

LES BIENFAITS DE L'HERITAGE

I Sans Héritage

```
public class Utilisation {  
  
    public static void main(String args[]){  
        EtudiantENPC eENPC = new EtudiantENPC("alfred", "paul", 05);  
        EtudiantESIEE eESIEE = new EtudiantESIEE("paul", "alfred", 2001);  
        EtudiantIUT eIUT = new EtudiantIUT("jean", "jacques","Paris");  
  
        CiteDescartes citeDescartes = new CiteDescartes();  
  
        citeDescartes.ajouterUnEtudiantENPC(eENPC);  
        citeDescartes.ajouterUnEtudiantESIEE(eESIEE);  
        citeDescartes.ajouterUnEtudiantIUT(eIUT);  
  
        System.out.println(citeDescartes.toString());  
    }  
}
```

```
public class EtudiantENPC {  
    private String nom;  
    private String prenom;  
    private int promotion;  
  
    public EtudiantENPC(String nom, String prenom, int promotion){  
        this.nom = nom;  
        this.prenom = prenom;  
        this.promotion = promotion;  
    }  
  
    public String loginENPC(){  
        // algorithme propre à l'école  
        return "adminENPC";  
    }  
  
    public String getNom(){ return nom; }  
  
    public String getPrenom(){ return prenom; }  
  
    @Override  
    public String toString(){  
        return getNom() + "_" + getPrenom() + ":" + promotion;  
    }  
}
```

```

public class EtudiantESIEE {
    private String nom;
    private String prenom;
    private int annee;

    public EtudiantESIEE(String nom, String prenom, int annee){
        this.nom = nom;
        this.prenom = prenom;
        this.annee= annee;
    }

    public String loginESIEE(){
        // algorithme propre à l'école
        return "adminESIEE";
    }

    public String getNom(){ return nom; }

    public String getPrenom(){ return prenom; }

    @Override
    public String toString(){
        return this.getNom() + "_" + this.getPrenom() + ":" + annee;
    }
}

```

```

public class EtudiantIUT {

    private String nom;
    private String prenom;
    private String adresse;

    public EtudiantIUT(String nom, String prenom, String adresse) {
        super();
        this.nom = nom;
        this.prenom = prenom;
        this.adresse = adresse;
    }

    public String getNom() { return nom; }

    public void setNom(String nom) { this.nom = nom; }

    public String getPrenom() { return prenom; }

    public void setPrenom(String prenom) { this.prenom = prenom; }

    public String getAdresse() { return adresse; }

    public void setAdresse(String adresse) { this.adresse = adresse; }
}

```

```

@Override
public String toString() {
    return "EtudiantIUT [adresse=" + adresse + ", nom=" + nom + ", prenom="
        + prenom + ", toString()=" + super.toString() + "];"
}

```

```

}

```

```

import java.util.ArrayList;

```

```

public class CiteDescartes {

```

```

    private ArrayList<Object> listeDesEtudiants;

```

```

    public CiteDescartes(){

```

```

        listeDesEtudiants = new ArrayList<Object>();

```

```

    }

```

```

    public void ajouterUnEtudiantENPC(EtudiantENPC e){

```

```

        listeDesEtudiants.add(e);

```

```

    }

```

```

    public void ajouterUnEtudiantESIEE(EtudiantESIEE e){

```

```

        listeDesEtudiants.add(e);

```

```

    }

```

```

    public void ajouterUnEtudiantIUT(EtudiantIUT e){

```

```

        listeDesEtudiants.add(e);

```

```

    }

```

```

@Override

```

```

public String toString() {

```

```

    return listeDesEtudiants.toString();

