

École Polytechnique
Département de Génie Informatique
INF-4100, Sujets spéciaux: le langage JAVA
Examen intratrimestriel: hiver 2003
Corrigé

Question 1: (8 points)

1.1 On désire une classe appelée Texte servant à conserver et à manipuler un texte d'un certain nombre de lignes qui peuvent être de longueurs différentes. Le texte sera conservé dans un tableau privé dont chaque élément est une chaîne de caractères. Deux constructeurs sont requis:

- un constructeur qui reçoit en argument un nombre de lignes et qui crée le tableau des chaînes de caractères, mais ne crée pas les éléments de ce tableau; tous les éléments du tableau doivent avoir la valeur "null";
- un constructeur "copie" qui reçoit en argument un autre objet de classe Texte et crée un nouvel objet identique à celui-là.

Écrivez les instructions requises pour réaliser cette classe, y compris les deux constructeurs.

Réponse:

```
public class Texte  
{
```

```
    private String L[ ];
```

```
    public Texte(int N)
```

```
{
```

```
    L = new String[N];
```

```
    for(int i = 0; i < N; i++) L[i] = null;
```

```
}
```

```
    public Texte( Texte A)
```

```
{
```

```
    L = new String[A.L.length];
```

```
    for(int i = 0; i < A.L.length ; i++) L[i] = new String(A.L[i]);
```

```
}
```

... et la suite de la classe, terminée par }

1.2 Puisque le tableau qui contient le texte est privé, on aura besoin de méthodes d'accès pour l'inspecter ou le modifier:

- une méthode `getLigne(n)` qui retourne le texte de la ligne numéro "n",
- une méthode `setLigne(n, T)` qui place la chaîne T dans la ligne n,
- deux méthodes appelées `getNbLignes()` et `getTailleLigne(n)` pour récupérer respectivement le nombre de lignes et la taille d'une ligne.

Écrivez les instructions requises.

Réponse:

```
public String getLigne(int n)
{
    if(n < L.length) return new String(L[n]); else return null;
}

public void setLigne(int n, String T)
{
    if(n < L.length) L[n] = new String(T);
}

public int getNbLignes( ) {return L.length;}

public int getTailleLigne(n)
{
    if(n < L.length) return L[n].length( ); else return 0;
}
```

1.3 On veut créer un objet de cette classe Texte nommé "VallonDeLamartine" et y placer les lignes suivantes:

```
"Mon cœur, lassé de tout même de l'espérance,"
"N'ira plus de ses vœux importuner le sort"
"Prêtez moi seulement, vallon de mon enfance,"
"Un asile d'un jour pour attendre la mort"
```

Écrivez les instructions requises.

Réponse:

```
Texte VallonDeLamartine = new Texte(4);

VallonDeLamartine.setLigne(0, "Mon cœur, lassé de tout, même de l'espérance");
VallonDeLamartine.setLigne(1, "N'ira plus de ses vœux importuner le sort");
VallonDeLamartine.setLigne(2, "Prêtez moi seulement, vallon de mon enfance");
VallonDeLamartine.setLigne(3, "Un asile d'un jour pour attendre la mort");
```

Question 2: (8 points)

2.1 Vous avez défini une classe ExQ2 qui contient les membres suivants:

```
class ExQ2
{
    private long Matricule;
    private String Prenom, Nom;
    private float Remuneration;
    private boolean Actif;
    .....
}
```

Vous désirez y inclure une méthode appelée EcrireSurDisque qui fait conserver tous les membres sur le disque dur en utilisant un objet de classe DataOutputStream qui lui est transmis en argument. Écrivez cette méthode.

Réponse:

```
public void EcrireSurDisque( DataOutputStream DOP)
{
    try
    {
        DOP.writeLong(Matricule);
        DOP.writeBytes(Prenom);
        DOP.writeBytes(Nom);
        DOP.writeFloat(Remuneration);
        DOP.writeBoolean(Actif);
    } catch (IOException E){ }
}
```

2.2 Dans le programme appelant, on a créé un objet de classe ExQ2 appelé Fevrier25; après initialisation et toutes sortes de calculs impliquant cet objet, on désire conserver les résultats sur disque dans un fichier d'octets dont le nom sera obtenu en utilisant un objet JFileChooser; pour une meilleure performance, les opérations de sortie doivent être tamponnées. Écrivez les instructions requises.

