UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences
Département d'Informatique

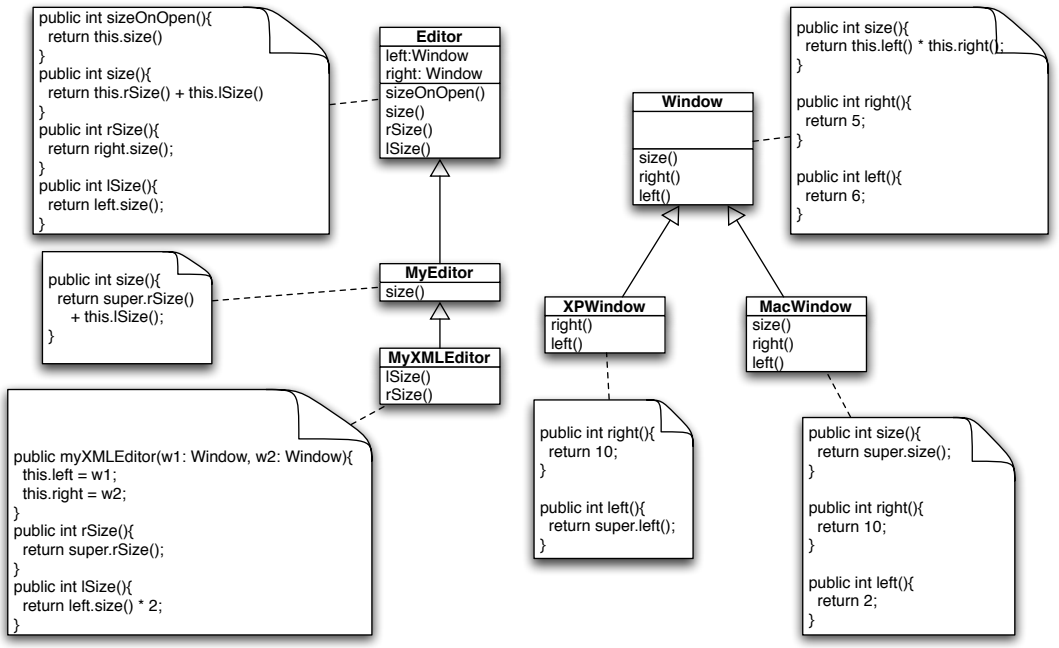
27 août 2008

INFO F 407 – Génie logiciel et gestion de projets
Examen de seconde session

Remarques préliminaires Cet examen dure quatre heures et se déroule à livre ouvert. Veillez à répondre sur une feuille séparée pour chaque question, en indiquant vos nom et prénom ainsi que le numéro de la question. Si vous ne connaissez pas la réponse pour une question, rendez une page vide avec les mêmes informations, mais en indiquant que vous n'avez pas de réponse.

Question 1 (OOP/Java) - 3

Soit le diagramme de classes :



et le code source Java :

```
MyEditor editor = new MyXMLEditor(new XPWindow(), new MacWindow());
editor.sizeOnOpen();
```

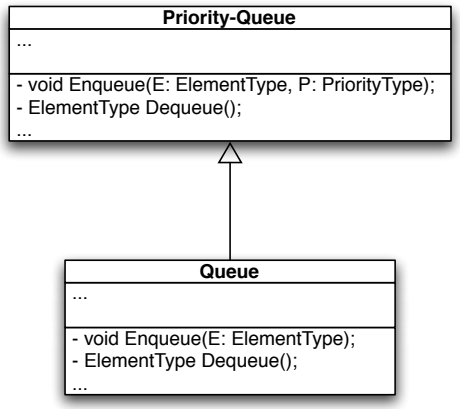
1. Donnez le résultat de ce programme ;
2. Donnez le flux d'appel des méthodes et les destinataires correspondants. Vous pouvez construire un diagramme de collaboration (« collaboration diagram ») ou un diagramme de séquences (« sequence diagram ») modélisant le comportement de ce code source ;

3. Quelles sont les conséquences d'utiliser le mot-clé `super`, resp. d'utiliser le mot-clé `this` dans une méthode qui peut être redéfinie dans une sous-classe ?

