TD 2 – Programmation Java Avancee

Exercice 1 :

1 - Le main affiche : 42,42,24,24,24,24.

Car en premier on créer un objet mère et on utilise les méthodes qui retourne deux fois la valeur 42.

Même idée pour la deuxième, on fait le même scénario avec un objet fille et on utilise les méthodes qui retourne deux fois la valeur 24.

Pour le 3e cas, on créer un objet de type Mère, mais on l’initialise avec new Fille. Ce qui est possible puisque fille étends Mère. Une fille est donc une sorte de mère. Malgré tout, elle va afficher 24 car on utilise la fonction qui est définit dans la classe fille.

2 -

Si elle est dans fille, elle pourra utiliser la fonction meth() de la classe Fille et de la classe mère.

Si elle est dans la classe Main, elle pourra utiliser la méthode meth() mais cela dépends de son implémentation.

3 et 4 –

|  |  |
| --- | --- |
| Statiques | Champs |
| 42 | 42 |
| 42 | 42 |
| 24 | 24 |
| 42 | 42 |
| 42 | 42 |
| 42 | 42 |

Pour appeler une méthode statique on n’a pas besoin d’appeler la méthode d’un objet.

Puisqu’il va attendre de compiler, pour lui le type de l’objet est mère d’où l’affichage de « 42 ».

Exercice 2 : Redéfinition et surcharges

1

Lest deux premières de compilation sont : qu’aucune méthode « Miage » n’est disponible pour la classe Mère, car aucune méthode avec cette signature n’est présente dans le code de la classe Mère. Il est donc impossible de les utiliser sur des objets de type « Mère » comme mère et merefille.

La méthode e de la classe fille provoque aussi une erreur, elle devrait être définit en tant que protected pour être utilisée afin de respecter la signature de la méthode mère, mais lors de sa redéfinition dans la classe fille elle ne respecte pas la signature et est définit en private.

La méthode h ne retourne pas le bon type, elle devrait retourner un type int et non char.

La méthode i de la classe fille provoque aussi une erreur, elle devrait ne doit rien retourner pour être utilisée afin de respecter la signature de la méthode mère, mais lors de sa redefinition dans la classe fille elle ne respecte pas la signature et retourne un entier.

La méthode K de la classe fille provoque aussi une erreur, elle devrait jeter une exception de type IOExcepetion pour être utilisée afin de respecter la signature de la méthode mère, mais lors de sa redéfinition dans la classe fille elle ne respecte pas la signature car elle retourne une exception de type Exception.

3

Une surcharge est le fait d’avoir plusieurs par exemple deux fonctions ayant le même nom, mais qui n’ont par exemple pas le même nombre d’arguments.

Une redéfinition est utilisée à travers l’héritage d’une classe, en effet, se précéder permet de réutiliser la signature d’une méthode tout en y modifiant le comportement. Lorsque l’on fait ceci, il est nécessaire d’écrire l’annotation @Override avant notre méthode pour que notre compilateur vérifie bien que c’est une redéfinition.

4

|  |  |  |
| --- | --- | --- |
| Sortie écran | Méthode utilisé | Justification |
| Miage | Fille.miage() | Affiche miage, car comportement de la fonction miage de la classe fille. |
| Miage | (fille)mereFille.miage() | Affiche miage, utilisation d’un cast pour signifier que l’objet est de type fille, utilisation de la méthode présente dans la classe fille. |
| Mere\_a | Mere.a() | Affichage simple de la méthode définit dans la classe Mere. |
| Fille\_a | mereFille.a() | Affichage simple de la méthode définit dans la classe fille. |
| Fille\_a | fille.a() | Affichage simple de la méthode définit dans la classe fille. |
| Fille\_a | ((Mere)mereFille).a() | - |
| Fille\_b(Fille) | mereFille.b(null) | - |
| Mere\_c | mereFille.c() |  |
| Fille\_c(Mere) | mereFille.c(mere) |  |
| Fille\_c(Mere) | mereFille.c(mereFille) |  |
| Fille\_c(Mere) | mereFille.c(fille) |  |
| Fille\_c(Fille) | fille.c(fille) ; |  |
| Static mere\_d | mere.d() |  |
| Static mere\_d | mereFille.d() |  |
| Mere\_f | Mere.printF() |  |
| Mere\_f | mereFille.printF() |  |
| Fille\_j | mereFille.j() |  |
| Mere\_k | mereFille.k() |  |
| Fille\_l | mereFille.l() |  |
| Fille\_m | mereFille.m() |  |

Exercice 3 : Expressions arithmétique.