Laboratoire 05A - Sous-Classes/héritage - Formes

|  |
| --- |
| **Formules d’aires et de volume pour les prismes :**  **Prisme – volume :** longueur \* largeur \* hauteur  **Prisme – aire :** 2 \* longueur \* largeur + 2 \* longueur \* hauteur + 2 \* largeur \* hauteur  **Cube – volume :** arete \* arete \* arete  **Cube – aire:** 6 \* arete \* arete = ( 2 \* arete \* arete + 2 \* arete \* arete + 2 \* arete \* arete)  **Pyramide base carrée – aire :** longueur \* largeur \* hauteur / 3  **Pyramide base carrée – volume :**  **Aire totale =** aire de base + 4 \* aire des côtés  **Aire de base =** longueur \* longueur  **Aire d’un côté =** apotheme \* longueur / 2  **Apotheme =** racine carrée (longeur2 + hauteur2) |

**Note : pour résoudre ce problème, pour le volume de la pyramide, vous devez trouver ce qu’est l’apotheme et comment l’intégrer à la solution.**

A- Créer une classe Prisme servant à modéliser des prismes rectangulaires. Au-delà des méthodes habituelles, codez 2 constructeurs : un sans paramètres initialisant toutes les variables à 1 unité et l'autre avec paramètres, permettant d'initialiser les variables à des valeurs quelconques.

B- De plus, coder une méthode permettant de calculer l'aire de la surface du Prisme modélisé et une permettant de calculer le volume du Prisme modélisé.

C- Créer dans le même package que la classe Prisme une nouvelle classe - Cube - modélisant des cubes !

1. Cette classe est une sous-classe de Prisme
2. Bâtir les deux constructeurs appropriés ( faire appel aux constructeurs de Prisme avec super )
3. Déterminer si de nouvelles versions d’aire et de volume sont nécessaires

D- Créer dans le même package que la classe Prisme une nouvelle classe - PyramideBaseCarree – modélisant des pyramides à base carrée

1. Cette classe est une sous-classe de Prisme
2. Bâtir les deux constructeurs appropriés
3. Implémenter en surcharge les méthodes aire() et volume() ( il vous faut connaître l'apothème )

E-Pour tester le tout, créer une classe TestPyramide contenant qu'une méthode, soit main ( String [] args ) dans laquelle vous…

* créez deux prismes: p1 avec le constructeur par défaut, p2 avec le constructeur ( 2, 4, 2).
* créez deux pyramides à base carrée : py1 avec le constructeur par défaut, py2 avec une hauteur de 6 et une arête de base de longueur = 9.
* créez un Cube c1 avec une arête de longueur 35.
* affichez les données / résultats de méthodes suivantes, lorsque possible :

- p1.setLongueur(25)

- p1.aire()

- py2.volume ()

- py1.setHauteur (35)

- py1.volume()

- py2.aire()

- c1.aire()

SOLUTIONS

Classe Prisme

|  |
| --- |
| package projetheritageforme;  public class Prisme {  private double longueur;  private double largeur;  private double hauteur;    public Prisme() {  longueur = 1;  largeur = 1;  hauteur = 1;  }   public Prisme(double longueur, double largeur, double hauteur) {  this.longueur = longueur;  this.largeur = largeur;  this.hauteur = hauteur;  }   public double volume () {  return longueur \* largeur \* hauteur;  }   public double aire() {  return longueur \* largeur \* 2 + longueur \* hauteur \* 2 + hauteur \* largeur \* 2 ;  }   public void setLongueur(double longueur) {  this.longueur = longueur;  }   public double getLongueur() {  return longueur;  }   public void setLargeur(double largeur) {  this.largeur = largeur;  }   public double getLargeur() {  return largeur;  }   public void setHauteur(double hauteur) {  this.hauteur = hauteur;  }   public double getHauteur() {  return hauteur;  } } |

Classe Cube

|  |
| --- |
| package projetheritageforme;   public class Cube extends Prisme {  public Cube() {  super ();  }   public Cube ( double arete ){  super ( arete, arete, arete );  } } |

Classe PyramideBaseCarree

|  |
| --- |
| package projetheritageforme;  public class PyramideBaseCarree extends Prisme {   public PyramideBaseCarree(){  super();  }   public PyramideBaseCarree(int coteBase, int hauteur ){  super (coteBase, coteBase, hauteur );  }   public double aire() {  double longueurBase = getLongueur();  *// aire de la base* double aireBase = longueurBase \* longueurBase;   *// calcul de l'apotheme* double apoCarre = Math.*pow*( getHauteur(), 2) + Math.*pow*(longueurBase/2, 2);  double apoToutCourt = Math.*sqrt*(apoCarre);   *// aire côté* double aireCote = longueurBase\*apoToutCourt /2;   return aireBase + 4\* aireCote;  }   public double volume() {  *// voulme du prisme correspondant / 3* return super.volume() /3;  } } |

Classe TestForme

|  |
| --- |
| package projetheritageforme;  import java.text.DecimalFormat;  public class TestForme {  public static void main(String[] args)  {   DecimalFormat df = new DecimalFormat("0.00");  Prisme p1 = new Prisme ();  Prisme p2 = new Prisme ( 2,4,2);  PyramideBaseCarree py1 = new PyramideBaseCarree();  PyramideBaseCarree py2 = new PyramideBaseCarree(9,6);  Cube c1 = new Cube ( 35);  p1.setLongueur(25);  System.*out*.println ( p1.aire());  System.*out*.println ( py2.volume ());  py1.setHauteur(35);  System.*out*.println (df.format(py1.volume()));  System.*out*.println (py2.aire());  System.*out*.println ( c1.aire());  } } |