

Examen de la matière P.O.O – 2^{ème} année licence Informatique

Questions : les réponses doivent être brèves !

- 1) Qu'elle est la différence entre une interface et une classe abstraite ?
- 2) Sachant qu'une classe abstraite ne peut pas être instanciée, est-ce que celle-ci peut avoir un constructeur ?
- 3) Un objet de type MAP possède-t-il un itérateur ? Pourquoi ?
- 4) Une application effectue beaucoup de modifications et nécessite une structure de type LIST, entre ArrayList et LinkedList, qu'allez-vous choisir ? Pourquoi ?

Exercice1

Pour les lignes 1;2;3; (mentionnées en commentaires dans le programme principal) vous dites si c'est juste en précisant ce qui sera affiché, en cas d'erreurs vous donnez la cause.

```
class kabir
{
    kabir (String b){System.out.println("1");}
    kabir (kabir o2){System.out.println("2");}
}
```

```
class saghir extends kabir{
    saghir (int a){this("3",a);System.out.println("3");}
    saghir (saghir o1){super(o1);System.out.println("4");}
    saghir (){super("5");System.out.println("5");}
    saghir (String b, int c){System.out.println("6");}
}
```

```
public class constructe {
    public static void main(String [] args){
        /*1*/ saghir objet1=new saghir();
        /*2*/ saghir objet2=new saghir(objet1);
        /*3*/ saghir objet3=new saghir(5);
    }
}
```

Exercice2

Indiquez pour chaque ligne s'il y a des fautes. En cas d'erreur vous indiquez, s'elle est signalée pendant la compilation ou bien l'exécution (indiquez par exemple si c'est un downcasting implicite ou explicite ; upcasting...).

```
class kabir{}
class saghir extends kabir{}
```

```
public class poly {
    public static void main(String [] args){
```

```
/*1*/ saghir x1= new saghir(); kabir y1=x1; x1= y1;
/*2*/ saghir x2= new saghir(); saghir y2= new saghir(); y2= (kabir)x2;
/*3*/ kabir x3= new saghir(); kabir y3= (kabir) new saghir(); saghir z3= y3;
/*4*/ saghir x4= (kabir) new saghir(); kabir y4= (saghir) new kabir();
/*5*/ kabir x5= new kabir(); saghir y5= (saghir) x5;
    }
}
```

Exercice3

Pour les lignes, 1;2;3;4;5;6;7;8 (mentionnées en commentaires dans le programme principal) vous dites ce qui va être affiché avec de très brèves explications.


```

class meree {
void A(){System.out.println("A de mere");B();}
void B(){System.out.println("B de mere");}
void C(){System.out.println("C de mere"); B();System.out.println("fin de C de mere");}
static void stat(){System.out.println("static de mere");}
}

```

```

class fillee extends meree {
void A(){System.out.println("A de fille");super.A();}
void B(){System.out.println("B de fille");}
void C(){System.out.println("C de fille");super.C();System.out.println("fin de C de fille");}
static void stat(){System.out.println("static de fille");}
}

```

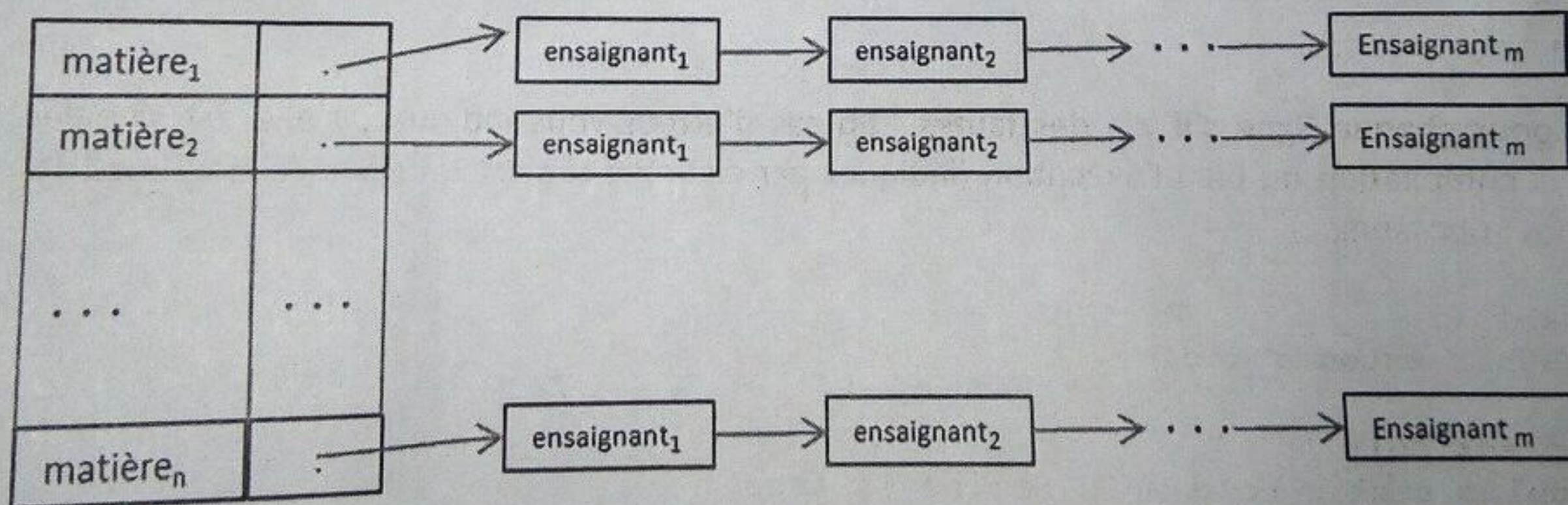
```

public class Main {
public static void main(String [] args) {
/*1*/   fillee obj=new fillee();
/*2*/   ((meree)obj).A();
/*3*/   obj.B();
/*4*/   ((meree)obj).stat();
/*5*/   meree objj= new meree();
/*6*/   objj.stat();
/*7*/   objj=obj;
/*8*/   ((meree)objj).A();
/*9*/   ((fillee)((meree)objj)).stat();
/*10*/  ((fillee)((meree)objj)).C();
}
}

```

Exercice4

On veut créer une classe java qui donne pour chaque matière, l'ensemble des enseignants qui l'enseignent. Selon la figure ci-dessous, on aura une HashMap ou chaque clé est un nom de la matière et la valeur est une LinkedList des noms des enseignants qui assurent cette matière.



Donner le code java :

- qui définit cette classe,
- qui permet de créer cette classe (ajouter les matières et les enseignants),
- qui permet de parcourir cette structure et l'afficher.

التوفيق !

Bonne chance !