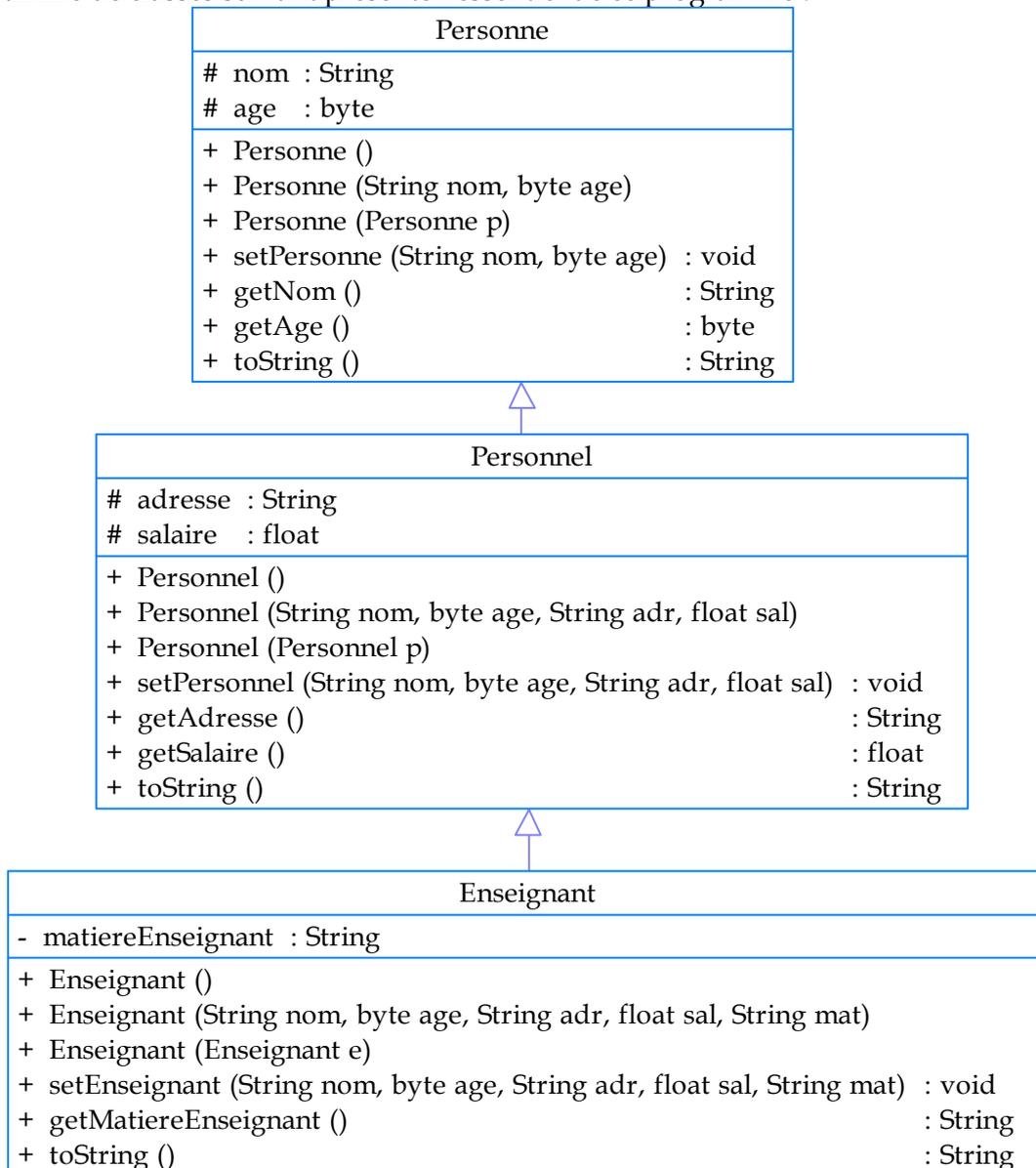


TD/TP 6 : héritage et polymorphisme

1. Gestion de personnel

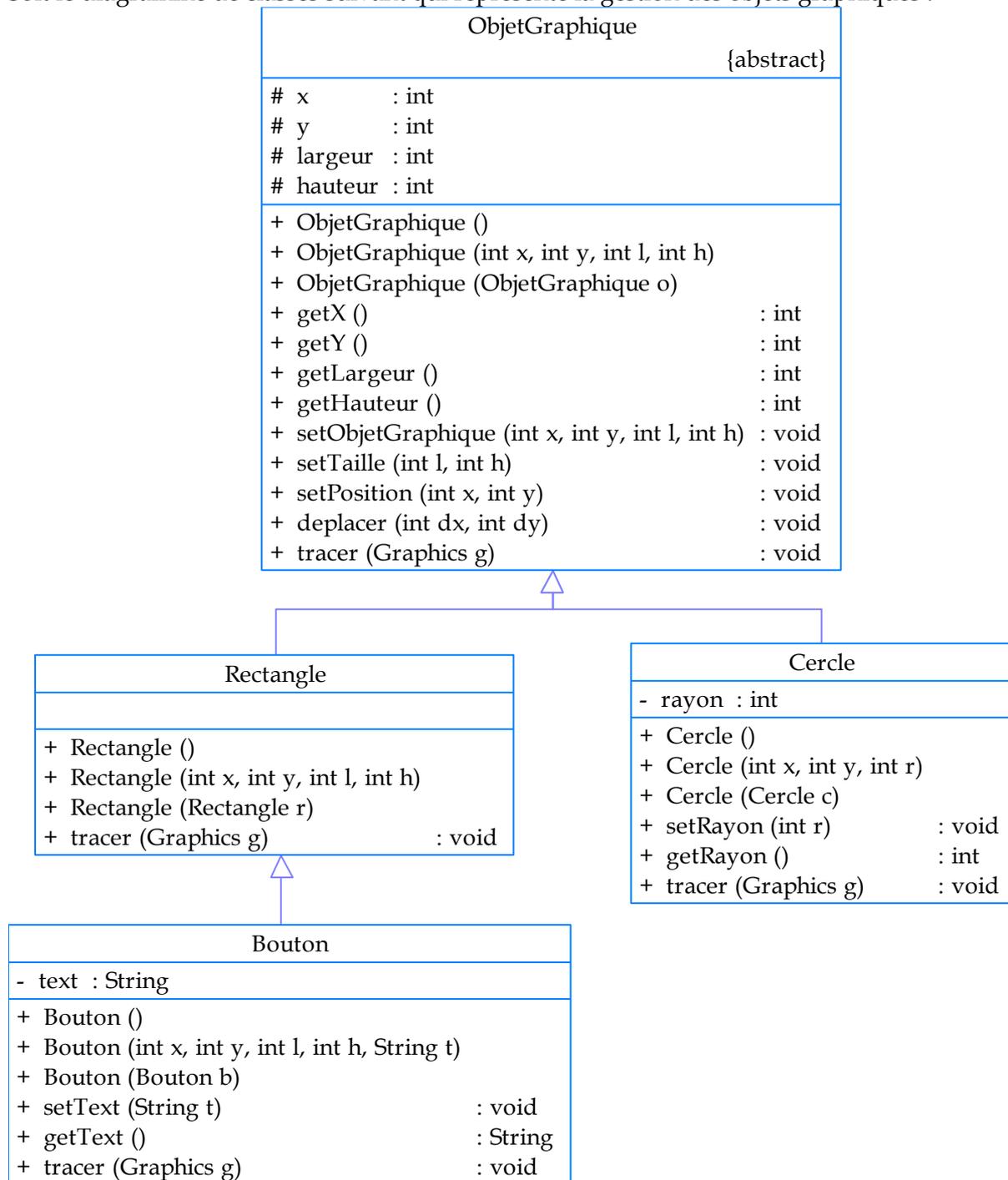
On désire réaliser un programme de gestion du personnel d'un centre de formation. Le diagramme de classes suivant présente l'essentiel de ce programme :



- Implémenter la classe **Personne** en définissant les différentes méthodes.
- Implémenter la classe **Personnel** qui hérite de la classe **Personne**.
- Implémenter la classe **Enseignant** qui hérite de la classe **Personnel**.
- Écrire une classe **TestPersonnel** qui contient la méthode main et qui permet de :
 - ✓ Créer un tableau hétérogène qui contient plusieurs objets de chacune de ces 3 classes.
 - ✓ Afficher les informations de chaque objet du tableau.

2. Objet graphique

Soit le diagramme de classes suivant qui représente la gestion des objets graphiques :



1. Implémenter la classe abstraite **ObjetGraphique**.
2. Implémenter la classe **Rectangle** qui hérite de la classe **ObjetGraphique**.
3. Implémenter la classe nommée **Bouton** qui hérite de la classe **Rectangle**, qui a une variable membre de type `String`, appelé `text`.
4. Implémenter la classe nommée **Cercle** qui hérite de la classe **ObjetGraphique**.
