TP N° 2

Exercice 1

Classe Ident

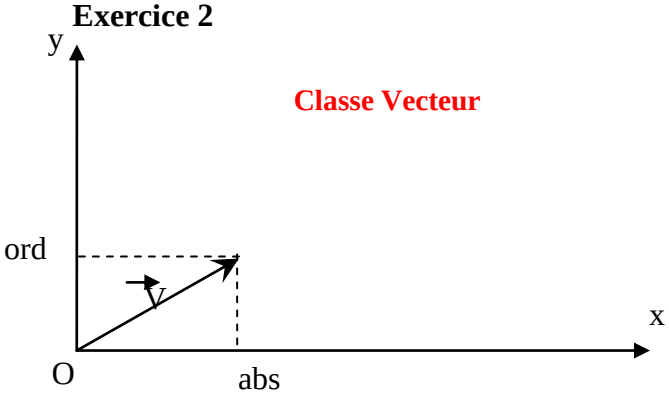
Réaliser une classe qui permet d'attribuer un **numéro** unique à chaque nouvel objet créé (1 au premier, 2 au suivant...). On dotera la classe d'un **constructeur** et d'une méthode **getIdentMax** fournissant le **numéro du dernier objet** créé.

Classe TestIdent

Écrire un petit programme d'essai (principal) qui permet de créer 3 objets a,b et c et d'afficher :

numéro de a : 1
numéro de b : 2
dernier numero 2
numéro de c : 3
dernier numéro 3

Exercice 2



Classe Vecteur

Il s'agit de modéliser un vecteur $\vec{V}$ dont l'origine est en (0, 0) (un tel vecteur est donc caractérisé par deux nombres entiers **abs** et **ord**). Les opérations que l'on souhaite faire sur ce vecteur sont :

- Calculer sa longueur, par une méthode, nommée **longueur**(), et qui retourne cette longueur sous forme d'un double. Longueur de $\vec{V}$(abs, ord) est : **Math.sqrt(x * x** + y * y).

- Savoir si le vecteur concerné est ou non plus petit qu'un autre vecteur donné ; on écrira pour cela une méthode nommée **plusPetitQue**(..) qui recevra en paramètre l'autre vecteur et qui retournera un booléen.

- Additionner au vecteur concerné un autre vecteur ; on écrira pour cela une méthode nommée **addition**(..) qui recevra en paramètre l'autre vecteur et qui ne retournera rien.

- Additionner deux vecteurs donnés ; on écrira pour cela une méthode **statique** nommée aussi **addition** (..) (surcharge) qui recevra en paramètres les deux vecteurs à additionner et qui retournera le résultat sous forme d'un objet de type Vecteur

- une méthode redéfinissant la méthode : public String **toString**() de la classe Object. Celle-ci décrira une instance de Vecteur sous la forme d'une chaîne de caractères (par exemple, le vecteur de coordonnées 1 et 2 pourra être décrit par la chaîne de caractères : "vecteur (1, 2)"). La méthode retournera cette chaîne.

- Que retourne toString(), si elle n'est pas surchargée ?

Classe EssaiVecteur

- On créera une classe **EssaiVecteur** contenant une méthode main pour tester la classe Vecteur. Les coordonnées des vecteurs seront saisies ou bien à l'aide de la classe Scanner (voir TP n°1), ou bien elles pourront être indiquées sur la ligne de commande (via les arguments (args) de la méthode main). Un exemple d'exécution de ce programme peut être :

```
> java EssaiVecteur 1 2 5 -3
v1 : vecteur (1, 2)
Longueur de vecteur (1, 2) : 2.23606797749979
v2 : vecteur (5, -3)
le vecteur (1, 2) est plus petit que le vecteur (5, -3)
v1 après addition de v2 : vecteur (6, -1)
v3 = v1 +  v2 : vecteur (11, -4)
```

Math.sqrt(abs*abs + ord*ord).

N.B. :

- Grace à la méthode « public **static int parseInt(String s)** » de la classe **Integer** vous pouvez convertir la chaine s en entier.
- args est un tableau de String où : args[0] représente le 1er argument, args[1] le 2ème, etc.
  dans l'exemple : args[0]="1", args[1]="2", args[3]= "5" et args[3]="-3"
- Si v1 est une instance de la classe Vecteur, alors System.out.println(v1) permettra d'afficher ce que retournera la méthode toString().

Exercice 3

Classe : **Score**

1. Créez une classe qui représente le score d'un joueur. Cette classe, nommée **Score**, devra gérer trois données : le **nom** du joueur, **son score (sonScore)**, le **score maximal (scoreMax)** autorisé.
2. Ecrivez une méthode "**afficher ()**" de cette classe qui affiche le nom et le score du joueur.
3. Créez un **constructeur** de cette classe où le nom par défaut serait "inconnu", le score initial serait 0 et le score maximal serait 100.
4. Pour ne pas créer que des inconnus, écrivez un second constructeur (surcharge) qui fixera le nom du joueur. Eliminez la **redondance** de codes entre les 2 constructeurs.
5. Dans un programme on peut lire ou modifier le contenu du score du joueur. Pour permettre d'agir sur lui en tenant compte de ses limites (0 <=sonScore <= scoreMax) vous devez ajouter deux nouvelles méthodes : l'une de lecture "**lireScore()**" (est une fonction qui retourne le score) et l'autre d'écriture "**ecrireScore(int sco)**". En plus ajouter une troisième nouvelle méthode "**ajouter(int points)**" permettant d'ajouter des points au score.

