

```

/**
 * Gestion d'une citerne (réservoir).
 *
 * @author gamca174 (Gamboa Carlos) / olial319 (Olinger Alex)
 * @version 19/11/2015 07:09:15
 * Classe: 11TG
 */
public class Cistern
{
    /** Volume maximum en litres. */
    private double maximumVolume;

    /** Volume actuel en litres. */
    private double currentVolume;

    /**
     * Constructeur : définit une nouvelle citerne par ses dimensions extérieures.
     *
     * @param pRadius le rayon de la citerne en m
     * @param pHeight la hauteur de la citerne en m
     */
    public Cistern(double pRadius, double pHeight)
    {
        maximumVolume = Math.PI * pRadius*pRadius * pHeight * 1000; // Attention à la conversion m^3 -> litres
        currentVolume = 0;
    }

    /**
     * Ajouter de l'eau à la citerne jusqu'à faire le plein.
     *
     * @param pVolume volume d'eau à ajouter
     */
    public void add(double pVolume)
    {
        if (currentVolume + pVolume <= maximumVolume)
        {
            currentVolume = currentVolume + pVolume;
        }
        else
        {
            currentVolume = maximumVolume;
        }
    }

    /**
     * Retirer de l'eau de la citerne jusqu'à la vider.
     *
     * @param pVolume volume d'eau à retirer
     */
    public void drain(double pVolume)
    {
        if (pVolume <= currentVolume)
        {
            currentVolume = currentVolume - pVolume;
        }
        else
        {
            currentVolume = 0;
        }
    }

    /**
     * Retourner le volume actuel.
     *
     * @return volume actuel d'eau dans la citerne
     */
    public double getCurrentVolume()
    {
        return currentVolume;
    }

    /**
     * Retourner le taux de remplissage actuel.
     *
     * @return taux de remplissage en pourcentage
     */
    public double getCurrentRate()
    {
        return currentVolume/maximumVolume * 100;
    }

    /**
     * Retourner un message indiquant le volume et le taux de remplissage actuels.
     *
     * @return texte contenant le message d'information
     */
    public String toString()
    {
        return "fill level: "+currentVolume+"l / "+maximumVolume +"l (" +getCurrentRate()+"%)" ;
    }
}

```

```
/**
 * Tester la classe Cistern.
 */
 *
 * @author      gamca174 (Gamboa Carlos) / olial319 (Olinger Alex)
 * @version    19/11/2015 07:09:15
 * Classe:    11TG
 */
public class TestCistern
{
    /**
     * Programme principal.
     */
    public static void main(String[] args)
    {
        double v;
        Cistern cistern = new Cistern(0.5, 2);

        System.out.println("Cistern (r=0.5m / h=2m): ");
        System.out.println(" --> "+cistern.toString());

        v = 120;
        System.out.println("Add "+v+"l: ");
        cistern.add(v);
        System.out.println(" --> "+cistern);

        v = 1600;
        System.out.println("Add "+v+"l: ");
        cistern.add(v);
        System.out.println(" --> "+cistern);

        v = 100.79632;
        System.out.println("Drain "+v+"l: ");
        cistern.drain(v);
        System.out.println(" --> "+cistern);

        v = 1500;
        System.out.println("Drain "+v+"l: ");
        cistern.drain(v);
        System.out.println(" --> "+cistern);
    }
}
```