

Session du 31 janvier 2013

Aucun document n'est autorisé. Le barème est donné à titre indicatif.

Exercice 1 : exécution de programme (3 points)

Retracez l'exécution du programme suivant en indiquant l'évolution des valeurs des variables manipulées par ce programme. Vous pourrez faire cela au moyen d'un tableau comportant une colonne pour chaque variable, en précisant sur chaque ligne du tableau le numéro de ligne de code exécutée.

```
1 public class Ex13_3{
2     public static void main(String [] args){
3         int x = 0;
4         String st = "abc";
5         boolean bool = true;
6         char ch1 = 'b';
7         char ch2 = 'x';
8         for (int i=0; i<5; i++){
9             bool = (i % 2 == 0);
10            if (bool == (i<3)){
11                st = st + ch1;
12            }else{
13                st = st + ch2;
14            }
15        }
16        Terminal.ecrireStringln("valeur_de_st:" + st);
17    }
18 }
```

On rappelle que % est l'opérateur modulo qui calcule le reste de la division entière. L'expression $(i \% 2 == 0)$ vaut true si i est pair et false si i est impair. `Terminal.ecrireStringln` affiche une chaîne de caractères à l'écran.

Exercice 2 : tableau de caractères (4 points)

Ecrivez un programme (méthode `main`) qui réalise successivement les opérations suivantes :

1. Déclaration et initialisation d'un tableau de 10 caractères en lisant les 10 valeurs au clavier.
2. Vérification que toutes les valeurs du tableau sont des lettres. Pour cela on pourra utiliser la méthode `Character.isLetter` qui prend en paramètre un caractère et renvoie true si et seulement si ce caractère est une lettre.
3. Calcul du nombre de majuscules et de minuscules dans le tableau. Vous pourrez utiliser la méthode `Character.isLowercase` qui prend en paramètre un caractère et renvoie true si et seulement si ce caractère est une lettre en minuscule, et la méthode `Character.isUppercase`

Exercice 3 : procédures et fonctions (3 points)

Donnez sans justification les affichages produits à l'exécution par le programme suivant :

```
public class ExoMethodes3{
    static int toDec(boolean b) {
        if (b) return 1; else return 0;
    }
    static void ch(int i, boolean b, boolean [] t) {
        if (0<= i && i <= t.length-1)
            t[i] = b;
    }
    static int eval(boolean [] tb) {
        int s = 0;
        int p = 1;
        for (int i=tb.length-1; i>=0; i--) {
            s = s + toDec(tb[i])*p;
            p= p*2;
        }
        return s;
    }
    static void af (boolean [] t) {
        for (int i=0; i<t.length; i++) {
            Terminal.ecrireString("_" + t[i]);
        }
        Terminal.sautDeLigne();
    }
    public static void main (String args[]) {
        boolean [] tab = {false, false, true};
        ch(3,true, tab);
        af(tab);
        ch(0,true, tab);
        af(tab);
        Terminal.ecrireStringln("resultat_=_"+ eval(tab));
    }
}
```

Exercice 4 : élections (5 points)

On souhaite utiliser un programme pour enregistrer les résultats d'une élection. Ces résultats donneront le nombre de voix obtenu par candidat et par bureau de vote. Les candidats sont Riri, Fifi et Loulou. Il y a 5 bureaux de votes.

1. définissez la ou les variables nécessaires à contenir ces données ainsi que le code nécessaire pour lire au clavier les données sur le vote.
2. écrivez une méthode qui calcule le pourcentage d'un candidat donné. Cette méthode ne doit rien afficher.
3. utilisez cette méthode pour faire afficher le pourcentage de vote obtenus par chacun des candidats.
4. affichez le nom du vainqueur.

Exercice 5 : nombres à virgule (5 points)

Un nombre à virgule est tapé au clavier au moyen d'une suite de caractères, un par ligne. Il peut y avoir des espaces avant le début du nombre et il y a au moins un espace après la fin du nombre. Les chiffres sont entrés sur des lignes consécutives. La partie décimale est séparée par une virgule (notation française). Tout caractère autre qu'un chiffre, un espace ou une virgule est une erreur.

Voici un exemple d'entrée correcte du programme :

```
5
6
,
7
0
5
```

Ecrivez un programme (méthode main) qui transforme une telle entrée en une valeur de type double stockée dans une variable et affichée à la fin du programme.

Chaque caractère peut être lu par la méthode `Terminal.lireChar` ou en utilisant un scanner.

Un caractère représentant un chiffre peut être converti en un nombre entier au moyen de la méthode suivante `Character.getNumericValue` qui prend en argument un caractère et renvoie l'entier correspondant.

On ne demande pas de traiter les erreurs, notamment si les caractères lus ne forment pas un nombre à virgule.