

Feuille de TP1

Classe CoupleEntiers

On vous propose le code java suivant (classe CoupleEntier et classe ExecutableEntier) :

```
public class CoupleEntiers {
    private int premier, second;
    public CoupleEntiers() {
        this.premier = 0;
        this.second = 0;
    }
    public void setPrem(int premier) {this.premier = premier;}
    public void permute() {
        int aux;
        aux = this.premier;
        this.premier = this.second;
        this.second = aux;
    }
    public int fraction() {
        return this.premier / this.second;
    }
}

public class ExecutableCoupleEntiers {
    public static void main(String [] args) {
        CoupleEntiers couple = new CoupleEntiers();
        couple.setPrem(-16);
        System.out.println(couple); // (1)
        System.out.println(couple.fraction()); // (2)
    }
}
```

1. Testez ces deux codes et exécutez les (rappelez les commandes permettant de le faire).
2. Que se passe-t-il avec l'instruction (2) comment y remédier ? Modifiez le code en conséquence.
3. Que constatez vous sur l'instruction (1) ? Que proposez vous ? Ecrivez ce code.
4. On veut maintenant ajouter les instructions suivantes à la classe **ExecutableEntier** :

```
couple.setSec(-6);
System.out.println(couple.getPrem()+" "+couple.getSec());
System.out.println(couple.somme()); // affiche -22
```

Définissez les méthodes nécessaires.

5. Définissez la méthode **plus** permettant d'ajouter deux *CoupleEntiers* pour donner un *CoupleEntiers* dont le premier élément est la somme des premiers éléments des deux *CoupleEntiers* et le second élément la somme des seconds éléments des deux

CoupleEntiers. Modifiez également votre code de la classe **ExecutableEntier** pour tester cette nouvelle méthode.

- Proposez maintenant un constructeur à deux paramètres. Vous pouvez alors modifier le code de la méthode **plus**, faites-le.

Classe Complexe

- Définissez la classe **Complexe** permettant de compiler et d'exécuter la classe **ExecutableComplexe** suivante :

```
public class ExecutableComplexe {
    public static void main(String [] args) {
        Complexe complexe1 = new Complexe(5, -6);
        Complexe complexe2 = new Complexe(2, -4);
        System.out.println(complexe1.getReel()+ " "+ complexe2.getImag());
        complexe1.afficheCartesien();
        // affiche (5, -6)
        complexe2.afficheComplexe();
        // affiche 2 +i -4
    }
}
```

- Ajouter le code permettant de calculer la somme de deux Complexe(s). Testez le en modifiant votre exécutable.
- Ecrivez une méthode **egal** permettant de tester si deux complexes sont identiques. Testez le en modifiant votre exécutable (tester le avec les couples (2, -4) et (4/2, -4).
- Testez le code suivant :

```
Complexe complexe2 = new Complexe(2, -4);
Complexe complexe3 = new Complexe(4/2, -4);
System.out.println(complexe2 == complexe3);
```

Essayez d'expliquer pourquoi.

- Vérifier la validité de votre classe **Complexe** en collant ci-dessous son code source :

1 Votre classe..

Classe Vecteur3f

On voudrait que l'exécutable suivant :

```
public class ExecutableVecteur3F {  
    public static void main(String [] args) {  
        Vecteur3f v3f = new Vecteur3f(4.5, 7.325, -6.875);  
        System.out.println(v3f.toString());  
    }  
}
```

affiche :

Vecteur3f : <4.5 7.325 -6.875 > De norme : 11.007781338671295

On rappelle que la norme d'un vecteur correspond à la racine carrée des produits des composantes du vecteur.

1. Définissez la classe **Vecteur3f** munie de 3 attributs de type double.
2. Ajoutez les getters, écrivez la méthode **modif** dont l'appel est le suivant :

```
v3f.modif(5.55, 1);
```

et permettant de modifier (par exemple) la première composante de v3f.

3. Testez le code suivant :

```
public class ExecutableVecteur3F {  
    public static void main(String [] args) {  
        Vecteur3f v3f = new Vecteur3f(4.5, 7.325, -6.875);  
        System.out.println(v3f.toString());  
        Vecteur3f v3ff = v3f;  
        v3ff.modif(5.55, 1);  
        System.out.println(v3ff.toString());  
        System.out.println(v3f.toString());  
    }  
}
```

Expliquez ce qui se passe et pourquoi ? que pouvez vous proposer ?

Classe Date

1. On vous demande de compléter la classe **Date** suivante

```
public class Date {
    private int jour, mois, annee;

    public Date(int j, int m, int a) { //TODO}

    public int getJour() { //TODO}

    public int getMois() { //TODO}

    public int getAnnee() { //TODO}

    public boolean bissextile() { //TODO}

    public int nbJourMois() { //TODO}

    public boolean valide() { //TODO}

    public String toString() { //TODO}

}
```

2. Testez votre classe en écrivant une classe **ExecutableDate**.
3. Ajouter une méthode **demain** qui indique la date du jour du lendemain (si celle-ci est valide).
4. (bonus) Ajouter une méthode **jourSemaine** indiquant quel est le jour de la semaine d'une date sachant que le 1/1/1900 était un **lundi**.

Bonus

Implémentez tous les codes de la feuille de TD.