

L3 Mathématiques, Informatique MLL5U20

Examen

Durée 1H30

Epreuve sans document

15/06/16

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

A la fin du temps réglementaire :

- les stylos doivent être posés**
- les copies sont regroupées en bout de rangée**

**Aucune sortie autorisée pendant la première heure
et pendant les 10 dernières minutes de l'épreuve.**

Question 1 : 3 points(cocher les cases correspondant à votre réponse)

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

Java	VRAI	FAUX
Il est impossible de passer des fonctions en paramètre à d'autres fonctions		
Une classe abstraite peut avoir comme membres des attributs abstraits		
Thread	VRAI	FAUX
La méthode start de la classe Thread est souvent redéfinie dans les classes dérivées de Thread		
La méthode run de la classe Thread est souvent redéfinie dans les classes dérivées de Thread		
Servlet	VRAI	FAUX
Les servlets ont une méthode main qui est exécutée lors d'une requête du client		
Chaque nouvelle requête donne lieu à la création d'un thread		
Chaque nouvelle requête donne lieu à la création d'une instance de servlet		
Entrées-sorties	VRAI	FAUX
Les classes de filtre comme <code>BufferInputStream</code> ou <code>DataInputStream</code> ont un constructeur prenant en paramètre un flux d'entrée.		
L'interface <code>Serializable</code> déclare une méthode qui est la méthode permettant de sérialiser.		
Beaucoup de méthodes ont dans leur entête <code>throws IOException</code> . Cela signifie que des erreurs de type <code>IOException</code> peuvent être lancées lors de l'exécution de ces méthodes sans y être attrapées.		
Divers	VRAI	FAUX
JUnit définit les unités utilisées en Java		
Un intérêt de Maven est de permettre la récupération automatique des jars utiles pour un projet		

Question 2**On considère la méthode main suivante :**

```

public static void main(String[] args) throws PileVideException {
    Pile pile1 = new Pile ("Pile 1");
    String [] tab = {"a","b","c"};
    Pile pile2 = new Pile ("Pile 2",tab);
    System.out.println(pile2.pop()); // c
    pile1.push("a");
    pile1.push("b");
    pile1.push("c");
    System.out.println(pile1.isPileVide()); // false
    System.out.println(pile1.pop()); // c
    System.out.println(pile1.getNom()); // Pile 1
    System.out.println(pile1); // [a , b]
    System.out.println(pile1.pop()); // b
    pile1.push("d");
    System.out.println(pile1); // [a , d]
    System.out.println(pile1.pop()); // d
    System.out.println(pile1.pop()); // a
    System.out.println(pile1.isPileVide()); // true
    System.out.println(pile1.pop()); // une exception PileVideException
    est lancée
}

```

Question 3 (7 points) Répondre à cette question sur la copie d'examen

On peut déduire de la lecture de la fonction main, les constructeurs et méthodes publiques de la classe Pile.

- a) Donner l'interface commentée de la classe Pile (on entend par interface commentée les entêtes commentées de chacun des constructeurs et méthodes publiques de la classe Pile)
- b) Définir soigneusement :
 - la classe PileVideException
 - la classe Pile (2 constructeurs, 5 méthodes)
 - o contrainte : la classe Pile a deux attributs d'instances : une String et une liste de String

Il sera tenu compte pour la correction

- **de la qualité du code**
- des commentaires des membres publics et des attributs
- du respect des conventions de nommage

Question 3 (2 points)

On remplace dans la méthode main précédente les trois premières instructions par :

```

Pile<String> pile1 = new Pile<>("Pile 1");
Integer [] tab = {new Integer(4),new Integer(5)};
Pile<Integer> pile2 = new Pile<>("Pile 2",tab);

```

Dites avec précision ce que cela modifie par rapport à la question précédente.