5. Écrire une classe de test qui permet de :
 - Créer un tableau hétérogène qui contient plusieurs cercles, rectangles et boutons.
 - Créer une fenêtre et tracer tous les objets du tableau.

6. Soit la classe **ListeObjetGraphique** suivante :

```
public class ListeObjetGraphique {
    private ObjetGraphique[] liste;
    private int taille;
    private int index;
    public ListeObjetGraphique(int taille) {
        this.taille = taille;
        liste = new ObjetGraphique[taille];
        index = 0;
    }
    public void ajouter(ObjetGraphique f) {
        if (index < taille) {
            liste[index] = f;
            index++;
        }
    }
    public void tracerObjetsGraphique (Graphics g) {
        for (int i = 0; i < index; i++)
            liste[i].tracer(g);
    }
}
```

Écrire une classe de test qui permet :

- Créer un objet de type **ListeObjetGraphique** qui permet de mémoriser jusqu'à 10 objets graphiques.
 - Ajouter plusieurs cercles, rectangles et boutons.
 - Créer une fenêtre et tracer tous les objets du tableau.
7. Le package *java.util* dispose des deux classes **ArrayList** et **ListIterator**, permettant la gestion de liste. Refaire la **question 6** en créant un objet de type **ArrayList** pour mémoriser les 10 objets graphiques.