```

```

}

```

```

}

```

II Avec Héritage

```
public class Utilisation {
```

```
    public static void main(String args[]){  
        EtudiantENPC    eENPC = new EtudiantENPC("alfred", "paul", 05);  
        EtudiantESIEE  eESIEE = new EtudiantESIEE("paul", "alfred", 2001);  
        EtudiantIUT    eIUT  = new EtudiantIUT("jean", "jacques", "Paris");
```

```
        CiteDescartes citeDescartes = new CiteDescartes();
```

```
        citeDescartes.ajouterUnEtudiant(eENPC);  
        citeDescartes.ajouterUnEtudiant(eESIEE);  
        citeDescartes.ajouterUnEtudiant(eIUT);;
```

```
        System.out.println(citeDescartes.toString());
```

```
        // test des affectations
```

```
        //DECOMMENTER LE CODE SUIVANT ET OBERVER.....
```

```
        Etudiant e = new Etudiant("Kennedy", "John");  
        System.out.println("e référence " + e);
```

```
        // Etudiant <--- EtudiantENPC un étudiant ENPC est un étudiant.
```

```
        e = eENPC;  
        System.out.println("e référence " + e);
```

```
        // Etudiant <--- EtudiantESIEE un étudiant ESIEE est un étudiant.
```

```
        e = eESIEE;  
        System.out.println("e référence " + e);
```

```
        // un étudiant ESIEE n'est pas un étudiant ENPC.
```

```
        // eENPC = eESIEE;
```

```
        // EtudiantENPC <--- Etudiant FAUX tout étudiant n'est pas un étudiant ENPC.
```

```
        // eENPC = e; // tout étudiant n'est pas un étudiant ENPC
```

```
    /*
```

```
        System.out.println("test des affectations");  
        System.out.println("e référence " + e);  
        System.out.println(eENPC);  
        System.out.println(eESIEE);
```

```
    */
```

// UN TABLEAU COMME CONTAINER

```
Etudiant tab [] = new Etudiant [4];
tab[0] = new Etudiant("Toto", "Toto");
tab[1] = new EtudiantENPC("Toto", "Toto", 12);
tab[2] = new EtudiantESIEE("Toto", "Toto", 2010);
tab[3] = new EtudiantIUT("Toto", "Toto", "Paris");

System.out.println("\n\n Affichage du container\n\n");
for (int i = 0; i < tab.length; i++)
{
    System.out.println(tab[i].toString());
}
}
}
```

```
public class Etudiant {
    private String nom;
    private String prenom;

    public Etudiant(String nom, String prenom){
        this.nom = nom;
        this.prenom = prenom;
    }

    public String getNom(){ return nom; }

    public String getPrenom(){ return prenom; }

    @Override
    public String toString(){
        return getNom() + "_" + getPrenom();
    }
}
```

```

public class EtudiantENPC extends Etudiant {

    private int promotion;

    public EtudiantENPC(String nom, String prenom, int promotion){
        super(nom, prenom);
        this.promotion = promotion;
    }

    public String loginENPC(){
        // algorithme propre à l'école
        return "adminENPC";
    }

    @Override
    public String toString(){
        return super.toString() + ":" + promotion;
    }
}

```

```

public class EtudiantESIEE extends Etudiant{
    private int annee;

    public EtudiantESIEE(String nom, String prenom, int annee){
        super(nom, prenom);
        this.annee= annee;
    }

    public String loginESIEE(){
        // algorithme propre à l'école
        return "adminESIEE";
    }

    @Override
    public String toString(){
        return super.toString() + ":" + annee;
    }
}

```

```

public class EtudiantIUT extends Etudiant {

    private String adresse;

    public EtudiantIUT(String nom, String prenom, String adresse) {
        super(nom, prenom);
        this.adresse = adresse;
    }

    public String getAdresse() {
        return adresse;
    }

    public void setAdresse(String adresse) {
        this.adresse = adresse;
    }

    @Override
    public String toString() {
        return "EtudiantIUT [" + ssuper.toString() + "adresse=" + getAdresse() + "];"
    }

}

```

```
import java.util.ArrayList;
```

```

public class CiteDescartes {

    private ArrayList<Etudiant> listeDesEtudiants;

    public CiteDescartes(){
        listeDesEtudiants = new ArrayList<Etudiant>();
    }

    public void ajouterUnEtudiant(Etudiant e){
        listeDesEtudiants.add(e);
    }

    @Override
    public String toString() {
        return listeDesEtudiants.toString();
    }

}

```