Solution série 2

Exo1 :

```
public class Point {
char nom;
double x;
double y;
public Point(char nom, double x, double y) {
	this.nom = nom;
	this.x = x;
	this.y = y;
}
void afficher()
{
	System.out.println("les coordonnéesle du point " + this.nom + " sont " + this.x +" et " +
this.y );

	}
void deplacer (double dx, double dy)
{
	this.x += dx;
	this.y += dy;

}
double distance(Point p)
{
	double dis ;
	dis = Math.sqrt((this.x-p.x)*(this.x-p.x) + (this.y-p.y)*(this.y-p.y));
	return dis;
	}
}
```

```
public class TestPoint {

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		// TODO Auto-generated method stub
Point m = new Point('m',3,4);
m.afficher();
m.deplacer(1, 2);
m.afficher();

	}

}
```

Exo2 :

```
public class MonEntier {
 MonEntier(int valeur) {
            this.valeur = valeur;
      }
int valeur ;

MonEntier ajouter(MonEntier e)
{
      //il faut créer un objet pour le resultat
      MonEntier result= new MonEntier(0);
      result.valeur = this.valeur + e.valeur;
      return result;
      }
MonEntier diminuer(MonEntier e)
{
      MonEntier result= new MonEntier(0);
      result.valeur = this.valeur - e.valeur;
      return result;
      }
MonEntier multiplier(MonEntier e)
{
      MonEntier result= new MonEntier(0);
      result.valeur = this.valeur * e.valeur;
      return result;
      }
MonEntier divEntier(MonEntier e)
{
      MonEntier result= new MonEntier(0);
      result.valeur = this.valeur + e.valeur;
      return result;
      }
int puissance(int power)
{
      int result = this.valeur;
      for (int i=1; i<=power;i++)
            result*=result;
      return result;
      }
int factoriel()
{
      int fact = 1;
      for (int i=1; i<=this.valeur;i++)
            fact*=i;
      return fact;
      }
void afficher()
{
      System.out.println("la valeur de ce nombre est : " + this.valeur);
      }
}


public class TestMonEntier {

      /**
       * @param args
       */
      public static void main(String[] args) {
            // TODO Auto-generated method stub
            MonEntier x = new MonEntier(5);
            MonEntier y = new MonEntier(5);
```

```
		/* le test d'egalité avant x=y
		 les deux variables x et y ne sont pas égaux parcequ'elles désignent deux
		objets différents.	*/
		if (x==y)
			System.out.println("les deux variable sont égaux ");
		else
			System.out.println("les deux variable sont différents ");


	x=y;

	/* le test d'egalité après x=y
	 les deux variables x et y  sont  égaux parcequ'elles désignent cette fois-ci le même
	objet. */
	if (x==y)
		System.out.println("les deux variable sont égaux ");
	else
		System.out.println("les deux variable sont différents ");


	MonEntier resultat= new MonEntier(0);
	resultat = y.ajouter(x);
	resultat.afficher();

}

}
```

Exo3 :

```
public class Livre {
	private String titre;
	private String auteur;
	private int nbrPage;
	private static int nbLivre = 0;
	private int numLivre ;
	public Livre (String titre)
	{
		this(); //on doit calculer le nbr de livre crée à travers le constructeur sans argument
		this.titre = titre;
		}
	public Livre (String titre, String auteur, int nbrPage)
	{

		this(titre); //appeler le constructeur qui permet de initialiser la valeur de titre
		this.auteur = auteur;
		this.nbrPage= nbrPage;
	}
	public Livre ()
	{
	this.nbLivre++;
	this.numLivre = this.nbLivre;
	}
	public void afficher ()
	{
		System.out.println("* le numéro de livre : " + this.numLivre );
		System.out.println("   - le titre : " + this.titre );
		System.out.println("   - l'auteur : " + this.auteur );
		System.out.println("   - le nombre de page : "+ this.nbrPage);

	}
	public void setTitre(String titre)
	{
	this.titre = titre;
	}
	public void setAuteur(String auteur)
	{
		this.auteur = auteur;
	}
```

```java
public void setnbrPage(int nbrPage)
{
	this.nbrPage= nbrPage;
	}
public String getTitre()
{return this.titre;
	}
public String getAuteur()
{return this.auteur;
	}
public int getNbrPage()
{return this.nbrPage;
	}
}

public class Bibliotheque {

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Livre l1 = new Livre("les Cahiers du Programmeur","Emmanuel Puybaret",367);
		Livre l2 = new Livre("Programmer en Java");
		l1.afficher();
		l2.afficher();

	}

}
```

Exo4 :

```java
class PileEntier  {
	    private int tailleMax = 1000;   // Taille maximale
	    private int sommet = -1;        // Sommet du tableau
	   private int[] tableau;

	    // Constructeur: crée une pile vide de taille maximale donnée
	    public PileEntier(int tailleMax) {
	        this.tailleMax = tailleMax;
	        this.sommet = -1;
	        this.tableau = new int [tailleMax];
	    }


	    /** Effacer tous les elements   */
	    public void viderPile() {
	        this.sommet = -1;
	    }

	    /** Empiler */
	    public void empiler(int element) {
	        if (this.sommet == this.tailleMax - 1)
	            System.out.println("Impossible d'empiler: la pile est pleine");
	        else
	        {
	        this.sommet++;
	        this.tableau[sommet] = element;
	        }
	    }
```

```
        /** Dépiler */


        public void depiler() {
            if (isPileVide())
                System.out.println("Impossible de desempiler: la pile est vide");
            else
                this.sommet--;
        }

        // Sélecteurs

        /** Teste si une pile est vide */
        public boolean isPileVide() {
            return (this.sommet == -1);
        }


        /** Retourne la valeur du sommet */
        public int sommet() {
            if (isPileVide()) {
                System.out.println("Impossible de donner la valeur du sommet: la pile est
vide");
                System.exit(-1); // A remplacer impérativement par un traitement
                                 // d'exceptions, dès que vous connaitrez cette notion!!
            }

            return this.tableau[this.sommet];
        }
        /** afficher le contenu de la pile */
        public void listerPile ()
        {
          if (isPileVide())
                System.out.println("la pile est vide");
          else
          {
                System.out.println("la pile contient : ");
                for (int i =0; i<=this.sommet;i++)
                      System.out.println(this.tableau[i]);
          }
        }

    }


public class TestPile {

    /**
     * @param args
     */
    public static void main(String[] args) {
          // TODO Auto-generated method stub
          PileEntier p1 = new PileEntier(10);
          p1.empiler(5);
          p1.empiler(-1);
          p1.empiler(9);
          p1.empiler(33);
          p1.listerPile();
          p1.depiler();
          p1.depiler();
          p1.listerPile();
                }

}
```