

Introduction à la programmation objet – Examen – Janvier 2023

Durée 2h. Notes personnelles et documents de cours autorisés.

Barème approximativement proportionnel au temps nécessaire à la résolution de chaque question.

On considère un document, par exemple une page web, composé de différents éléments rectangulaires. Certains éléments sont juxtaposés, d'autres sont emboîtés. On veut réaliser un système d'affichage pour un tel document.

Classe Element. Tous nos éléments vont étendre la classe abstraite suivante.

```
public abstract class Element {
    public abstract int getWidth();
    public abstract int getHeight();

    // Affiche l'élément aux coordonnées (x, y)
    public abstract void draw(Graphics g, int x, int y);

    // Affiche le cadre de l'élément, aux coordonnées (x, y)
    protected void drawBorder(Graphics g, int x, int y) {
        g.drawRect(x, y, getWidth(), getHeight());
    }
}
```

1. Que signifient les différentes occurrences du mot **abstract** dans ce fragment de code? Si les occurrences n'ont pas toutes la même signification, préciser ce qui s'applique à chacune.
2. Voici une erreur que l'on peut observer en travaillant avec la classe Element.

```
~/Main.java:5
java: Element is abstract; cannot be instantiated
```

Cette erreur apparaît-elle à la compilation ou à l'exécution? Proposer un fragment de code susceptible de la produire, et expliquer ce qui doit être fait pour l'éviter.

3. Que signifie le mot **protected**? À quoi pourra-t-il servir?
4. Dire dans quelle classe est définie la méthode drawRect, et donner son type.
Note : pas besoin de documentation ici, le fragment de code donné suffit à répondre à la question.

Élément simple. Un élément simple est un élément rectangulaire de dimensions fixées. Pour afficher un tel élément, on se contentera d'afficher son cadre.

5. Donner le code d'une classe ElementSimple répondant à la description précédente. On doit pouvoir créer un tel élément avec le code `new ElementSimple(width, height)`.
Note : savoir quelles méthodes doivent être définies fait partie de la question.

Interface Conteneur. On se donne l'interface suivante pour des boîtes contenant plusieurs éléments.

```
interface Conteneur {
    void add(Element e);
    List<Element> getElements();
}
```

6. Une mauvaise utilisation de l'interface Conteneur peut mener à l'erreur suivante.

```
~/MonConteneur.java:5
java: add() in MonConteneur cannot implement add() in Conteneur
    attempting to assign weaker access privileges (package-private); was public
```

Cette erreur apparaît-elle à la compilation ou à l'exécution? Proposer un fragment de code susceptible de la produire, et expliquer ce qui doit être fait pour l'éviter.

7. En l'état, un Conteneur peut-il contenir d'autres Conteneurs? Expliquer.

Conteneur horizontal. On veut représenter une boîte contenant plusieurs éléments disposés horizontalement, l'un à la suite de l'autre. La largeur d'une telle boîte est déterminée par la somme des longueurs des éléments contenus, et sa hauteur est déterminée par le plus haut des éléments contenus.

8. On veut définir une classe `ConteneurH` qui concrétise à la fois `Conteneur` et `Element`. Compléter la première ligne de la définition :

```
public class ConteneurH ...
```

9. Proposer des attributs pour la classe `ConteneurH`, et définir un constructeur créant un conteneur initialement vide et la méthode `add`.
10. Donner le code des méthodes `getWidth` et `getHeight`, calculant les dimensions du conteneur.
11. Donner le code de la méthode `draw`, de sorte à ce que qu'elle affiche chaque élément contenu, mais pas le cadre du conteneur lui-même. Les éléments doivent être juxtaposés, et alignés sur la même abscisse y .
12. On souhaite ajouter un espacement configurable entre deux éléments consécutifs. Décrire une stratégie pour réaliser ceci : quels nouveaux attributs ? quelles nouvelles méthodes ? quelles modifications des méthodes déjà écrites ? *On ne demande pas d'écrire le code.*

Conteneur en lignes. On veut représenter une boîte contenant plusieurs éléments, mais dont la largeur est fixée. Les éléments sont disposés l'un à la suite de l'autre horizontalement, puis forment de nouvelles lignes lorsque la ligne courante est pleine. Voici un début de code pour cette classe.

```
public class ConteneurF extends Element implements Conteneur {
    private int width;
    private ArrayList<ConteneurH> lignes;
    public ConteneurF(int width) {
        this.width = width;
        this.lignes = new ArrayList<>();
    }
    ...
}
```

13. Expliquer comment les attributs choisis dans l'extrait précédent permettent bien de répondre au problème.
14. Donner le code des méthodes `getWidth` et `getHeight`.
15. On veut le comportement suivant pour la méthode `add(Element e)` : si la dernière ligne contient assez de place pour accueillir l'élément e , alors on ajoute e à cette ligne. Sinon, on l'ajoute à une nouvelle ligne. Écrire le code de cette méthode.
16. Dans quelle situation pourrait-on avoir un élément qui déborde d'un tel conteneur, malgré les précautions prises par `add` ? Quelle réponse peut-on apporter à ce problème ?
17. Écrire la méthode `getElements`, de sorte à ce qu'elle renvoie les éléments individuels du conteneur (ceux qui ont été ajoutés avec la méthode `add` de ce conteneur).
18. Écrire la méthode `draw` pour ce conteneur, de sorte que :
- le cadre du conteneur est dessiné,
 - chaque nouvelle ligne est dessinée sous la précédente,
 - chaque ligne est centrée horizontalement (c'est-à-dire : on doit laisser autant d'espace libre à gauche qu'à droite de la ligne).

Modification des dimensions. On veut ajouter à la classe `Element` une méthode abstraite `setWidth(int w)` pour modifier la largeur de l'élément.

19. Donner le code concret de `setWidth` pour la classe `ElementSimple`.
20. Cela n'a pas de sens de fixer la largeur d'un `ConteneurH`. Pourtant, la méthode `setWidth` doit être concrétisée dans cette classe également. Proposer un code adéquat.
21. Lorsque l'on modifie la largeur d'un `ConteneurF`, il faut rééquilibrer les lignes. Proposer une manière de procéder. *On ne demande pas de code.*
22. Lorsqu'un élément appartenant à un `ConteneurF` change de dimensions, on voudrait également rééquilibrer les lignes de ce conteneur. Expliquer pourquoi ce n'est pas possible avec les définitions données jusque là. Proposer une manière de procéder. *On ne demande pas de code.*