Devoir java

Examen 2018

Réalisé par : Kanane SOUFIANE

Question 1 :

@override

public int compareTo(Etudiant etudiant){

return this.cne.compareTo(etudiant.cne);

}

Question 2 :

public PFE(string sujet,Encadrant encadrant){

this.sujet=sujet;

this.encadrant=encadrant;

this.groupe = new TreeSet();

}

Question 3 :

public class MaxEtudiantsDepasse extends Exception{

public MaxEtudiantsDepasse(message){

super(message)

}

Question 4 :

public void ajouterGroupe(set<Etudiant> groupe) throws MaxEtudiantsDepasse{

if((this.groupe.size()+groupe.size)>3) {throw MaxEtudiantDepasse("la taille du groupe que vous voulez ajouter est grande")}

else{

this.groupe.addAll(groupe);

}

}

Question 5 :

public Encadrant(String nom, Set<PFE> projets){

this.nom=nom;

this.projets=projets;

}

Question 6 :

public void supprimerProjet(PFE projet){

if (this.projets.contains(projet){this.projets.remove(projet);}

else{systeme.out.println("projet introuvable");}

}

Question 7 :

public int NombreEtudiantsParEncadrant(){

int nbr=0;

for(pfe p:projets){

nbr=nbr+p.getGroupe.size();

}

return nbr ;

}

Question 8 :

public Responsable(String nom, Set<PFE> projets){

super(nom,projets);

listEncadrants=new ArrayList<Encadrant>();

}

Question 9 :

public void echangerEtudiants(Etudiant etudiant1, Etudiant etudiant2)throws EtudiantSansPFEException{

set<PFE> s = new treeset<PFE>() ;

for(Encadrant enc : listEncadrants){

s .addAll(enc.getProjets);

}

PFE pfe1=null;

PFE pfe2=null;

for(PFE p:s){

if (p.getGroupe.contains(etudiant1)) {

pfe1=p;

}

else if (p.getGroupe.contains(etudiant2)){

pfe2=p;

}

}

if(pfe1==null && pfe2==null){throw EtudiantSansPFEException("les etudiants "+etudiant1+" et "+etudiant2+"ne sont pas itegres dans des groupes pfe")

else if (pfe1==null){throw EtudiantSansPFEException("l etudiant"+etudiant1+"n est pas integré dans un groupe pfe") }

else if (pfe2==null){throw EtudiantSansPFEException("l etudiant"+etudiant2+"n est pas integré dans un groupe pfe") }

else if (pfe1.getSujet.equals(pfe2.getSujet)){throw EtudiantSansPFEException("les deux etudiants appartiennent au meme groupe") }

else{

pfe1.remplacerEtudiant(etudiant1,etudiant2);

pfe2.remplacerEtudiant(etudiant2,etudiant1);

}

}

Question 10 :

public void transfererGroupe(PFE pfeSource, String NouveauSujet, Encadrant nouveauEncadrant)throws PFESansGroupeException{

set<Etuddiant> s= new set<Etudiant>();

if(pfeSource.getGroupe==null) {throw PFESansGroupeException("ce pfe n as pas de groupe")}

else {

s.addAll(pfeSource.getGroupe);

PFE nouveauPfe=new PFE(NouveauSujet,nouveauEncadrant);

nouveauPfe.ajouterGroupe(s);

pfeSource.getGroupe.removeAll();

}

}

Question 11 :

public List<Etudiant> getListEtudiantsTriée() {

list<Etudiant> l=new ArrayList<Etudiant>();

for(Encadrant e : listEncadrants){

for(PFE p : e.getProjets){

for(Etudiant et : p.getGroupe){

l.add(et);

}

}

}

l.sort();

return l;

}

Question 12 :

public void repartirPFE(Set<PFE> projets,List<Etudiant> etudiants){

Random r = new Random();

int nbe=listEncadrants.size();

for(PFE p : projets){

int random1=0;

while(random1==0){

random1=r.getInt(4);

}

for(int i=0;i<random1;i++){

if(etudiants.size()>0){

int random2=r.getInt(etudiants.size());

p.ajouterEtudiant(etudiants.get(random2));

p.remove(random2);

}

}

}

for(Encadrant e : listEncadrants){

if(projets.size()>0){

int random3=r.getInt(projets.size());

for(int i=0;i<random3;i++){

e.ajouterProjet(projets.iterator().next())

projets.remove(projets.iterator().next());

}

}

}

}