TD 1

## Classes et visibilité

1. Parmi les appels suivants, indiquez ceux qui sont corrects et ceux qui ne le sont pas :

public class A {  
 public int i;  
 private int a;  
 protected int b;  
   
 public void afficher() {  
 System.out.println(i + a + b);  
 }  
}  
   
public class B extends A {  
 public void afficher() {  
 System.out.println(i + a + b);  
 }  
}  
   
public class D {  
 public A a = new A();  
   
 public void afficher() {  
 System.out.println(a.i + a.a + a.b);  
 }  
}  
   
public class E {  
 public B b = new B();  
   
 public void afficher() {  
 b.afficher();  
 }  
}  
   
public class F {  
 public B b = new B();  
   
 public void afficher() {  
 System.out.println(b.i + b.a + b.b);  
 }  
}

Réponse :

1. Ce code compile t-il et, si non, quelle(s) erreur(s) le compilateur va t-il indiquer ?

class Toto {  
   
 int toto = 0;  
   
 Toto() {  
 toto = toto + 1;  
 }  
   
 public static void main(String[] tutu) {  
 Toto t1 = new Toto();  
 Toto t2 = new Toto();  
 System.out.println("Toto : " + toto);  
 }  
}

Réponse :

1. Le code suivant compile t-il? Si non, indiquez les erreurs affichées par le compilateur et proposez des corrections. À quel affichage conduit l’exécution du programme (éventuellement corrigé)?

class Test {  
   
 int i;  
   
 Test(int i) {  
 this.i = 12;  
 i = 15;  
 }  
   
 void i() {  
 Test i = new Test(3);  
 System.out.println(i.i);  
 i.i();  
 }  
   
 public static void main(String[] toto) {  
 Test i = new Test(34);  
 i.i();  
 }  
}

## Relations entre classes

1. Définir les relations entre les classes suivantes :
   * Personne, homme, femme, enfant
   * Animal, chien, chat, ravet, dalmatien, domestique, sauvage, racoon, ours, comestible > Réponse :
2. Généraliser :
   * Bateau, voiture, avion > Réponse :
   * Souris, clavier, écran > Réponse :
3. Spécialiser :
   * Ordinateur, … > Réponse :
4. D’après le code suivant, donnez le diagramme de classe (classes et relations).

public class A {  
 public B b;  
}  
   
public class B {  
 public C c;  
 public B(C c) {  
 this.c = new C(c);  
 }  
}  
   
public class C {  
 public int i, j;  
 public C(C c) {  
 this.i = C.i;  
 this.j = C.j;  
 }  
}  
   
public class D {  
 void test() {  
 B b = new B();  
 A a = new A(b);  
 A a2 = new A(b);  
 }  
}

Réponse :

## Analyse de code

public class Parent {  
 int x;  
 Parent(int k) {  
 x = k;  
 }  
 int ajoute(int a) {  
 return x + a;  
 }  
 public void moi() {  
 System.out.println(" x = " + x);  
 }  
}  
public class Enfant1 extends Parent {  
 int y;  
 Enfant1(int k, int l) {  
 super(k);  
 y = l;  
 }  
 int ajoute(int a) {  
 return x + 2 \* a;  
 }  
}  
public class Enfant2 extends Enfant1 {  
 int z;  
 Enfant2(int k, int l, int m) {  
 super(k, l);  
 z = m;  
 }  
 int ajoute(int a) {  
 return x + 3 \* a;  
 }  
 public void moi() {  
 super.moi();  
 System.out.println(" z = " + z);  
 }  
}  
public class Essai {  
 public static void main(String args[]) {  
 int a = 2;  
 Parent p = new Parent(3);  
 p.moi();  
 System.out.println(" ajoute(" + a + ") = " + p.ajoute(a));  
 Enfant1 e1 = new Enfant1(3, 4);  
 e1.moi();  
 System.out.println(" ajoute(" + a + ") = " + e1.ajoute(a));  
 e1 = new Enfant2(3, 4, 5);  
 e1.moi();  
 System.out.println(" ajoute(" + a + ") = " + e1.ajoute(a));  
 }  
}

1. Quels sont les attributs dont disposent les classes Enfant1 et Enfant2 ? > Réponse :
2. Écrivez le résultat de l’exécution de la classe Essai. > Réponse :

## Cours

1. Quelle est la différence entre une classe et un objet ? > Réponse :
2. En quoi l’héritage est-il un outil puissant pour le développement d’applications ? > Réponse :
3. Quelle est la différence entre :

Point p[][] = new Point[5][4];  
Point p = new Point(5,4);

Réponse : 4. Expliquez la notion d’Exception en java (fonctionnement, déclaration, …) Réponse :