

**Calculatrices non autorisées.
Documents limités à deux pages recto format A4.**

Les exercices sont indépendants. Les temps donnés sont indicatifs.

1–Questions de cours (15 minutes)

Répondez brièvement aux questions suivantes :

1. Donnez (au moins) 3 critères qui caractérisent les *interfaces* des *classes abstraites* .
2. Dans le TP du jeu d'échec (TP6-7), quelle a été l'utilité de l'héritage ?
3. Que représente la *surcharge* ? Donnez un exemple de surcharge du constructeur (prendre la classe Point).

2–Polygone (30 minutes)

On vous propose de définir une classe qui modélise un Polygone. Un Polygone sera représenté par une liste d'objets de type Point.

```
public class Point {
    public double x;
    public double y;

    public Point(double x,double y){
        this.x = x;
        this.y = y;
    }

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

    public String toString(){
        return " ( x = " + x +
            " , y = " + y + " ) ";
    }
}

public class Polygone {
    public LinkedList listeDePoints = null;

    public Polygone(){ // crée une liste de points vide
        listeDePoints = new LinkedList();
    }

    /*
     * A completer
     */
}
```

La documentation des méthodes des classes suivantes vous est donnée :

classe LinkedList

- void add(Object e1) : ajoute un élément à la fin de la liste.
- int size() : donne le nombre d'éléments contenus dans la liste.
- void remove(int i) : retire le i^{eme} élément de la liste.
- Object get(int i) : donne le i^{eme} élément de la liste.
- Iterator Iterator() : donne un itérateur sur la liste qui débute au premier élément.

classe ListIterator

- boolean hasNext() : retourne true si l'élément courant possède un suivant, false sinon
- Object next() : donne l'élément suivant dans la liste

Les éléments d'une liste sont indexés à partir de 0

L'opérateur de concaténation de chaînes de caractères est l'opérateur +

Question 1 Définissez une méthode de profil `int nbPoints()` qui retourne le nombre de points du polygone.

Question 2 Définissez une méthode de profil `void ajoute(Point p)` qui ajoute le point `p` passé en paramètre au polygone (ajout en fin de liste).

Question 3 Définissez une méthode de profil `String toString()` qui retourne une chaîne de caractères contenant l'énumération des coordonnées des points du polygone. (vous utiliserez un itérateur ainsi que la méthode `String toString()` de la classe `Point`).

Question 4 Définissez une méthode de profil `Point distanceMax(Point p)` qui retourne le point du polygone qui est le plus éloigné du point `p` passé en paramètre. (vous utiliserez un itérateur, la méthode `double distance(Point)` de la classe `Point`, ainsi que la méthode `double Math.max(double, double)`).

Question 5 En utilisant la méthode de la question 4, définissez la méthode de profil `double Diametre()` qui retourne la plus grande distance possible entre deux points du polygone.

3–QCM (30 minutes)

n'utilisez que la feuille réponse

Vous devez effectuer un ou plusieurs choix parmi ceux proposés. Vous préciserez éventuellement, selon la question, les raisons de vos choix (calcul, raisonnement suivi, aléas utilisés...)

Question 1 Quelles classes sont dites *génériques* ?

- a. `boolean`
- b. `LinkedList`
- c. `Integer`
- d. `PileDecimale` (pile d'entiers vue en cours)

Question 2 Quelle(s) instruction(s) possible(s) pour compléter ce code ?

```
LinkedList list = new LinkedList();
int a = 7;
/* A completer */
```

- a. `list.add(a);`
- b. `list.add(new Integer(a));`
- c. `list.add((Object)a);`
- d. `list.add((Object)new Integer(a));`

Question 3 On code les entiers en utilisant le codage complément à deux sur 8 bits. Quel est alors le codage hexadécimal de -23_d

- a. 17_h
- b. EA_h
- c. $E8_h$
- d. $E9_h$

Question 4 On utilise un code en virgule fixe sur 16 bits (8 bits / 8bits). Quelle est la valeur codée par $1A33$

- a. environ $26,50$
- b. environ $26,20$
- c. environ $27,20$
- d. environ $27,50$

Question 5 Comment réaliser une multiplication d'un entier positif x par 320_d sans utiliser l'opérateur de multiplication ? (rappel opérateur de décalage des bits à gauche est donné par " \ll ". Exemple : $x = x \ll 3$ est équivalent à $x = x * 8$)

- a. $x = x \ll 8 + x \ll 6$
- b. $x = x \ll 320$
- c. $x = x \ll 7 + x \ll 5$
- d. $x = x \ll x$

Question 6 Qu'est ce que signifie les attributs `static final` devant une variable de classe ?

