

TP02 Correction

Matière : ATELIER DE PROGRAMMATION OBJET

Classes : SEM2

**package stylo;**

```
public class Stylo {
    // attributs
    private String marque;
    private String couleur;
    private int niveauEncre;

    // constructeurs
    public Stylo() {
        marque = "Bic";
        couleur = "Bleu";
        niveauEncre = 100;
    }

    public Stylo(String marque) {
        this.marque = marque;
        couleur = "Bleu";
        niveauEncre = 100;
    }

    public Stylo(String marque, String couleur) {
        this.marque = marque;
        this.couleur = couleur;
        niveauEncre = 100;
    }

    public Stylo(Stylo stylo) {
        marque = stylo.marque;
        couleur = stylo.couleur;
        niveauEncre = 100;
    }

    // méthodes
    public void afficher() {
        System.out.println("-----");
        System.out.println("Marque : " + marque);
        System.out.println("couleur : " + couleur);
        System.out.println("Niveau d'encre : " + niveauEncre);
        System.out.println("-----");
        System.out.println();
    }

    // ecrire retourne true si elle affiche quelque chose et
    // retourne false sinon (pas d'encre)
    public boolean ecrire(String chaine) {
        if (niveauEncre > 0) {
            System.out.println(chaine);
            niveauEncre--;
        }
    }
}
```

```

        return true;
    } else
        // ce else est optionnel!
        return false;
}

public void charger() {
    niveauEncre = 100;
}

public String getMarque() {
    return marque;
}

public String getCouleur() {
    return couleur;
}
}

```

**package stylo;**

```

public class TestStylo {
    public static void main(String[] args) {
        // déclaration
        Stylo s1, s2;
        // instantiation
        s1 = new Stylo();
        s2 = new Stylo("Bureau");
        // déclaration et instantiation
        Stylo s3 = new Stylo("Bureau", "Vert");

        s1.afficher();
        s3.afficher();
        s1.ecrire("Bonjour");
        s2.ecrire("Test de la méthode écrire");
        s1.afficher();
        s1.charger();
        s1.afficher();
        System.out.println("Marque : " + s1.getMarque());
        System.out.println("Couleur : " + s1.getCouleur());

        Stylo s4;
        s4 = new Stylo(s1);
        s4.afficher();
    }
}

```



## package rectangle;

```
public class Rectangle {
    // attributs
    private int longueur;
    private int largeur;
    private char caractereAffichage;

    // constructeurs
    public Rectangle() {
        longueur = 1;
        largeur = 1;
        caractereAffichage = '*';
    }
    public Rectangle(int longueur, int largeur) {
        this.longueur = longueur;
        this.largeur = largeur;
        caractereAffichage = '*';
    }
    public Rectangle(int longueur, int largeur, char caractereAffichage) {
        this.longueur = longueur;
        this.largeur = largeur;
        this.caractereAffichage = caractereAffichage;
    }
    public Rectangle(Rectangle rectangle) {
        longueur = rectangle.longueur;
        largeur = rectangle.largeur;
        caractereAffichage = rectangle.caractereAffichage;
    }
    // méthodes
    public int getSurface() {
        return longueur * largeur;
    }

    public int getPerimetre() {
        return 2 * (longueur + largeur);
    }

    public void agrandir(int coefficient) {
        if (coefficient > 0) {
            longueur = longueur * coefficient;
            largeur = largeur * coefficient;
        }
    }
    public void setCaractere(char caractereAffichage) {
        this.caractereAffichage = caractereAffichage;
    }
    public void afficher() {
        int i, j; // i pour la largeur et j pour la longueur
        for (i = 0; i < largeur; i++) {
            for (j = 0; j < longueur; j++)
                System.out.print(caractereAffichage);
            // nouvelle ligne
            System.out.println("");
        }
        System.out.println("");
    }
}
```



## package rectangle;

```
public class TestRectangle {  
    public static void main(String[] args) {  
        // déclaration  
        Rectangle r1, r2;  
        // instantiation  
        r1 = new Rectangle();  
        r2 = new Rectangle(8, 5);  
        // déclaration et instantiation  
        Rectangle r3 = new Rectangle(9, 6, '-');  
  
        r1.afficher();  
        r2.afficher();  
        r3.afficher();  
  
        System.out.println("Surface de r2 : " + r2.getSurface());  
        System.out.println("Périmètre de r3 : " + r3.getPerimetre());  
  
        r3.agrandir(2);  
        r3.afficher();  
  
        r2.setCaractere('|');  
        r2.afficher();  
    }  
}
```



## package compteur;

```
public class Compteur {
    private int valeur;
    private int cycle;
    private int pas;

    public Compteur(int cycle, int pas) {
        valeur = 0;
        this.cycle = cycle;
        this.pas = pas;
    }
    public Compteur(int cycle) {
        this(cycle, 1);
    }
    public Compteur() {
        this(10);
    }
    public int getValeur() {
        return valeur;
    }

    public void setValeur(int valeur) {
        this.valeur = valeur;
    }

    public int getPas() {
        return pas;
    }

    public void setPas(int pas) {
        this.pas = pas;
    }

    public int getCycle() {
        return cycle;
    }

    public void afficher() {
        System.out.println("Valeur :" + valeur);
    }

    public void incrementer() {
        valeur = (valeur + pas) % cycle;
    }

    public boolean incrementer1() {
        int ancienneValeur = valeur;
        incrementer();
        if (valeur < ancienneValeur) {
            System.out.println("Cycle achevé...");
            return true;
        }
        return false;
    }
}
```



