Correction de l'examen de rattrapage Java

Exercice 1 : Générateur de code HTML (04 pts)

1. Définition de la méthode `genererTitre` **(0.5 pts)**.
2. Définition de la méthode `genererEntete` **(0.5 pts)**.
3. Définition de la méthode `genererContenu` **(0.75 pts)**.
4. Définition de la méthode `genererCorps` **(0.5 pts)**.
5. Le programme dans la méthode principale **(1.75 pts)**.

```
public class Exercice1{

   public static String genererTitre(String titre){
      return ("<title> "+titre+" </title>");
   }

   public static String genererEntete(String titre){
      return ("<head> "+titre+" </head>");
   }

   public static String genererContenu(String contenu, int type){
      return ("<h"+type+"> "+contenu+"</h"+type+"> ");
   }

   public static String genererCorps(String contenu){
      return ("<body> "+contenu+" </body>");
   }

   public static void main(String[] args){
      String TitreHtml = genererTitre("Master 2 ESE");
      String EnteteHtml = genererEntete(TitreHtml);
      String ContenuHtml = genererContenu("Ceci est l examen de rattrapage", 1);
      String CorpsHtml = genererCorps(ContenuHtml);
      System.out.println("<html> "+EnteteHtml+ " "+CorpsHtml+" </html>");
   }
}
```

Exercice 2 : Répertoire téléphonique (10 pts)

1. Classe Personne :
   (a) Définition des attributs **(0.75 pts)**.
   (b) Définition du constructeur **(0.25 pts)**.
   (c) Définition des setters et getters **(1.5 pts)**.
2. Classe Numero :
   (a) Définition des attributs **(0.5 pts)**.
   (b) Définition du constructeur **(0.25 pts)**.
   (c) Définition des setters et getters **(0.5 pts)**.
   (d) Condition sur la taille et le premier chiffre 0 **(0.5 pts)**.
3. Classe EMail :
   (a) Définition des attributs **(0.25 pts)**.
   (b) Définition du constructeur **(0.25 pts)**.

(c) Définition des setters et getters **(0.5 pts)**.

(d) Condition sur les caractères spéciaux, les chiffres et le caractère '@' **(0.75 pts)**.

4. Classe Contact :
   (a) Définition des attributs **(1 pts)**.
   (b) Définition de l'héritage **(0.5 pts)**.
   (c) Définition du constructeur **(0.5 pts)**.
   (d) Définitions des setters et getters **(1 pts)**.
5. Classe Repertoire :
   (a) Définition des attributs **(0.5 pts)**.
   (b) Exemple dans la méthode principale **(0.5 pts)**.

```
public class Personne{

      private String nom;
      private String prenom;
      private String pseudo;

      public Personne(String nom, String prenom, String pseudo){
         this.nom = "nom";
         this.prenom = "prenom";
         this.pseudo = "pseudo";
      }

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

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

      public void setPseudo(String pseudo){
         this.pseudo = pseudo;
      }

      public String getNom(){
         return this.nom;
      }

      public String getPrenom(){
         return this.prenom;
      }

      public String getPseudo(){
         return this.pseudo;
      }

   }
public class Numero{
      private int [] num;
      public Numero (){
         this.num = new int[10];
      }
      public int[] getNum(){
         return this.num;
      }

      public void setNum(int[] num){
         if (num.length == 10)
         if (num[0] != 0){
            this.num = num;
```

```
        }
        else {
           System.out.println("Error");
        }
     }
  }
public class EMail{
     private String email;

     public EMail(){
        this.email = "";
     }

     public String getEmail(){
        return this.email;
     }

     public void setEmail(String email){
        if (!email.contains("+") && !email.contains("-") && !email.contains("*") &&
            !email.contains("/")){
           if (!email.startsWith("0") && !email.startsWith("1") && !email.startsWith("2") &&
               !email.startsWith("3") && !email.startsWith("4")
              && !email.startsWith("5") && !email.startsWith("6") && !email.startsWith("7") &&
                  !email.startsWith("8") && !email.startsWith("9")){
              if(email.contains("@")) this.email = email;
              else System.out.println("une adresse mail doit contenir le caractere @");

           }
           else System.out.println("une adresse mail ne doit pas commencer par un chiffre");
        }
        else System.out.println("une adresse mail ne doit pas contenir un caractere special");

     }
  }
public class Contact extends Personne{

     private ArrayList<Numero> numero;
     private ArrayList<EMail> email;

     public Contact(String nom, String prenom, String pseudo, Numero num, EMail mail){
        super(nom, prenom, pseudo);
        this.numero = new ArrayList<Numero>();
        this.email = new ArrayList<EMail>();
        this.numero.add(num);
        this.email.add(mail);
     }

     public void setNumero(ArrayList<Numero> numero)
     {
     this.numero = numero;
     }
     public ArrayList getNumero(){
        return this.numero;
     }

     public void setEmail(ArrayList<EMail> email)
     {
     this.email = email;
     }
     public ArrayList getEmail(){
        return this.email;
     }
  }
public class Repertoire{
```

```
    public static void main(String[] args){
       ArrayList<Contact> contacts = new ArrayList<Contact>();
       Numero num = new Numero();
       EMail email = new EMail();


       int[] n = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

       num.setNum(n);
       email.setEmail("java@usto.dz");
       Contact contact = new Contact("nom", "prenom", "pseudo", num, email);


    }
  }
```

Exercice 3 : File d'attente (06 pts)

1. Définition de la classe avec implémentation de l'interface **(0.5 pts)**.
2. Définition de l'attribut **(0.25 pts)**.
3. Définition du constructeur **(0.25 pts)**.
4. Redéfinition des méthodes **(04 pts)**.
5. Définition de la méthode d'affichage **(01 pts)**.

```
public class Exercice3 implements File{

  private ArrayList<Object> file;

  public Exercice3 (){
     this.file = new ArrayList<Object>();
  }


  public int size(){
     return this.file.size();
  }


  public boolean isEmpty(){
     return this.file.isEmpty();
  }

  public void enfiler(Object o){
     this.file.add(o);
  }


  public Object defiler(){
     return this.file.remove(0);
  }

  public void afficher(){

     System.out.println(this.file);
  }

}
```