Corrigé type de l'examen final de la POO

Exercice 1 : Questions de compréhension du cours

R1 : Parmi les différences entre les classes abstraites et les interfaces, on peut citer : ------**2 pts**

- A l'opposé de la classe abstraite, l'interface n'est pas une classe.
- Dans une classe abstraite, on trouve au moins une fonction abstraite, tandis que toutes les fonctions d'une interface sont abstraites.

R2 : Le mot clé d'autoréférence *this* ne peut pas être utilisé dans une fonction statique parce qu'elle est indépendante de tout objet et elle peut être appelée sur une classe (sans créer d'instances de classes). ------**2 pts**

Exercice 2 :

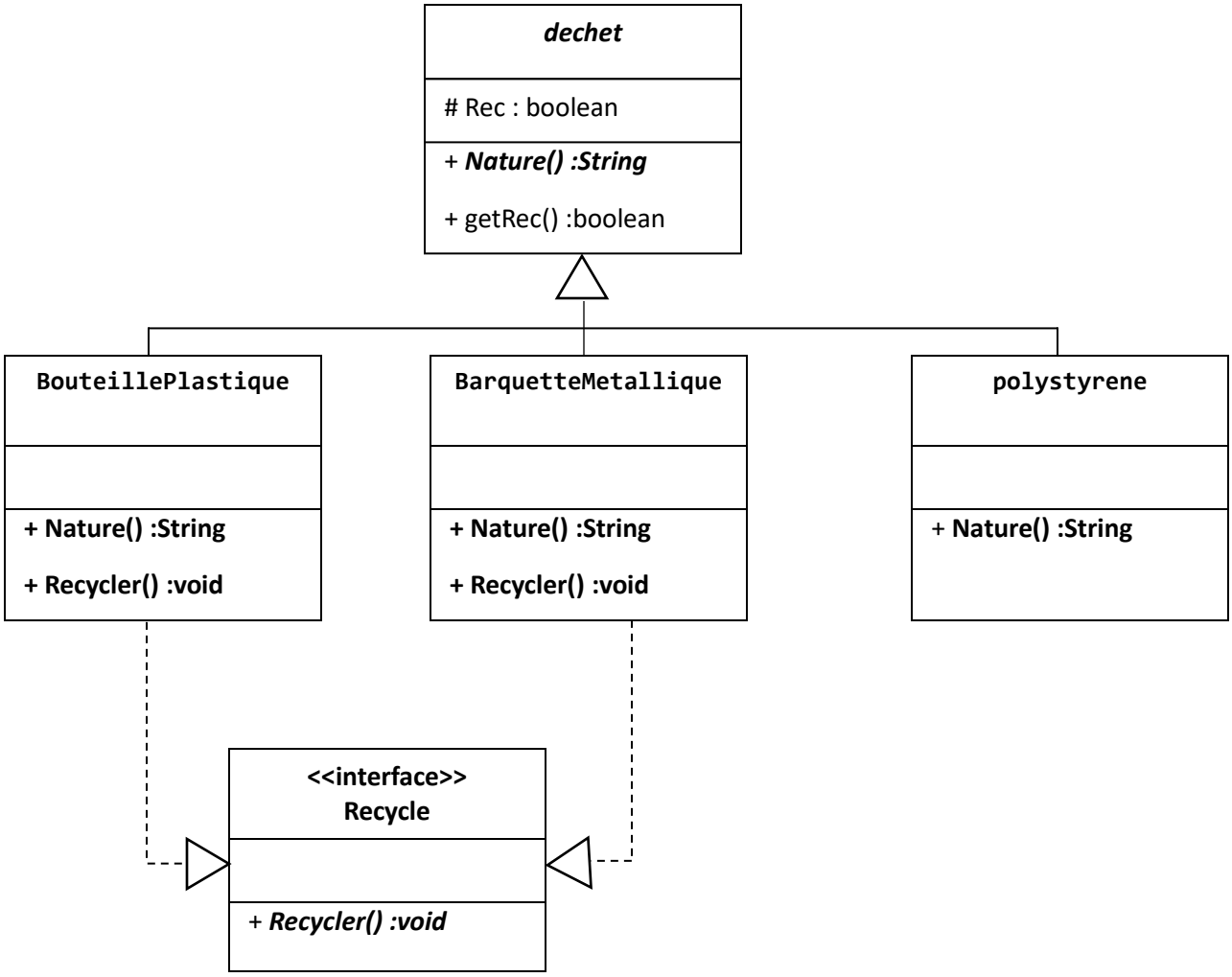
Ci-après le code java corrigé

```
public class Personne {                                          ---- 1 pts
      private String Nom;
      private String Prenom;
      public Personne (String Nom, String Prenom) {
            this.Nom=Nom;
            this.Prenom=Prenom;
      }

}
```

```
public class Etudiant extends Personne{                          ----3 pts
private int Ninsc;
private String Spec;
public Etudiant(String Nom, String Prenom, int Ninsc, String Spec){
            super(Nom, Prenom);
            this.Ninsc=Ninsc;
            this.Spec=Spec;

      }
}
```

