

# Examen de programmation par objets

Septembre 2000 - UE2241 MBE

Licence informatique Resp. J.Ferber - Durée : 2 h

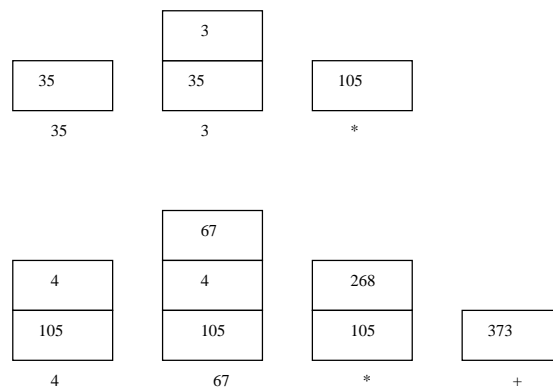
Tous documents manuels (cours, polys) autorisés. Livres non autorisés

## Calculatrice en notation polonaise postfixée

On se propose d'écrire un programme de calculatrice qui fonctionne en notation polonaise postfixée (comme le font encore certaines calculatrices Hewlett Packard). On rappelle que la notation polonaise postfixée place les opérateurs après les opérands, et qu'elle offre l'avantage de pouvoir s'affranchir des parenthèses. Par exemple, une opération de la forme  $(35 * 3) + (4 * 67)$  (qui donne le résultat 373) s'exprime ainsi en notation polonaise postfixée : 35 3 \* 4 67 \* +

### Première partie

L'interprétation de telles expressions nécessite l'utilisation d'une pile. Ainsi, l'interprétation de l'expression précédente passe par les états de pile ci-contre:



### Question A

On décide de représenter une pile d'entiers sous la forme d'un vecteur. Sachant qu'une pile est définie par les méthodes suivantes :

```
//empile un élément au sommet de la pile
```

```
void empiler(int x){ ... }
```

```
// retourne l'élément qui est au sommet de la pile
```

```
// en le supprimant de la pile
```

```
int depiler(){ .. }
```

```
// retourne l'élément qui est au sommet de la pile sans le supprimer
```

```
int sommet(){...}
```

Ecrire la classe `Pile` (donc avec les variables, les constructeurs si nécessaires et le code des méthodes).

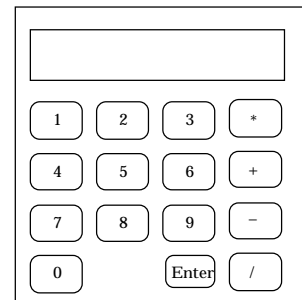
### Question B

On veut que les méthodes `depiler` et `sommet` retournent une erreur si la pile est vide. Implémenter, en utilisant les exceptions, les mécanismes permettent de récupérer une erreur de type `PileVideErreur`. Exemple :

```
    Pile p = new Pile() ;
    ...
    try{
        int v = p.depiler() ;
    }
    catch(PileVideErreur p){
        System.out.println("la pile est vide");
    }
}
```

### Deuxième partie

L'interface de la calculatrice doit avoir la forme ci-contre :



### Question C

Ecrire le code de la "Frame" de telle manière que l'on obtienne la mise en page suivante en n'utilisant que des gestionnaire de layout. Pas de positionnement à l'aide de coordonnées absolues. En particulier les méthodes `setLocation` ou `setSize` ne doivent pas être utilisées.

### Question D

On suppose que la saisie des nombres se fait uniquement par les boutons d'interface (On suppose qu'on ne peut pas entrer des expressions par le clavier de l'ordinateur). Pour simplifier, on supposera de plus qu'on ne manipule que des entiers.

Voici un tableau qui donne, pour chaque touche tapée, ce qui est visualisée dans l'afficheur. Voici ce qui se passe pour entrer l'expression `35 3 * 4 67 * + :`

Touche tapée	Expression visualisée	Touche tapée (suite)	Expression visualisée (suite)
3	3	enter	4
5	35	6	6
enter	35	7	67
3	3	*	268
*	105	+	373
4	4		

Ecrire le code de la calculatrice de façon à ce qu'elle fonctionne comme cela est spécifié. On utilisera la pile de la question B (la pile qui gère les exceptions). Les erreurs seront simplement envoyées dans le flux de sortie standard.

### Question E

On veut ajouter une "mémoire" qui sera manipulée par deux touches `STO` et `RCL`. La première mémorise le sommet de pile dans la mémoire, et la seconde place la mémoire sur la pile. Modifier le code de la calculatrice pour qu'elle prenne en compte ces deux nouvelles opérations (vous ne mettez que ce qui change par rapport à la question D).