Question 4 (3 points) JUNIT

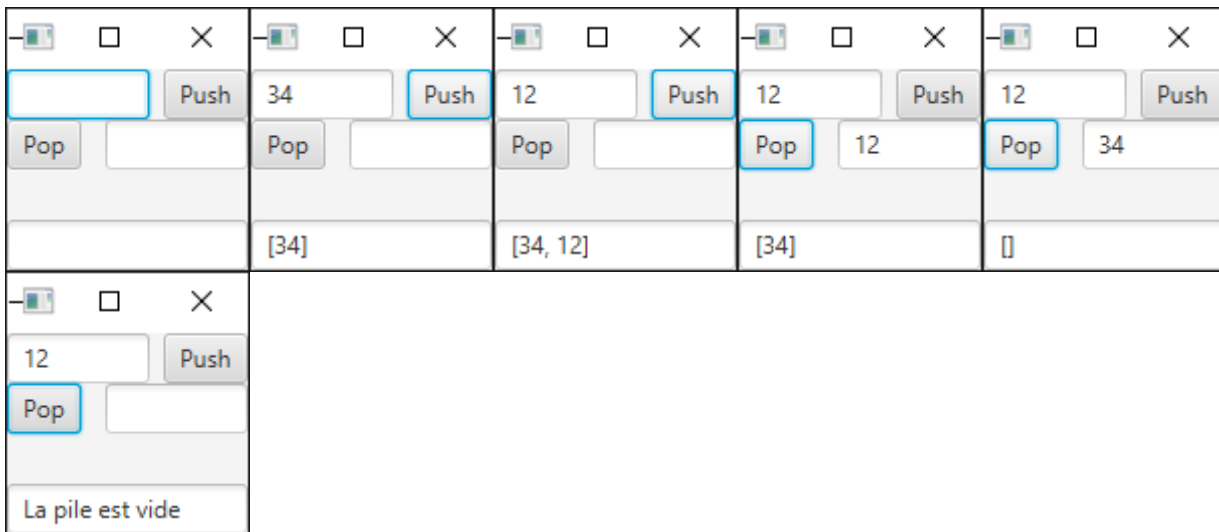
Faire une classe de tests unitaires JUnit pour la classe Pile.

On définira deux méthodes de test :

- la première fera trois push puis 3 pop en vérifiant que les résultats obtenus par pop sont bien les résultats attendus
- la deuxième vérifiera que le pop sur une pile vide lance bien une exception

Question 5 JavaFX (5 points)

On s'intéresse à une interface graphique JavaFX permettant l'utilisation d'une pile.



Ci dessus 6 impressions écran d'une interface JavaFX

1. Au lancement de l'application
2. L'utilisateur met 34 dans la pile en cliquant sur Push. (La pile est affiché en bas de l'interface)
3. L'utilisateur met 12 dans la pile en cliquant sur Push
4. L'utilisateur a cliqué sur pop (l'élément rendu par pop est affiché à droite du bouton)
5. L'utilisateur a cliqué sur pop
6. L'utilisateur a cliqué sur pop (il n'y a plus d'élément à rendre : un message d'erreur est affiché en bas de l'interface)

Voici le programme correspondant.

Compléter le constructeur de la classe PilePane.

```
public class Appli extends Application{
    @Override
    public void start(Stage primaryStage) throws Exception {

        Pile<String> pile = new Pile<>("JavaFX");
        Pane pane = new PilePane(pile);
        Scene scene = new Scene(pane,100,100);
        primaryStage.setTitle("2016");
        primaryStage.setScene(scene);
        primaryStage.show();

    }

    public static void main(String[] args) {
        Launch(args);
    }
}
```

```

public class PilePane extends BorderPane {

    public PilePane(Pile<String> pile){
        Button buttonPush=new Button("Push");
        TextField tfPush=new TextField();
        tfPush.setPrefColumnCount(5);
        Button buttonPop=new Button("Pop");
        TextField tfPop=new TextField();
        tfPop.setPrefColumnCount(5);
        TextField console = new TextField();
        console.setPrefColumnCount(50);

        // A COMPLETER

    }

}

```

Question 6 (3 points) Base de données

On souhaite mettre le contenu d'une pile de string en base de données.
 Une table, tablepile, a deux champs ordre et valeur.
 A chaque élément de la pile correspondra une ligne de la table tablepile.
 Le champ ordre est un entier indiquant la position de l'élément dans la liste
 et valeur est la valeur de l'élément.

```

public static void popToBd(BD bd,Pile<String> pile) throws SQLException{
    try (
        Connection connection = bd.getConnection();
        PreparedStatement ps = connection.prepareStatement(
            "insert into tablepile (ordre,valeur) VALUES(?,?)")
    {
        int ordre=0;
        // A COMPLETER

```

```

    }
    catch (PileVideException e) {
        e.printStackTrace();
    }
}

```