Exercice 3 :

1. La figure suivante montre le diagramme de classe décrivant l'aspect statique de l'application ----- **2 pts**



2. Ci-après le programme java de l'application

```
public abstract class dechet {                                    ---1 pts
      protected boolean Rec;
      public dechet() {
            System.out.println("Je suis un dechet");
            }
      public abstract String Nature();
      public boolean getRec() {
            return Rec;
      }
}
```

```java
public interface Recycle {                                    ---1 pts
      public abstract void Recycler();

}
public class BouteillePlastique extends dechet implements Recycle{
                                                              --- 1pts
   public BouteillePlastique() {
           System.out.println("Je suis une bouteille plastique");
           Rec=true;
           }
   public String Nature() {
           return "plastique";
      }
   public void Recycler() {
   System.out.println("Me Recycler, je suis une BouteillePlastique");
      }
}
public class BarquetteMetallique extends dechet implements Recycle {
   public BarquetteMetallique(){                              --- 1pts
           System.out.println("Je suis une BarquetteMetallique");
           Rec=true;
      }
   public String Nature() {
           return "metal";
      }
   public void Recycler() {
   System.out.println("Me Recycler, je suis une BarquetteMetallique");

      }
}


public class polystyrene extends dechet {                  --- 1pts
      public polystyrene() {
           System.out.println("Je suis un polystyrene");
           Rec=false;
      }
      public String Nature() {
           return "polystyrene";
      }
}
```

```
public class principale {
      static dechet Dech []=new dechet[6];
      static dechet Recy []=new dechet[4];
      static dechet NRecy []=new dechet[2];

    public static void CollecterDechet(BouteillePlastique bp1,
BouteillePlastique bp2, BarquetteMetallique bm1, BarquetteMetallique
bm2, polystyrene p1, polystyrene p2 ) {                --- 2pts
      Dech[0]=bp1;
      Dech[1]=p1;
      Dech[2]=bm1;
      Dech[3]=bp2;
      Dech[4]=p2;
      Dech[5]=bm2;

    }
    public static void TriDechet() {                    --- 2pts
      int j=0; int k=0;
      for (int i=0; i<= 5; i++) {
            if (Dech[i].getRec()) {
                                    Recy[j]=Dech[i];
                                    j++;
                                 }
            else {
                        NRecy[k]=Dech[i];
                        k++;
            }
      }
    }


    public static void Afficher(dechet [] D ) {
      for (int i=0; i<= D.length-1; i++) {
            System.out.println("Je suis un dechet de nature  " +
D[i].Nature());

      }

    }
```

```java
public static void main(String[] args) {      --- 1pts
        // TODO Auto-generated method stub

        BouteillePlastique bp1=new BouteillePlastique();
        BouteillePlastique bp2=new BouteillePlastique();
        BarquetteMetallique bm1= new BarquetteMetallique();
        BarquetteMetallique bm2= new BarquetteMetallique();
        polystyrene p1= new polystyrene();
        polystyrene p2= new polystyrene();

        CollecterDechet(bp1, bp2, bm1, bm2, p1, p2 );

        TriDechet();

        Afficher(Dech);
        Afficher(Recy);
        Afficher(NRecy);
    }
}
```