## package compteur;

```
public class TestCompteur {
    public static void main(String[] args) {
        Compteur compteur1;
        compteur1 = new Compteur();
        compteur1.afficher();
        compteur1.incrementer();
        compteur1.afficher();
        System.out.println("          Cycle : " + compteur1.getCycle());
        System.out.println("          Pas : " + compteur1.getPas());
        for (int i = 0; i < 20; i++) {
            compteur1.incrementer();
            compteur1.afficher();
        }
        compteur1.setPas(2);
        System.out.println("          Cycle : " + compteur1.getCycle());
        System.out.println("          Pas : " + compteur1.getPas());
        for (int i = 0; i < 20; i++) {
            compteur1.incrementer1();
            compteur1.afficher();
        }
    }
}
```

## package ascenseur;

```
public class Ascencseur {
    // constantes
    public static final int ETAT_FERME = 0;
    public static final int ETAT_OUVERT = 1;
    // attributs
    private int nbEtage;
    private float capacite;
    private float poidsCharge;
    private int etat;
    private int etageEnCours;

    public Ascencseur(int nbEtage, float capacite) {
        this.nbEtage = nbEtage;
        this.capacite = capacite;
        poidsCharge = 0;
        etat = ETAT_FERME;
        etageEnCours = 0;
    }

    public Ascencseur(int nbEtage) {
        this(nbEtage, 600);
    }

    public Ascencseur() {
        this(4);
    }

    public int getNbEtage() {
        return nbEtage;
    }

    public float getCapacite() {
        return capacite;
    }

    public float getPoidsCharge() {
        return poidsCharge;
    }
}
```



```

public int getEtat() {
    return etat;
}
private String getChaineEtat() {
    return (etat == ETAT_OUVERT) ? "OUVERT" : "FERME";
}
public int getEtagEnCours() {
    return etageEnCours;
}
public void afficher() {
    System.out.println("-----");
    System.out.println("    Nb Etage      : " + nbEtage);
    System.out.println("    Capacite      : " + capacite);
    System.out.println("    Poids Charge   : " + poidsCharge);
    System.out.println("    Etat          : " + getChaineEtat());
    System.out.println("    Etage En Cours : " + etageEnCours);
    System.out.println("-----");
}
public void ouvrir() {
    etat = ETAT_OUVERT;
}
public void fermer() {
    etat = ETAT_FERME;
}
public void monter(float poids) {
    if (etat == ETAT_OUVERT) {
        poidsCharge += poids;
        System.out.println("    Poids Charge   : " + poidsCharge);
    } else
        System.out.println("Ouvrir les porte SVP!");
}
public boolean descendre(float poids) {
    if (etat == ETAT_OUVERT) {
        if (poids <= poidsCharge) {
            poidsCharge -= poids;
            System.out.println("    Poids Charge   : " + poidsCharge);
            return true;
        }
    } else
        System.out.println("Ouvrir les porte SVP!");
    return false;
}
public void descendreTous() {
    descendre(poidsCharge);
}
public boolean atteindre(int etageCible) {
    if ((etat == ETAT_FERME) && (poidsCharge <= capacite)
        && (etageEnCours != etageCible)) {
        if (etageEnCours < etageCible)
            monterA(etageCible);
        else
            descendreA(etageCible);
        ouvrir();
        return true;
    }
    afficherErreurAtteindre();
    return false;
}
}

```



```

private void monterA(int etageCible) {
    while (etageEnCours < etageCible) {
        etageEnCours++;
        System.out.println("Montée d'un étage...");
        System.out.println("Etage En Cours : " + etageEnCours
            + "    Etage Cible : " + etageCible);
    }
}

private void decendreA(int etageCible) {
    while (etageEnCours > etageCible) {
        etageEnCours--;
        System.out.println("Décente d'un étage...");
        System.out.println("Etage En Cours : " + etageEnCours
            + "    Etage Cible : " + etageCible);
    }
}

private void afficherErreurAtteindre() {
    if (etat == ETAT_OUVERT)
        System.out.println("Fermer les portes SVP!");
    else if (poidsCharge > capacite)
        System.out.println("Descendre des personne SVP!");
    else
        System.out.println("Vous êtes dans l'étage cible!");
}

public boolean atteindrel(int etageCible) {
    fermer();
    return atteindre(etageCible);
}
}

```

## package ascenseur;

```

public class TestAscencseur {
    public static void main(String[] args) {
        Ascencseur ascenseur;
        ascenseur = new Ascencseur(10, 800);
        ascenseur.afficher();
        ascenseur.ouvrir();
        ascenseur.monter(50);
        ascenseur.monter(70);
        ascenseur.afficher();
        ascenseur.atteindre(7);
        ascenseur.fermer();
        ascenseur.atteindre(7);
        ascenseur.decendre(50);
        ascenseur.monter(100);
        ascenseur.fermer();
        ascenseur.atteindre(7);
        ascenseur.atteindre(10);
        ascenseur.decendre(70);
        ascenseur.atteindrel(1);
        ascenseur.decendreTous();
        ascenseur.afficher();
    }
}

```

