

École Polytechnique
Département de Génie Informatique
INF-4100, Sujets spéciaux: le langage JAVA
Examen intrasemestriel: Automne 2002

Note: *les solutions données ici sont celles du professeur, et ne sont pas obligatoirement les seules acceptables pour fin de correction de cet examen. Toute autre solution qui répond au problème posé sera évaluée à sa juste valeur.*

Question 1: (2 points)

Si on inclut une méthode appelée "init" dans une application JAVA, en quoi cette méthode sera-t-elle différente d'une méthode du même nom incluse dans un applet?

Réponse:

Dans le cas d'un applet, la méthode init sera appelée implicitement par l'exécuteur; elle sera ainsi appelée une fois lors du lancement de l'applet, et normalement, ne serait plus appelée par la suite.

Dans le cas d'une application, une méthode init est une méthode comme les autres; elle doit être appelée explicitement par une autre méthode, et pas nécessairement au début; elle peut être appelée plusieurs fois.

Question 2: (3 points)

Un programme lit une ligne de données contenant un nombre indéterminé de valeurs entières séparées par des virgules et terminée par le symbole de fin de ligne '\n'. Ces valeurs doivent être placées dans un tableau dont la taille est exactement égale au nombre de valeurs. La ligne est un objet de classe String appelé LaLigne et le tableau se nommera LeTablo. Écrivez les instructions requises pour créer ce tableau et y placer les valeurs lues.

Réponse:

```
StringTokenizer ST = new StringTokenizer(LaLigne, ",\n");
int LeTablo[] = new int[ST.countTokens()];
for(int i=0; i<LeTablo.length; i++)
    try
    {
        LeTablo[i] = Integer.parseInt(ST.nextToken());
    } catch (NumberFormatException E){ };
```

Question 3: (5 points)

Une application en sécurité informatique requiert l'usage d'un encrypteur/décrypteur de type RC4. On désire créer une classe appelée RC4 qui contiendra les variables privées entières I et J initialisées à zéro de même qu'une variable entière n, un tableaux d'entiers S et un tableau d'octets K. Toutes ces variables doivent être inaccessi-

bles à d'autres modules de l'application. Le constructeur reçoit deux valeurs: un entier qui est la valeur que doit avoir n, et un objet de classe String dont les caractères devront être conservés sous forme d'octets dans le tableau K. Le tableau S est de taille 2^n , et devra être initialisé de la façon suivante:

$$S[i] = i; \quad // \text{ pour } i \text{ de } 0 \text{ à } 2^n \text{ exclusivement}$$

Écrivez les instructions requises pour déclarer la classe et ses variables de classes, ainsi que le constructeur.

Note: la valeur de 2^n peut se calculer en effectuant un décalage de la constante 1 de n position vers la gauche.

Réponse:

```
class RC4
{
    private int I, J, n, S[ ];
    private byte K[ ];

    public RC4(int nn, String SS)
    {
        I = J = 0;
        n = nn;
        S = new int[1 << n];
        for (int i = 0; i < S.length; i++) S[i] = i;
        K = SS.getBytes();
    }
}
```

Question 4: (5 points)

Chaque ligne du fichier fournie par le Service Informatique pour communiquer la liste des étudiants inscrits à un cours se présente sous la forme suivante:

Matr: 999999 "Nom de l'étudiant" <Adresse de courriel>

(Note explicative: 999999 sert à indiquer qu'il s'agit d'un champs numérique; le nom de l'étudiant est effectivement inscrit entre guillemets et l'adresse entre petits crochets)

Écrivez une méthode publique qui reçoit en argument une ligne de ce fichier sous forme d'un objet de classe String, et retourne un objet de classe String dont la valeur est le nom de l'étudiant excluant les guillemets. Si la ligne ne commence pas par Matr:, ou si elle ne contient pas un nom sous la forme indiquée, la méthode retourne la constante null.

Réponse:

```
public String TrouveNom(String L)
{
    if(!L.startsWith("Matr:")) return null;
    int G1 = L.indexOf(" ");
```

```

        if(G1 < 0) return null;
        int G2 = L.indexOf("\n", G1 + 1);
        if(G2 < 0) return null;
        return L.substring(G1+1, G2);
    }

```

Question 5: (5 points)

Écrivez une méthode publique qui fait effectuer les opérations suivantes:

- ?? activer la fenêtre de dialogue permettant de faire la sélection d'un fichier
- ?? vérifier si ce fichier existe vraiment
- ?? si oui retourner un objet de classe File qui est le fichier sélectionné; si non, retourner la constante null.

Réponse:

```

public File Question5( )
{
    JFileChooser JF = new JFileChooser( );
    JF.setFileSelectionMode(JFileChooser.FILES_ONLY);
    int Res = JF.showOpenDialog(this);
    File FichierChoisi = JF.getSelectedFile( );
    if(FichierChoisi.exists( )) return FichierChoisi;
    else return null;
}

```

Le professeur
Bernard Lanctôt