et getPlusPetitCote()

La classe DessinGeometrique possède une méthode **validerCadre**() qui permet de valider si la base et la hauteur sont utilisables. Vous remarquerez que le constructeur fait appel à cette méthode. La version de cette méthode que vous avez reçue ne fait que retourner vrai. Vous allez la modifier pour ne retourner vrai que si les 2 variables d'instances sont supérieures à 0 et inférieures ou égales aux constantes BASE_MAX et HAUTEUR_MAX. Dans tous les autres cas, elle doit retourner faux.

Faites les tests de cette méthode dans la classe **DessinGeometriqueTest** que vous allez créer dans le package **tests**.

Étape 2 : Boucles : programmer la création des formes géométriques.

a) **Créer des lignes**:

La classe DessinGeometrique possède 2 méthodes pour créer des lignes, la méthode creerLigneHorizontale() qui retourne une ligne sur une seule rangée et la méthode creerLigneVerticale() qui retourne une ligne sur une seule colonne. **Les méthodes sont déjà présentes dans la classe, vous n'avez qu'à y insérer le code nécessaire pour produire une ligne d'étoiles.**

- creerLigneHorizontale() crée une ligne dont la longueur est déterminée par l'attribut **base**. Par exemple, si la base est 10, l'affichage donnera

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

- creerLigneVerticale() crée une colonne dont la longueur est déterminée par l'attribut d'instance **hauteur**. Par exemple, si la hauteur est 4, l'affichage donnera

```
*
*
*
*
```

b) **Créer un rectangle ou un carré**

La classe DessinGeometrique permet de créer des contours de rectangles et de carrés avec les méthodes gererContourCarre() et gererContourRectangle(). Ces méthodes sont déjà programmées. Ces 2 méthodes délèguent la majeure partie du travail à la méthode **creerContour**(). Cette dernière reçoit en paramètres une **base** et une **hauteur** (pBase et pHauteur). Vous devez programmer le code qui va retourner le contour de la forme correspondant aux paramètres pBase et pHauteur. Par exemple, si les paramètres sont **pBase** = 8 et **pHauteur** = 4, la forme sera:

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

TRUC: commencez par afficher un rectangle plein. Vous pourrez ensuite modifier votre code pour n'afficher que le contour.

c) Créer un triangle

La classe DessinGeometrique peut retourner un triangle appuyé à droite ou appuyé à gauche à l'aide des méthodes creerTriangleDroit() et creerTriangleGauche(). Vous devez maintenant programmer ces 2 méthodes afin d'obtenir des triangles respectant les variables d'instances **base** et **hauteur**. Par exemple, si les variables d'instance sont **base** = 8 et **hauteur** = 4, l'affichage pour la version droite sera:

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

Et pour la version gauche

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

ATTENTION, ici on veut que vous utilisiez obligatoirement des boucles « **while** » pour compléter le code des 2 méthodes

Si votre affichage varie quelque peu (erreurs d'arrondis), ce n'est pas grave tant que le triangle est du bon côté et possède le bon nombre de colonnes et de lignes.

d) Créer un losange

ATTENTION, cette méthode est un peu plus difficile. Comme pour le carré, la création du losange devra se faire avec le plus petit côté. De plus si le plus petit côté est pair vous devrez lui ajouter 1 pour qu'il devienne impair. Par exemple, si les variables d'instance sont base = 8 et hauteur = 10, l'affichage sera un losange 9 x 9:

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

Étape 3 : Boucles : programmer une méthode de saisie avec validation **sans feedback**

La classe **Application** possède une méthode demanderCadre() qui permet de créer un nouvel objet DessinGeometrique avec un nouveau cadre. La méthode que vous avez reçue demande à l'utilisateur s'il veut saisir un nouveau cadre, mais elle ne fait rien si l'utilisateur répond **oui**. Vous devez programmer la partie de la méthode où se trouve le commentaire "À faire…".

- Vous devez demander à l'utilisateur de saisir une nouvelle **base** et une nouvelle **hauteur**.
- Cependant, tant que la base et la hauteur saisies ne sont pas **valides** (voir méthode validerCadre()) elle redemande à l'utilisateur d'entrer les informations.
- Lorsque les données saisies sont valides un nouveau dessin est instancié et retourné
- La méthode main de la classe Application appelle la méthode demanderCadre() dans une boucle qui se répète tant que la méthode ne retourne pas « null » (pas d'objet DessinGeometrique). Utiliser cette dernière pour vérifier que votre code fonctionne bien.

Étape 4 : Valeurs aléatoires : consulter la JavaDoc et le Web…

Dans la classe DessinGeometrique le constructeur place des valeurs par défaut (BASE_DEFAUT et HAUTEUR_DEFAUT) lorsque la validation retourne faux. On aimerait que le constructeur utilise des valeurs aléatoires au lieu des valeurs par défaut. On vous demande d'aller voir sur le web ou API Java pour trouver comment créer ces valeurs aléatoires en Java. **INDICE**: Utiliser la méthode random(). Trouver la formule mathématique qui générera des nombres valides (supérieur à 0 et inférieur ou égal aux bornes maximales).