Réponse:

```
ExQ2 Fevrier25 = new ExQ2( ); // Seulement pour montrer comment l'objet est créé
```

```
JFileChooser JF = new JFileChooser;
```

```
File Fichier = null;
```

```
DataOutputStream Sortie = null;
```

```
JF.setSelectionMode(JFileChooser.FILES_ONLY);
```

```
int Resultat = JF.showOpenDialog(null);
```

```
if(Resultat == JFileChooser.APPROVE_OPTION)
```

```

{
    Fichier = JF.getSelectedFile();
    try {
        Sortie = new DataOutputStream(new BufferedOutputStream(
            new FileOutputStream(Fichier)));
    } catch (FileNotFoundException FNF){ };

    Fevrier25.EcritDisque(Sortie);
}

```

2.3 Quels sont les avantages d'inclure une méthode `EcritSurDisque` dans la classe `ExQ2` plutôt que d'écrire les instructions détaillées requises dans une classe appelante qui utilise un objet `ExQ2`?

Réponse: cette façon de faire est conforme au principe d'encapsulation des données en programmation par objets, et il en découle les avantages suivants:

- *cette méthode définit une partie de l'interface de l'objet indépendamment de l'implantation: l'usager de cette classe ne voit que l'interface, et il n'a pas besoin de connaître la façon dont la méthode est réalisée;*
- *l'écriture des valeurs des membres de la classe se fait au moyen d'une seule instruction;*
- *si la liste des membres de la classe change, ou que leurs types sont modifiés, la façon de les écrire sur disque reste la même;*
- *les membres étant privés, un classe-usagère ne peut pas les accéder directement; si on n'a pas prévu de méthodes d'accès pour les membres, la classe usagère n'a aucun moyen de faire l'outpu tsur disque.*

Question 3: (2 points)

Vous désirez qu'une application puisse interpréter une chaîne de caractères qui représente une expression arithmétique, donc composée de lettres, de chiffres, et des symboles `+ - * / (et)`; par exemple: `(ab - cd) / ef`. À cette fin vous décidez de créer un tableau de chaînes de caractères dont chaque élément contiendra une partie ("jeton") de la chaîne de caractères en question; par exemple:

```

(
ab
-
cd
)
/
ef

```

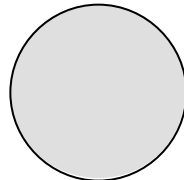
Écrivez les instructions requises, en prenant pour acquis que la chaîne de caractères s'appelle `TexteA`, qu'elle existe, et possède une valeur.

Réponse:

```
StringTokenizer ST = new StringTokenizer(TexteA, "+-*/()\n", true);
String Resultat[ ] = new String[ST.countTokens( )];
for(int i = 0; i < Resultat.length; i++) Resultat[i] = new String(ST.nextToken( ));
```

Question 4: (2 points)

L'output d'un applet doit avoir l'apparence suivante:



Voici un cercle rouge avec un remplissage jaune

Le cercle rouge avec un remplissage jaune doit avoir un rayon de 100 pixels et son centre être placé aux coordonnées [300, 300]. Pour simplifier la question, il est suffisant que le texte sous le cercle soit approximativement centré sur ce dernier. Écrivez les instructions requises.

Réponse:

```
public void paint(Graphics g)
{
    g.setColor(Color.yellow);
    g.fillOval(200, 200, 200, 200);
    g.setColor(Color.red);
    g.drawOval(200, 200, 200, 200);
    g.setColor(Color.black); // Optionnel: n'était pas demandé
    g.drawString("Voici un cercle rouge avec un remplissage jaune", 0, 450);
}
```

-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-

Le professeur
Bernard Lanctôt
le 25 février 2003