**CORRECTION TEST DE RECRUTEMENT ANALYSTE PROGRAMMEUR JAVA/CSHARP**

**1ère PARTIE: COURS (sans document)**

1. **QCM (35 points)**

1 – a

2 – a

3 – (a et c)

4 – a

5 – c

6 – a

7 – a

8 – b

9 – (b et c)

10 – b

11 – (a et b)

12 – b

13 – a

14 – (a et b)

15 – b

16 – (a et c)

17 – b

18 – b

19 – c

20 – c

21 – a

22 – c

23 – b

24 – b

25 – a

26 – c

27 – b

28 – c

29 – c

30 – (a et c)

31 – b

32 – b

33 – a

34 – (a et b)

35 - b

**II**- **Questions libres (15 points)**

Q1-

Les rôles d'un constructeur sont :

- allouer en mémoire l'objet c'est-à-dire réservez une place mémoire pour chacun des attributs qu'ils soient privés ou publiques

- initialiser les attributs avec des valeurs calculées ou passées en paramètre

- réaliser des traitements qui doivent être fait à chaque fois qu'un objet est créé

**public Individu(String insee, String nom, String prenom, int age)**

**{**

**this.insee = insee;**

**this.nom = nom;**

**this.prenom = prenom;**

**this.age = age;**

**this.adresse = "adresse inconnue";**

**this.numeroCarteIdentite = accederFichierNational(insee);**

**if (this.numeroCarteIdentite==null)**

**System.out.println("Le numero de caret d'identite de "+insee+" est inconnue";**

**}**

Q2-

La classe StringTokenizer est utilisée pour extraire, depuis une chaine de caractère, des "tokens (ou mots)" qui sont séparés par des caractères spécifiques (souvent des caractères spéciaux). Par exemple, extraire les mots d'une phrase espacés par des blancs, extraire des noms de répertoire d'un path espacés par le caractère "/".

Q3-

En programmation objet, un attribut static d'une classe est un attribut qui est commun à toutes les instances de la classe. Il est souvent initialisé dans le corps de la classe. Il peut être utilisé dans toutes les méthodes (non statiques et statiques) de la classe. Par exemple un attribut statique est une constante : public static double PI = 3.141592654.

**2ème PARTIE: PROGRAMMATION (avec document)**

**Exercice 1 [25 points]**

**public class RendezVous**

**{**

**private String date;**

**private int duree; // en minutes**

**private String objet;**

**private String lieu;**

**private ArrayList<String> invites;**

**public RendezVous()**

**{**

**date=dateAujourdhui(); //date système d'aujourd'hui**

**duree=60;**

**objet="sans objet";**

**lieu="";**

**invites=new ArrayList<String>();**

**}**

**public RendezVous(String date,String objet)**

**{**

**this.date=date;**

**this.duree=60;**

**this.objet=objet;**

**this.lieu="";**

**this.invites=new ArrayList<String>();**

**}**

**public RendezVous(String date,int duree,String objet,String lieu,**

**ArrayList<String> invites)**

**{**

**this.date=date;**

**this.duree=duree;**

**this.objet=objet;**

**this.lieu=lieu;**

**this.invites=invites;**

**}**

**public void saisir()**

**{**

**Terminal.ecrireString("Date:");**

**String sdate = Terminal.lireString();**

**Terminal.ecrireString("Duree:");**

**String sduree = Terminal.lireString();**

**Terminal.ecrireString("Objet:");**

**String sobjet = Terminal.lireString();**

**Terminal.ecrireString("Lieu:");**

**String slieu = Terminal.lireString();**

**if (sdate.equals(""))**

**{ Terminal.ecrireStringln("Erreur:La date est vide"); return; }**

**if (sobjet.equals(""))**

**{ Terminal.ecrireStringln("Erreur:La date est vide"); return; }**

**this.date=sdate;**

**if (sduree.equals("")) this.duree=0;**

**else this.duree=Integer.parseInt(sduree);**

**this.objet=sobjet;**

**this.lieu=slieu;**

**ArrayList<String> liste = new ArrayList<String>();**

**String invite;**

**do{**

**Terminel.ecrireString("invite:");**

**invite=Terminal.lireString();**

**liste.add(invite);**

**}while(! invite.equals(""));**

**this.invites=liste;**

**}**

**}**

**publi class Agenda**

**{**

**private ArrayList<RendezVous> rdvs;**

**public Agenda**

**{**

**rdvs = new ArrayList<RendezVous>();**

**}**

**public void ajouterRdv()**

**{**

**RendezVous rdv = new RendezVous();**

**rdv.saisir();**

**rdvs.add(rdv);**

**}**

**}**

**Exercice 2 [15 points]**

**public ArrayList<String> exercice2(ArrayList<String> phrases)**

**{**

**ArrayList<String> resultats = new ArrayList<String>');**

**String[) exceptions = {"l'", "le", "la", "les", "un, "une", "des"};**

**for(String phrase : phrases)**

**{**

**StringTokenizer strtok = new StringTokenizer(phrase);**

**while(strtok.hasMoreTockens())**

**{**

**String mot = strtok.nextToken();**

**boolean trouve=false;**

**for(String s:exceptions)**

**if (mot.equals(s)) trouve=true;**

**if (!trouve)**

**{**

**if (! resultats.contains(mot))**

**resultats.add(mot);**

**}**

**}**

**}**

**}**