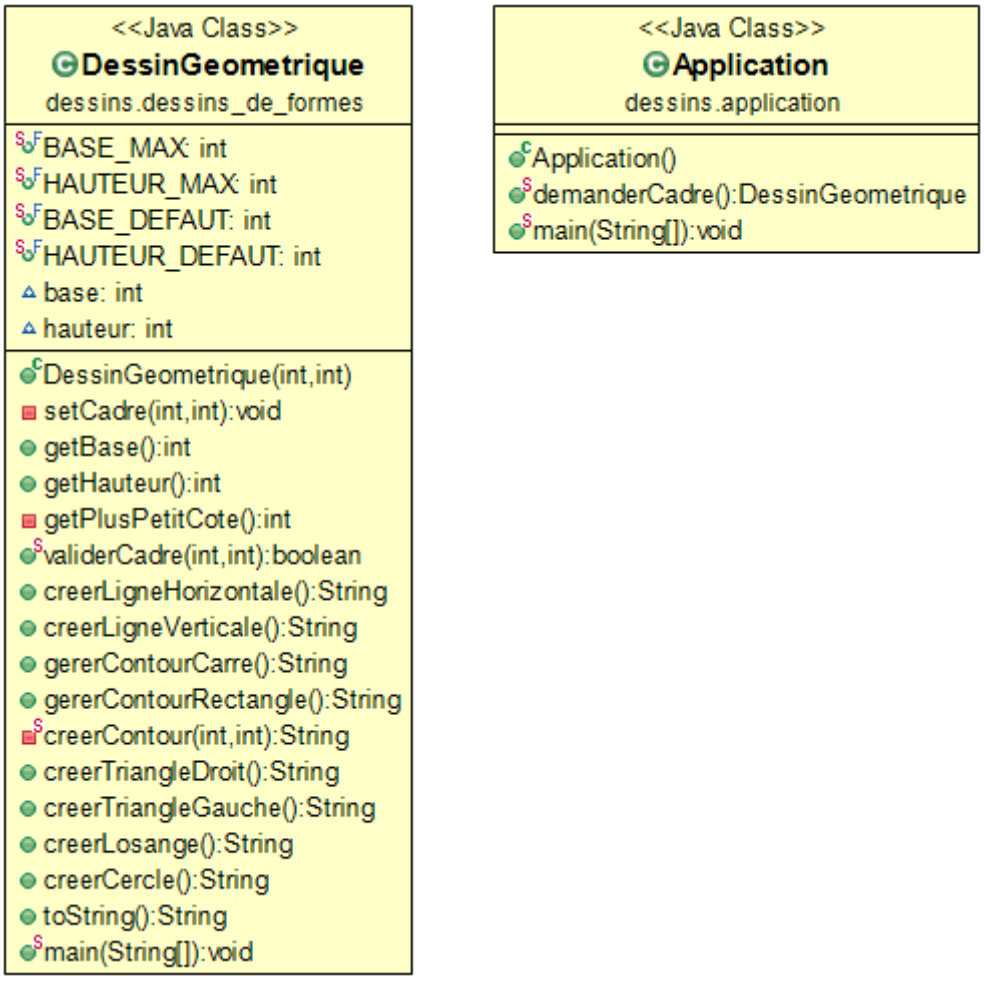
Faites les tests du constructeur dans la classe **DessinGeometriqueTest** que vous avez créée dans le package **tests**.

BONUS Étape 5 : Boucles : programmer la création d'un cercle.

ATTENTION, cette méthode est plus difficile (difficulté mathématique). Ne la faites qu'à la toute fin. Complétez la méthode creerCercle() pour retourner un cercle qui entre dans le cadre. **INDICE**: Pensez au triangle de Pythagore (R2 = b2 + h2). Si le cadre n'est pas carré, la méthode utilise 2 fois la plus petite dimension comme dit précédemment (**hauteur** ou **base**). Si le cadre est de 16 x 20 le cercle sera (16 x16), l'affichage sera:

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

Diagramme de classes :



ANNEXE 1

Affichage attendu de la méthode main de la classe Application à la fin du travail:

```
On affiche les formes avec les valeurs initiales

Ligne horizontale, 5 ligne(s) X 14 colonne(s).

**************

Ligne verticale, 5 ligne(s) X 14 colonne(s).

*
*
*
*
*

Contour carré, 5 ligne(s) X 16 colonne(s).

*****
*   *
*   *
*   *
*****


Contour rectangle, 5 ligne(s) X 16 colonne(s).

****************
*               *
*               *
*               *
****************


Triangle droit, 10 ligne(s) X 4 colonne(s).

   *
   *
   *
  **
  **
  **
 ***
 ***
 ***
****


Triangle gauche, 8 ligne(s) X 25 colonne(s).

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

```
Losange, 8 ligne(s) X 25 colonne(s).

     *
    * *
   *   *
  *     *
 *       *
  *     *
   *   *
    * *
     *


On affiche les formes dans les cadres fournis par l'utilisateur

Voulez-vous saisir un autre cadre? o /n
o
Entrez la base de la forme
20
Entrez la hauteur de la   forme
13

13 ligne(s) X 20 colonne(s).

********************

*
*
*
*
*
*
*
*
*
*
*
*
*

**************
*            *
*            *
*            *
*            *
*            *
*            *
*            *
*            *
*            *
*            *
**************


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

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


*
**
****
*****
*******
********
**********
************
************
****************
****************
******************
**********************



       *
      * *
     *   *
    *     *
   *       *
  *         *
 *           *
*             *
 *           *
  *         *
   *       *
    *     *
     * *
      *
Voulez-vous saisir un autre cadre? o /n
n

Merci d'avoir utilisé ce programme !
```