Classe : **ScoreEssai**

1. Pour les besoins d'un programme de jeu particulier, pour pouvoir gérer le score d'un joueur, et aussi le nombre d'essais qui ont été effectués pour obtenir ce score, créez une classe dérivée de la classe score (**ScoreEssai**) qui prend en compte le **nombre d'essais (nbEssai)**.
2. Dans cette nouvelle classe créer deux constructeurs en complétant les constructeurs de la classe mère (utilisation du mot clé "**super**").
3. Surcharger la méthode **ecrireScore(int sco)** qui complète le code de celle de la classe Score (utilisation du mot clé "**super**") pour que chaque mise à jour du score s'accompagne d'une augmentation du nombre d'essais.
4. Ajouter une méthode **lireNbEssais()** qui permet de retourner le contenu de l'attribut **nbEssai**.

Classe : **Jeu**

1. Ecrire la classe principale **Jeu** qui simule un jeu qui consiste à lancer un dé jusqu'au moment où on obtient un total supérieur ou égal à 21 points. Afficher le nombre d'essais.

N.B. :

Syntaxe de la boucle est : while (condition) { … }
Double d; //Double est une classe "réelle". Sa méthode intvalue() renvoie sa partie entière.
..
d=new Double(6*Math.random()+1); // permet de générer un nombre aléatoire réel en 1 et 6.

Solutions

Exo1

```
class Ident {
   static int numCour=0 ; // dernier numero attribué
   int num ; // numero de l'objet

   Ident (){ numCour++ ;num = numCour ;}
   static int getIdentMax(){ return numCour ;}}

class TestIdent {
   public static void main (String args[]) {
      Ident a = new Ident(), b = new Ident() ;
      System.out.println ("numero de a : " + a.num) ;
      System.out.println ("numero de b : " + b.num) ;
      System.out.println ("dernier numero " + Ident.getIdentMax()) ;
      Ident c = new Ident() ;
      System.out.println ("numero de c : " + c.num) ;
      System.out.println ("dernier numero " + Ident.getIdentMax()) ;}}
```

Exo2

```
public class Vecteur{
   int abs, ord;

   Vecteur(int _abs, int _ord){       abs = _abs;       ord = _ord;    }
   double longueur(){       return Math.sqrt(abs * abs + ord * ord);    }
   void addition(Vecteur v){       abs = v.abs + abs;       ord = v.ord + ord;    }

   static Vecteur addition(Vecteur v1, Vecteur v2){
      return new Vecteur(v1.abs + v2.abs, v1.ord + v2.ord);    }

   boolean plusPetitQue(Vecteur v){       return longueur() < v.longueur();    }
   public String toString(){       return "vecteur (" + abs + ", " + ord + ")";    } }

public class EssaiVecteur{
      public static void main(String[] argv){
         Vecteur v1 = new Vecteur(Integer.parseInt(argv[0]), Integer.parseInt(argv[1]));
         System.out.println("v1 : " + v1);
         System.out.println("Longueur de " + v1 + " : " + v1.longueur());
         Vecteur v2 = new Vecteur(Integer.parseInt(argv[2]), Integer.parseInt(argv[3]));
         System.out.println("v2 : " + v2);
         if (v1.plusPetitQue(v2))
            System.out.println("Le " + v1 + " est plus petit que le " + v2);
         else
            System.out.println("Le " + v1 + " est au moins aussi grand que " + "le " + v2);

         v1.addition(v2);
         System.out.println("v1 après addition de v2 : " + v1 +"--"+ v1.getClass());
         Vecteur v3 = Vecteur.addition(v1, v2);
         System.out.println("v3 = v1 + v2 : " + v3);       }}
```

Exo3

```
class Score {
        String nom;    int sonScore; int scoreMax;

        void affiche() { System.out.println(nom+" : "+sonScore);}

        Score() { nom="inconnu";   sonScore=0;   scoreMax=100;       }
        //Score() { this("inconnu");} // pour éliminer la redondance de codes
        Score(String nom) { this.nom=nom;          sonScore=0;   scoreMax=100;       }


        //méthode de lecture
        public int lireScore() { return sonScore;   }
        //méthode d'écriture
        public void ecrireScore(int sco) {
                if (sco>scoreMax) sonScore=scoreMax;
                else if (sco<0) sonScore=0; else sonScore=sco;     }
        public void ajoute(int points) { ecrireScore(sonScore+points); }}



class ScoreEssai extends Score {
        int nbEssais;
        ScoreEssai() {          super(); nbEssais=0;}
        ScoreEssai(String nom) {super(nom); nbEssais=0;       }
        public void ecrireScore(int sco) { super.ecrireScore(sco);          nbEssais++;  }
        int lireNbEssais() {    return nbEssais;      }}



class Jeu {
   public static void main(String args[]) {
      Double d;
      int tirage;
      ScoreEssai reda=new ScoreEssai("Réda");
      while (reda.lireScore()<21){
         d=new Double(6*Math.random()+1);
         tirage=d.intValue();
         reda.ajoute(tirage);
         reda.affiche();
      }
      System.out.println("Réda a gagné en "+  reda.lireNbEssais()+" coups.");     }}
```