

**Licence Informatique troisième année**

GROUPE DE TD : lundi matin    lundi après midi    mercredi    jeudi

**Contrôle continu n°1**

**MLL5U20**

**Durée 0H45**

**Epreuve sans document**

**18/10/2016 9H45**

**Ne pas consulter le sujet  
avant le signal donné  
par le responsable de l'épreuve**

**Les stylos doivent être posés à la fin du temps réglementaire.**

**Aucune sortie autorisée pendant cette épreuve.  
Barème indicatif sur 10**

Question 1 (2 points)

Soit le programme suivant :

```
package P1 ;
public class C {public static int A(int B){int A1=0;int F=A1;for(int
D=A1;D<=B;D=D*8/8+1)F=D+F;return F;}}
```

Récrire ce programme en utilisant les conventions d'usage en Java, et en en faisant un programme lisible et commenté (javadoc).

Question 2 (2 points)

On considère une méthode static de la classe Util nommée `verifierMotDePasse`. Cette méthode prend en paramètre deux chaînes de caractères : une chaîne représentant le login et une autre le mot de passe. Son rôle est de vérifier que le mot de passe correspond bien au login. Si ce n'est pas le cas, elle lance une exception `MotDePasseIncorrectException`.

```
public static void verifierMotDePasse(String login,String password)
                                throws MotDePasseIncorrectException
```

Une méthode `saisirLoginMotDePasse` de cette même classe Util prend en paramètre un terminal et demande à l'utilisateur son login et son mot de passe. Elle réitère cette demande tant que la vérification échoue. Elle rend au final le login pour lequel le mot de passe a été validé.

Définir la méthode. `saisirLoginMotDePasse`

**Question 3 (1,5 point) (cocher les cases correspondant à votre réponse)**

Bonne réponse : +0,25 point   mauvaise réponse : -0,25 point   pas de réponse : 0

	VRAI	FAUX
List est une interface générique		
Une classe interne a accès aux attributs privés de la classe englobante		
Toutes les méthodes d'instance d'une classe abstraites sont abstraites		
Une interface peut dériver d'une classe abstraite		
Truc truc = ... : Sachant que Truc est une classe non abstraite, on peut en déduire que l'objet référencé par truc peut être une instance directe de Truc.		
Maven permet entre autre la production d'un site Web de reporting.		

**Question 4 (1 point)**

Quel est l'affichage produit par l'exécution de la fonction main suivante :

```
package up5.mi.pary.premiercc2016;
```

```
public class PasPar {
```

```
    public static String foncTestPassageParametre  
        (String str1,StringBuilder sb1,String str2,StringBuilder sb2){  
        str1.concat("de");  
        sb1.append("gh");
```

```

    str2=str1;
    sb2=sb1;
    return str1;
}

public static void main(String[]args){
    String str1="abc";
    StringBuilder sb1 = new StringBuilder("def");
    String str2="A";
    StringBuilder sb2 = new StringBuilder("B");
    String res=foncTestPassageParametre(str1,sb1,str2,sb2);
    System.out.println(str1.length()+" "+sb1.length()+
        " "+str2.length()+" "+sb2.length()+
        " "+res.length());
}
}

```

#### Question 5

On considère la méthode main suivante :

```

public static void main(String [] args) throws UnknownStudentException {

    List<Note> list = new ArrayList<>() ;
    list.add(new Note(15,2));
    list.add(new Note(10,1));
    list.add(new Note(10,1));
    System.out.println(list+ " "+UtilNote.getMoyennePonderee(list)) ;
}

```

Affichage produit :

[15.0(coeff 2.0), 10.0(coeff 1.0), 10.0(coeff 1.0)] 12.5(coeff 4.0)

La classe Note a deux attributs privés : une valeur (**entre 0 et 20**) et un coefficient (**entre 1 et 10**).

Question 5.1 Compléter la définition de la classe Note (1 point)

```

package up5.mi.pary.jc.premierControle2016V;

public class Note {

    private double valeur;
    private double coeff;
}

```

```
public Note(double valeur, double coeff) throws IllegalArgumentException {
```

```
    this.valeur = valeur;  
    this.coeff = coeff;  
}
```

```
public double getValeur() {  
    return valeur;  
}  
public double getCoeff() {  
    return coeff;  
}
```

```
public String toString(){
```

```
    }  
}
```

Question 5.2 Compléter la définition suivante : (1 point)

```
package up5.mi.pary.jc.premierControle2016V;
```

```
import java.util.List;
```

```
public class UtilNote {  
    public static Note getMoyennePonderee (List<Note> list) {  
        double somme=0,coeff=0;  
        for (Note note:list){
```

```
    }  
}
```

On souhaite maintenant utiliser une classe ListNote :

```
package up5.mi.pary.jc.premierControle2016V;
```

```
import java.util.ArrayList;  
import java.util.List;
```

```

public class Main {

    public static void main(String [] args) throws UnknownStudentException {

        ListNote listNote = new ListNote() ;
        listNote.add("Dupond",15,2);
        listNote.add("Dupond",10,1);
        listNote.add("Dupond",10,1);
        listNote.add("Durand",17,2);
        listNote.add("Durand",15,2);
        System.out.println("Moyenne de Dupond="+listNote.getMoyenne("Dupond")) ;
        System.out.println("Moyenne de Durand="+listNote.getMoyenne("Durand")) ;
        System.out.println("Moyenne de Dupont="+listNote.getMoyenne("Dupont")) ;

    }
}

```

// AFFICHAGE PRODUIT lors de l'exécution de la méthode main :

Moyenne de Dupond=12.5(coeff 4.0)

Moyenne de Durand=16.0(coeff 4.0)

Exception in thread "main" [up5.mi.pary.jc.premierControle2016V.UnknownStudentException](#):  
Etudiant inconnu : Dupont

at up5.mi.pary.jc.premierControle2016V.ListNote.getMoyenne([ListNote.java:25](#))

at up5.mi.pary.jc.premierControle2016V.Main.main([Main.java:25](#))

Question 5.3 (1,5 points)

Définir la méthode add de la classe ListNote

```
package up5.mi.pary.jc.premierControle2016V;
```

```
import java.util.ArrayList;
```

```
import java.util.HashMap;
```

```
import java.util.List;
```

```
import java.util.Map;
```

```
public class ListNote {
```

```
    /**
```

```
     * une map dont les clefs sont des noms d'étudiants et la valeur est la liste des  
Note d'un étudiant
```

```
     */
```

```
    Map<String,List<Note>> map = new HashMap<>();
```

```
    /**
```

```
     * ajoute une note d'une valeur et d'un coefficient donné pour un étudiant donné
```

```
     * @param student l'étudiant concerné
```

```
     * @param valeur la valeur de la note entre 0 et 20
```

```
     * @param coeff le coefficient de la note entre 1 et 10
```

```
     */
```

```
    public void add(String student, double valeur, double coeff) {
```

```

    }
/**
 * rend la moyenne des notes d'un étudiant
 * @param student l'étudiant pour lequel on veut la moyenne
 * @return la Note correspondant à cette moyenne
 * @throws UnknownStudentException si l'étudiant n'a pas de note
 */
    public Note getMoyenne(String student) throws UnknownStudentException{
        ...
    }
}

```