Exercice 1:

1. Output:42 42 24 24 24 24

42 🡪 print la valeur de la mere, System.***out***.println(mere.meth());

42 🡪 print la valeur de la classe mere

mere.printMeth();

24 🡪 print la valeur de la fille

System.***out***.println(fille.meth());

24 🡪 print la valeur de la file

fille.printMeth();

24 🡪 print la valeur de la fille

System.***out***.println(mereFille.meth());

24 🡪 mereFille.printMeth();

Parcequ’il est de type mere et il appelle la methode meth de la classe fille

1. On peut acceder a la valeur de meth fille et mere dans la classe fille et on peut acceder a la valeur de meth mere juste dans la classe mere.
2. Output: 42 42 24 42 42 42

42 🡪 print la valeur de la mere, System.***out***.println(mere.meth());

42🡪 print la valeur de la classe mere

mere.printMeth();

24 🡪 print la valeur de la fille

System.***out***.println(fille.meth());

42 🡪 PrintMeth appelle la methode meth(statique) de la classe mere

42 🡪parceque c’est statique donc polymorphisme ne marche pas ici, il utilise la methode de la classe mere

42 🡪 print la valeur de la classe mere parceque la methode meth est statique donc elle a pas changer

1. Output: 42 42 24 42 42 42

Il y a pas de polymorphisme dans les champs

Exercice 2:

1. Erreur de compilation:

FILLE

**private** **void** e() {

System.***out***.println("Fille\_e");

}

**char** h() {System.out.println("Fille\_h"); **return** ’c’;}

**int** i() {

System.***out***.println("Fille\_i");

**return** 3;

}

**void** k() **throws** Exception {

System.***out***.println("Fille\_k");

}

Parceque dans la classe mere les methodes sont defines autrement donc on a des erreurs de compilation dans la classe fille

MAIN:

mere.miage();

Parceque dans la classe mere on a pas de methode miage, mais on a une methode miage dans la classe fille mais la classe mere n’a pas acces a cette methode.

mereFille.miage();

mereFille est une mere qui a comme attribute fille, donc les methodes defines dans la classe fille ne marche pas pour mereFille

1. Redefinition: methode meme nom et meme parametre mais pas meme contenu

Surcharge: methode meme nom mais pas meme parametre

REMAQRUE:

1. si on a utiliser public dans la classe mere, redefinition dans la classe fille🡪 protected ou privet ou sans modificator

si on a utilise protected dans la classe mere, REDEFINITION DANS LA classe fille🡪 private ou protected

si classe mere 🡪 private, redefinition dans la classe fille🡪 private

si classe mere modificator, redefinition dans la classe fille🡪 protected, private, sans modificator

1. Methode de la mere 🡪 Exception

Redefinition dans la classe fille 🡪 excetionIO, ExceptionNullPointer,…

1. Methode de la classe mere return type primitive

Redefinition dans la classe mere 🡪 la meme valeur de retorune

PAS MEME CHOSE POUR TYPE objet