UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE

TD n°2 : Les classes
Informatique
M2TTT

Exercice 1

Soit le programme :

```
class A
{
	public A(int coeff)
	{
		nbre *= coeff;
		nbre += decal;
	}
	public void affiche()
	{
		System.out.println("nbre = " + nbre + "  decal = " + decal);
	}

	private int nbre = 20;
	private int decal;
};

public class Init
{
}
```

1. Que fait ce programme ?
2. Pour utiliser la classe A
   a. Dans quelle classe, un objet de type A doit-il être créé ?
   b. Créer un objet a de type A et utiliser la méthode affiche()
   c. Obtenez-vous le résultat attendu à la question 1 ?

Exercice 2

```
public class Entier
{
	private Entier (int nn) { n = nn;}

	private void affiche()
	{
		System.out.println("n = " + n);
	}

	private int n;
}
public class TestEntier
{
```

```
	public void main(String args[])
	{
		Entier n1 = new Entier(2);

		n1.affiche();
	}
}
```

1. Ce programme comporte des erreurs, les corriger
2. Ajouter à la classe entier un méthode void increment(int dn) qui incrémente de dn la valeur de n
3. Dans la fonction main
   a. Déclarer un entier n2 initialisé à 5,
   b. Interpréter l'instruction suivante :
      System.*out*.println("n1==n2 est "+ (n1==n2));
   c. Que faut-il faire pour que n1==n2 soit vraie ?
4. Peut-on accéder à n ? Pourquoi ?
   a. Implémenter une méthode getN() donnant la valeur de n
   b. Implémenter une méthode setN(int nn) permettant d'attribuer la valeur nn à n

Exercice 3

Que fait ce programme ? Que constatez-vous sur les valeurs de n et de p avant et après appel de la méthode ajout ? Comment contourner le problème ?

```
class Add
{
	void ajout(int n1, int n2)
	{
		System.out.println(« Debut ajout : n1 = «  + n1 + «   n2 = «  + n2) ;
		n1 = n1+n2 ;
		System.out.println(« Fin de ajout : n1 = «  + n1 + «  n2 = «  + n2) ;
	}
}

public class TestAdd
{
	public static void main(String args[])
	{

		int n = 3, p = 4;
		Add A = new Add();
		System.out.println(« Avant appel de ajout  n = «  + n + «   p = «  + p) ;
		A.ajout(n, p) ;
		System.out.println(« Apres appel de ajout  n = «  + n + «   p = «  + p) ;

	}
}
```

1. Que fait ce programme ? Que constatez-vous sur les valeurs de n et de p avant et après appel de la méthode ajout ? Comment contourner le problème ?
2. Pour contourner la difficulté rencontrée dans la question 1, créer une classe MonInt qui aura
   a. Un champ int a
   b. Un constructeur qui affectera une valeur de type int à a

    c. Une methode int getValue() qui renvoie la valeur de a
    d. Une methode void setValue(int) qui permet d'affecter une nouvelle valeur à a
3. Surcharger la methode ajout pour qu'elle prenne en argument des variable de type monInt
4. Créer deux instances de la classe MonInt et appeler les deux fonctions ajout en affichant à chaque fois les valeurs des arguments avant et après appel de la fonction. Que constatez-vous ?

Exercice 4

On dispose de la classe Point telle que vue en cours. En ajoutant les fonctionnalités nécessaires à la classe Point, réaliser une classe Segment permettant de manipuler des segments et ayant les fonctionnalités suivantes :

Segment(Point origine, Point extremite)
Segment(double xOr, double yOr, double xExt, double yExt)
double longueur();
void deplaceOrigine(double dx, double dy)
void deplaceExtremite(double dx, double dy)
void affiche()