

TD Exceptions

Exercice 1

Quels résultats fournit le programme suivant ?

Que se passe-t-il si on inverse l'ordre des deux gestionnaires dans le second bloc *try* ?

```
class Erreur extends Exception {
    protected int num;
    public Erreur(int n){
        num = n;
    }

    public String toString(){
        return "Exception Erreur " + num;
    }
}

class Erreur_d extends Erreur {
    private int code;
    public Erreur_d(int n, int c){
        super(n);
        code = c;
    }

    public String toString(){
        return "Exception Erreur_d " + num + " " + code;
    }
}

class A {
    public A(int n) throws Erreur_d{
        if(n==1){
            throw new Erreur_d(999,12);
        }
    }
}

public class Chemin1 {

    public static void main(String[] args) {
        try{
            A a = new A(1);
            System.out.println("Après la création de a(1)");
        }catch (Erreur e){
            System.out.println(e);
        }
        System.out.println("Suite du main");

        try{
            A b = new A(1);
            System.out.println("Après la création de b(1)");
        }catch (Erreur_d e){
            System.out.println(e);
        }catch (Erreur e){
            System.out.println(e);
        }
    }
}
```

Exercice 2

Que fournit le programme suivant ?

```
class Erreur extends Exception {}
class Erreur1 extends Erreur {}
class Erreur2 extends Erreur {}
class A {
    public A(int n) throws Erreur {
        try {
            if(n==1) throw new Erreur1();
            if(n==2) throw new Erreur2();
            if(n==3) throw new Erreur();
        }
        catch(Erreur1 e) {
            System.out.println("*** Exception Erreur1 dans constructeur A");
        }
        catch(Erreur e) {
            System.out.println("*** Exception Erreur dans constructeur A");
            throw(e);
        }
    }
}

public class Redec1 {
    public static void main(String args[]) {
        int n;
        for(n=1;n<=3;n++) {
            try {
                A a = new A(n);
            }
            catch(Erreur1 e) {
                System.out.println("*** Exception Erreur1 dans main");
            }
            catch(Erreur2 e) {
                System.out.println("*** Exception Erreur2 dans main");
            }
            catch(Erreur e) {
                System.out.println("*** Exception Erreur dans main");
            }
            System.out.println("-----");
        }
        System.out.println("Fin main");
    }
}
```

Exercice 3

Quels résultats fournit le programme suivant ?

```
class Except extends Exception {}

public class FinReth {

    public static void f(int n) throws Except {
        try {
            if(n!=1) throw new Except();
        }
        catch(Except e) {
            System.out.println("catch dans f avec n = " + n);
            throw e;
        }
        finally {
            System.out.println("dans finally de f avec n = " + n);
        }
    }

    public static void main(String args[]) {
        int n = 0;
        try {
            for(n=1;n<5;n++)
                f(n);
        }
        catch(Except e) {
            System.out.println("catch dans main avec n = " + n);
        }
        finally {
            System.out.println("dans finally de main avec n = " + n);
        }
    }
}
```