M1 MIAGE

Programmation Java Avancé : TP2

BRIAIRE Celia

Exercice 1

1. Le *Main* affiche à la suite : 42, 42, 24, 24, 24, 24

Car quand on crée *mereFille*, l’objet appelle la fonction la plus précise, donc celle de la sous classe *Fille*, *meth*() renvoie donc 24.

1. S’il est dans *Fille*, il a accès a deux méthodes *meth*(), celle de *Mere* et *Fille*, on peut y accéder grâce au *super*. S’il est dans le *Main* il a également accès aux deux méthodes et dans ce cas-là il appellera la méthode de *Mere* ou *Fille* selon si l’objet est de type *Mere* ou *Fille* (on peut forcer le passage de l’un a l’autre).
2. Le *Main* affiche à la suite : 42, 42, 24, 42, 42, 42
3. On obtient le même affichage qu’avec les *static*, c’est-à-dire : 42, 42, 24, 42, 42, 42

Exercice 2

1. La méthode *miage* n’existe pas pour les objets de type Mère. Problème d’exception avec le méthode *k*() de la classe Fille.
2. /
3. La surcharge survient lorsque deux méthodes ou plus dans une classe ont le même nom de méthode mais des paramètres différents. Une redéfinition signifie avoir deux méthodes avec le même nom et les mêmes paramètres, l’une des méthodes est dans la classe parente et l’autre dans la classe fille. Ici ce sont les méthodes *a*(), *b*(*Fille* fille), *c*(*Mere* mere), *d*(), *e*(), *f*() …

Affiche :

**Miage** //fille.miage();

**Miage**  //((Fille)mereFille).miage(); appelle la méthode *miage* de la classe *Fille*

**Mere\_a**  //mere.a();

**Fille\_a** //mereFille.a(); *mereFille* est un objet de type *Fille* donc appelle la methode *a*() de la classe *Fille*

**Fille\_a** //fille.a();

**Fille\_a** //((Mere)mereFille).a(); appelle la méthode de la classe *Fille*

**Fille\_b(Fille)** //mereFille.b(**null**);

**Mere\_c** //mereFille.c(); appelle la méthode de la classe mère car pas de redefinition dans la classe *Fille*

**Fille\_c(Mere)** //mereFille.c(mere); méthode de la classe *Fille* (fille + mère)

**Fille\_c(Mere)** //mereFille.c(mereFille); appelle la méthode de la classe *Fille* (fille+ mère)

**Fille\_c(Mere)** //mereFille.c(fille); (fille + mère)

**Fille\_c(Fille)** //fille.c(fille); appelle la méthode de la classe *Fille* (fille + fille)

**static Mere\_d** //mere.*d*();

**static Mere\_d** //mereFille.*d*();

**Mere\_f** //mere.printF();

**Mere\_f**  //mereFille.printF(); print ce que la méthode *f*() renvoie

**Fille\_j** //mereFille.j();

**Mere\_k**  //mereFille.k(); renvoie la methode de la classe *Mère* car on avait une erreur sur cette methode dans la classe *Fille*

**Fille\_l** //mereFille.l();

**Fille\_m** //mereFille.m();