- a. La variable est une constante
- b. Tous les objets de cette classe ont accès à une seule et unique variable
- c. Il y aura une variable pour chaque objet de la classe
- d. Tous les objets de cette classe peuvent modifier cette variable

Question 7 Quelles sont les fonctions récursives terminales ?

a.

```
int f1(int n){
    if (n == 0)
        return 1;
    else
        return f1(n-1) * n;
}
```

c.

```
double g1(double x,int n,double a,double b){
    if (n == 0)
        return a;
    if (n%2 == 0)
        return g1(x, n/2, a , b*b);
    else
        return g1(x, n/2, a*b, b*b);
}
```

b.

```
int f2(int n, int acc){
    if (n == 0)
        return acc;
    else
        return f2(n-1,acc*n);
}
```

d.

```
double g2(double x, int n){
    double r = 1;
    double b = x;
    while(n > 0){
        if (n%2 != 0)
            r*=b;
        n/=2;
        b*=b;
    }
    return r;
}
```

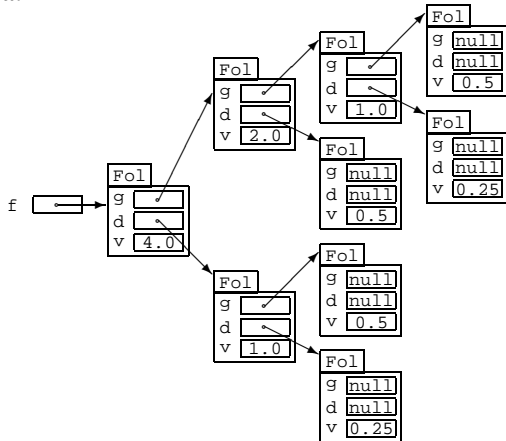
Question 8 Quel est l'état mémoire juste après l'exécution de l'instruction contenue dans la fonction main().

```
public class Fol{
    public Fol g = null;
    public Fol d = null;
    public double v;

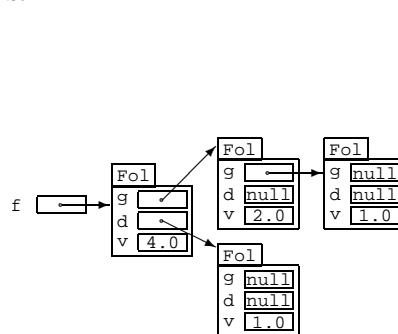
    public Fol(double a){
        this.v = a;
        if (a >= 1){
            g = new Fol(a/2);
            d = new Fol(a/4);
        }
    }
}
```

```
public Class App {
    public static void main(String args[]){
        Fol f =new Fol(4.0);
    }
}
```

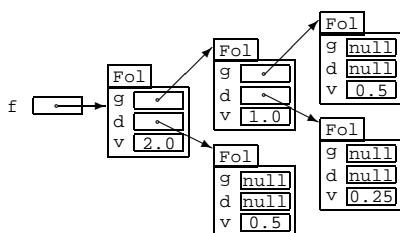
a.



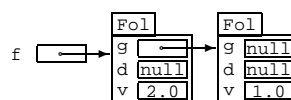
b.



c.



d.



4–Héritage (35 minutes)

n'utilisez que la feuille réponse

Question 1 à 6 Indiquez ce qu'affiche l'exécution de la commande `java App`. Vous pouvez éventuellement compléter les schémas afin de vous aider

```
public class A {
    public int cH ;
    public int cR ;

    public A(){
        cH = 10;
        cR = 20;
        System.out.println("Constructeur de A");
        System.out.println(" -cH = "+cH);
        System.out.println(" -cR = "+cR);
    }

    public void mH(){
        System.out.println("mH() de A");
        System.out.println(" -cH = "+cH);
        System.out.println(" -cR = "+cR);
    }

    public void mR(){
        System.out.println("mR() de A ");
        System.out.println(" -cH = "+cH);
        System.out.println(" -cR = "+cR);
    }
}

public class B extends A {
    public int cR;

    public B(){
        cH = 30;
        cR = 40;
        System.out.println("Constructeur de B");
        System.out.println(" -cH= " +cH);
        System.out.println(" -cR= " +cR);
        System.out.println(" -super.cR= "+super.cR);
    }

    public void mR(){
        cH = 0;
        if (super.cH == 0)
            super.mR();
        System.out.println("mR() de B");
        System.out.println(" -cH= " +cH);
        System.out.println(" -cR= " +cR);
        System.out.println(" -super.cR= "+super.cR);
    }
}

public class App {
    public static void main(String[] args){
        System.out.println("Question 1");
        A a = new A();

        System.out.println("Question 2");
        B b = new B();

        System.out.println("Question 3");
        a.mR();

        System.out.println("Question 4");
        a.cR = 5;
        System.out.println("    b.cR = "+    b.cR);
        System.out.println("((A)b).cR = "+ ((A)b).cR);

        System.out.println("Question 5");
        b.mH();

        System.out.println("Question 6");
        ((A)b).mR();
    }
}
```