Question 2 (OOP/Héritage) - 4

Albert découvre dans une librarie la classe `Priority-Queue` qui lui offre des opérations comme `Enqueue` et `Dequeue`. L'opération `Enqueue` a comme arguments un élément (`ElementType`) et une priorité (`PriorityType`) et enregistre cet élément avec sa priorité. L'opération `Dequeue` n'a pas d'arguments et l'idée est que l'élément avec la plus grande priorité est retourné.

Albert veut développer une simple file (queue) basée sur le principe : Premier arrivé, premier servi. Alors, il décide de coder une classe `Queue` qui est une sous-classe de la classe `Priority-Queue` qui existe déjà. L'idée est exprimée dans le diagramme de classes suivant :



1. Expliquez comment on peut construire les opérations essentielles d'une file (`Enqueue`, `Dequeue`, ...) en utilisant les opérations de la classe Priority-Queue (quelques phrases suffisent).
2. Quel type d'héritage est utilisé dans ce cas-ci ? Est-ce que le principe de substitution est respecté ou non ? Motivez votre réponse.
3. Quelle est la différence entre une sous-classe et un sous-type ? Est-ce que c'est possible de spécifier cette différence en Java ?
4. Paola vérifie la solution d'Albert et remarque qu'une autre solution est possible : coder la file comme une classe qui a une priority-queue comme attribut. Argumentez cette solution et donnez des avantages et désavantages.

Question 3 (Requirements Engineering, Processus) - 1

La documentation des exigences est primordiale. Elle sert à présenter les exigences d'une manière structurée. Cette documentation est encore plus primordiale quand un entrepreneur externe développe le système. Les méthodes agiles ne produisent pas de la documentation des exigences. Quelle est leur motivation ?

Question 4 (Refactorisation) - 2

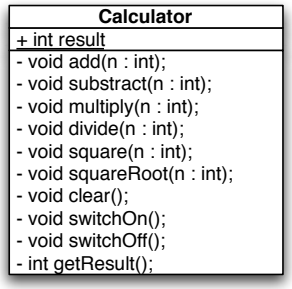
Dans le cours on a vu la phrase suivante :

Refactoring also fits naturally in the agile methods philosophy.

Argumentez ce point de vue.

Question 5 (Tests d'unité/JUnit) - 4

Sur base du code source ci-dessous, écrivez des tests d'unité en JUnit4 (« unit tests ») permettant de tester le comportement de la classe `Calculator`. Énoncez et expliquez les bogues que vos tests devraient découvrir.



Définition de la classe `Calculator` :

```
public class Calculator {

    private static int result;

    public void add(int n) {
        result = result + n;
    }

    public void substract(int n) {
        result = result - 1;
    }

    public void multiply(int n) {
    }


    public void divide(int n) {
        result = result / n;
    }

    public void square(int n) {
        result = n * n;
```

```
    }

    public void squareRoot(int n) {
        for (; ;) ;
    }

    public void clear() {
        result = 0;
    }

    public void switchOn() {

        result = 0;
        System.out.println("...Switch On...");
    }

    public void switchOff() {
        System.out.println("...Switch Off...");
    }

    public int getResult() {
        return result;
    }
}
```

Question 6 (Gestion/Processus de développement) - 2

Une activité assez importante en gestion de projets est l'estimation du coût, de l'effort et des ressources nécessaires. Il existe plusieurs stratégies et techniques d'estimation. Comment est-ce que l'estimation de l'effort (et des ressources) nécessaires est fait si on utilise la programmation extrême (Extreme Programming) comme méthodologie de développement ?

Question 7 (Patterns/ Conception) - 2

Eclipse est une plateforme ouverte aux *plug-ins* et alors permet de développer des plug-ins.

1. Expliquez ce qu'est ce concept de plug-in ;
2. Donnez les avantages ainsi que les désavantages.

Question 8 (Qualité de code) - 2

1. Expliquez les problèmes qu'on a avec du 'duplicated code', et comment on peut résoudre ces problèmes en général.
2. Motivez dans ce contexte et dans le contexte d'héritage l'utilisation des méthodes courtes.