	Référence	Constructeur	Instanciation(new)
Class	Oui	Oui	Oui
Class abstract	Oui	Oui	Non
Interface	Oui	Non	Non
enum	Oui	Oui	Non

Différence Ant &

Eclipse : Ant est un système de build (comme make), alors qu'un IDE est un gestionnaire de projet. L'IDE va permettre d'éditer le code, de browser les sources, de debugger son programme, et de le compiler ; il va donc *appeller* Ant à unmoment donné.

@Override : L'annotation @Override, placée avant la définition d'une méthode de classe, sert à signaler qu'on redéfinit (surcharge) cette méthode par rapport à celle fournie par la classe mère. C'est ce qu'on fait pour toString quand on veut spécifier un pretty-printing pour une classe.

BufferedReader

```
File f = new File("mydata.txt");
FileInputStream fis =
        new FileInputStream(f);
InputStreamReader isr =
        new InputStreamReader(fis);
BufferedReader br =
        new BufferedReader(isr);
```

:

Iterator :

```
import java.util.*;
public class Test implements Iterable{
    Iterator<TrucAIterer> iter=j.iterator();
        while(iter.hasNext()){
        iter.next();
        }
        }
```

Scanner :

```
import java.util.*;

Scanner sc=new Scanner();
String chaine=sc.nextLine();
Int numero=sc.nextInt();
```

Sérialisation :

```
// chargement d'un cbjet d'un
fichier
FileInputStream fin = new

FileInputStream("c:/biblio.obj");
ObjectInputSteam oin = new
    ObjectInputStream(fin));
Biblioteque biblio =
oin.readObject();
// on peut naviguer vers des
livres et des adhérents et
exemplaires
```

```
public class Bibliotheque
        implements Serializable
{
   List<Livre> livres;
   List<Adhérent> adherents;
}
```

```
public class Adherent
        implements Serializable {
  String nom;
}
```

```
public class Livre
        implements Serializable
{
   String nom;  String auteur;
   List<Exemplaire> exemplaires;
}
```

```
public class Exemplaire
        implements Serializable {
  int no;  Adherent emprunteur;
    // null signifie libre
}
```

Exceptions :

```
// DivParZero.java
class DivParZero {
 public static void main (String argv[] ) {
    int val=0;
    try {
        val = 1997/val;
    }
    catch (ArithmeticException e ) {
        System.out.println("Une exception arithmétique a été levée");
        System.out.println("Message : " + e.getMessage());
        System.out.println("Pile :");
        e.printStackTrace();
    }
    finally {
        System.out.println("Fin du calcul");
    }
    System.out.println("Fin du programme");
  //fin de main
}//fin de la classe
```

A l'exécution, on obtient :

```
Une exception arithmétique a été levée
Message : / by zero
Pile :
java.lang.ArithmeticException: / by zero
at DivParZero.main(DivParZero.java:5)
Fin du calcul
Fin du programme
```

Résumé des droits

	Clas	Paqt	Sous-cls	Monde
• public	O	O	O	O
• protected	O	O	O	N
• *implicite*	O	O	N	N
• private	O	N	N	N

```
package calculmathematique;

import java.util.Scanner;
import java.io.*;

public class linearAppli implements java.io.Serializable{
   private String nomapli;
   private Matrix matrice;
   private int dimension;
   private Matrix vecteur;
   private Matrix result;
   private boolean A=true;
   public  File repR= new File(System.getProperty("user.dir"));
//----------------------CONSTRUCTEUR-----------------------------
   public linearAppli(){
      String nomapli="";
      int dimension;
      Matrix matrice= new Matrix();
      boolean A=true;
      File repR= new File(System.getProperty("user.dir"));
   }
//-----------------------GETTER & SETTER-----------------------------
   public File get_File(){
     return repR;
   }
   public boolean get_A(){
     return A;
   }
   public int get_dimension(){
     return this.dimension;
   }
   public String get_nom(){
     return this.nomapli;
   }
//-----------------------------INITIALIZE-----------------------------------
   public linearAppli Initialize(){
     Scanner saisir= new Scanner(System.in);
     String lol;
     System.out.println("Saisir le nom de l'appli : \n");
     this.nomapli=saisir.next();
     System.out.println("Saisir la dimension de la matrice associé");
     this.dimension=saisir.nextInt();
     Matrix matrice = new Matrix();
     this.matrice=matrice.Initialize(matrice,nomapli+"_Matrix",dimension,dimension);
     return this;
   }
//-----------------------------AFFICHAGE-----------------------------------
   public void Show(linearAppli ali){
     System.out.println("Nom de l'application : "+ali.nomapli);
     ali.matrice.show();
   }
   public Matrix vectImage(Matrix Vect){
     Matrix M= new Matrix(1,this.get_dimension());
     M = Vect.multi(this.matrice);
     return M;
   }
   }
```

```
//----------------------Lister les fichiers-----------------------------
     public void listerRepertoire(File repertoire){
     String [] listeEspace;
     int i;
     System.out.println("Applications linéaire Disponible :");
     listeEspace=repertoire.list();
     for(i=0;i<listeEspace.length;i++){
        if(listeEspace[i].endsWith("_LinearAppli.ser"))
          System.out.println(listeEspace[i].replaceFirst("_LinearAppli.ser",""));
     }
   }
//-------------------------------------------------------------
//----------------------SERIALISATION-----------------------------------
  public void serialize(linearAppli apl ){
     try
     {
       FileOutputStream fichier = new FileOutputStream(apl.nomapli+"_LinearAppli.ser");
       ObjectOutputStream oos = new ObjectOutputStream(fichier);
       oos.writeObject(apl);
       oos.flush();
       oos.close();
     }
     catch (java.io.IOException e) {
     e.printStackTrace();
     }
    public linearAppli deserialize(String nom, boolean S){
     linearAppli ap = new linearAppli();
     try {
       FileInputStream fichier = new FileInputStream(nom+"_LinearAppli.ser");
       ObjectInputStream ois = new ObjectInputStream(fichier);
       ap = (linearAppli) ois.readObject();
       }
       catch (java.io.IOException e) {
         System.out.println("    -----Erreur!!!-----\n nom de l'application non trouvé\n\n");
         listerRepertoire(ap.get_File());
         ap.A=false;
       }
       catch (ClassNotFoundException e) {
         System.out.println("    -----Erreur!!!-----\n nom de l'application non trouvé\n\n");
         listerRepertoire(ap.get_File());
         ap.A=false;
       }
     return ap;
   }
}
//-------------------------------------------------------------
```