

1. (1 pt) Soit le code suivant :

```
public class Position {  
    public enum Colonne { A, B, C, D, E, F, G, H }  
  
    private int    _ligne;  
    private Colonne _colonne;  
}
```

Écrivez une méthode `equals` qui retourne `true` si deux Positions sont identiques, `false` sinon. Deux positions sont identiques si les deux champs de la classe contiennent les mêmes valeurs.

```
public boolean equals( Object a_objet ) {  
    boolean resultat = false;  
  
    if( a_objet instanceof Position ) {  
        Position p = (Position) a_objet;  
        resultat = _ligne == p._ligne &&  
                   _colonne == p._colonne;  
    }  
  
    return resultat;  
}
```

2. Soit les classes et interfaces suivantes :

```
interface Iterable< E >  
interface Collection< E > implements Iterable< E >  
interface List< E > implements Collection< E >  
  
class AbstractCollection< E > implements Collection< E >  
class AbstractList< E > extends AbstractCollection< E > implements List< E >  
class ArrayList< E > extends AbstractList< E > implements Iterable< E >
```

(a) ( $\frac{1}{2}$  pt) La classe `ArrayList< E >` contient la méthode `addAll( Collection< E > c)`. Encerchez les classes se qualifiant comme argument à cette méthode.

- `Object` : non
- `AbstractCollection< E >` : oui
- `AbstractList< E >` : oui
- `ArrayList< E >` : oui

(b) ( $\frac{1}{2}$  pt) Nous écrivons une méthode ayant la signature suivante :

```
m2( ArrayList< List< E > > c)
```

Encerchez les classes se qualifiant comme argument à cette méthode.

- `ArrayList< Object >` : non
- `ArrayList< AbstractCollection< E > >` : non
- `ArrayList< AbstractList< E > >` : oui
- `ArrayList< ArrayList< E > >` : oui