Canard) Concordance des types : statique et dynamique

**Question 1)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Z | T | X=y.f(z,t) |
| Animal | Animal | Animal | Animal | Erreur |
| Animal | Animal | Animal | Canard | Erreur |
| Animal | Animal | Canard | Animal | Erreur |
| Animal | Animal | Canard | Canard | Erreur |
| Animal | Canard | Animal | Animal | Erreur |
| Animal | Canard | Animal | Canard | Erreur |
| Animal | Canard | Canard | Animal | Erreur |
| Animal | Canard | Canard | Canard | Erreur |
| Canard | Animal | Animal | Animal | Ok |
| Canard | Animal | Animal | Canard | Ok |
| Canard | Animal | Canard | Animal | Ok |
| Canard | Animal | Canard | Canard | Ok |
| Canard | Canard | Animal | Animal | Ok |
| Canard | Canard | Animal | Canard | Ok |
| Canard | Canard | Canard | Animal | Ok |
| Canard | Canard | Canard | Canard | Ok |

**Question 2)**

F renvoie toujours un canard

2) Utilisation de instanceOf

**Question 3)**

Object c = new Canard();

Animal a = new Animal();

("c est une instance de Canard : " + (c instanceof Canard)); TRUE

("c est une instance de Animal : " + (c instanceof Animal)); TRUE

("c est une instance de Object : " + (c instanceof Object)); TRUE

("c est une instance de Integer : " + (c instanceof Integer)); FALSE

("a est une instance de Canard : " + (a instanceof Canard)); FALSE

("a est une instance de Animal : " + (a instanceof Animal)); TRUE

("a est une instance de Object : " + (a instanceof Object)); TRUE

("a est une instance de Integer : " + (a instanceof Integer)); FALSE

**Question 4)**

public static Boolean instanceOf(A a , B b){

try(a = (B) b){

return(true)

}

catch(ExceptionType e){

return(false)

}

}

3) Constructeur et héritage

**Question 5)**

Main : constructeur de B

constructeur de C

**Question 6)**

Main : autre constructeur de B

constructeur de C

**Question 7)**