

Conception d'application

Groupe : NimporteQuoi

Membres : Loana MOTTAIS, Anouk GOUHIER-DUPUIS, Lin WAN, Mohammed TAHRI, Aymane AIT CHARKI

ReadMe

JavaDoc pour l'interface non générée : nous n'avons pas réussi à la générer avec Eclipse.
JavaDoc pour le reste du projet dans le dossier projet_events/doc.

Avancement du projet

Implémentation des classes & tests

Implémenté	A terminé	A faire
<ul style="list-style-type: none">- Classe Membres- Classe GestionMembres- Classe Evenement- Classe GestionEvenement- Classe GestionAssociation- TestGestionAssociation- TestMembre- TestGestionMembres- TestGestionEvenement		
<ul style="list-style-type: none">- TestEvenement		

JavaDoc

Fait	A terminé	A faire

- Classe GestionMembres
- Classe GestionEvenement
- Classe GestionAssociation
- Classe Membres
- Classe Evenement
- TestGestionEvenement
- TestGestionAssociation
- TestEvenement
- TestMembre
- TestGestionMembres

Plan de tests

Classe GestionMembres

Attributs de la classe TestGestionMembres

- gestionVideTest ⇒ gestionnaire de membre vide (.ensembleMembres=null, .president=null)
- gestion Complète ⇒ gestionnaire de membre dont les attributs ont des valeurs
(.ensembleMembres → liste de quelques membres, .president → le premier membre)

Méthode de Test	Actions	Attendu
testAjouterMembre	<p>Instancie un nouveau membre "membre1" et l'ajoute aux gestionnaires vide et complet.</p> <p>Vérifie que "membre1" a bien été ajouté à chaque gestionnaire.</p> <p>Puis cherche à ré-ajouter ce même membre au gestionnaire complet.</p>	<p>Ajout du membre à la liste .ensembleMembres de chaque gestionnaire.</p> <p>Pas de deuxième ajout du membre au gestionnaire complet (car déjà présent dans la liste) → test avec assertFalse.</p>

Méthode de Test	Actions	Attendu
testSupprimerMembre	<p>Instancie un nouveau membre (ré-instancie un de ceux du gestionnaire complet, autre que le président) et cherche à le supprimer des gestionnaire vide et complet.</p> <p>Puis instancie un nouveau membre (ré-instancie le membre président de l'attribut gestionComplete), et le supprime de gestionComplete</p>	<p>Ne supprime rien de la liste vide (vérification avec assertFalse).</p> <p>Supprime le membre du gestionnaire complet. Et vérifie qu'il n'est plus dans la liste ensembleMembres.</p> <p>Supprime le membre président du gestionnaire complet, remet l'attribut président à null.</p>
testDesignerPresident	<p>Instancie 2 nouveaux membres :</p> <p>membre1 = ré-instancie un de ceux du gestionnaire complet</p> <p>membre2 = instancie un complètement nouveau</p> <p>Cherche à désigner ces membres comme président de l'association pour chaque gestionnaire.</p>	<p>gestionVideTest : il ne se passe rien, la liste des membres est vide donc aucun des membres instancié ne peut être désigné comme président.</p> <p>gestionComplete : désigne membre1 comme président, renvoie false pour l'essai avec membre2 qui ne fait pas partie de la liste des membres.</p>
testEnsembleMembres	<p>Initialise un set de membres vide (pour le comparer avec le retour de la méthode ensemblesMembres() sur le gestionnaire vide)</p> <p>Récupère la liste ensembleMembres de chaque gestionnaire, compare ce qui est récupéré à l'attribut attendu avec un assertEquals.</p>	<p>gestionVideTest : l'attribut.ensembleMembres est un set vide (comparaison avec le set vide initialisé)</p> <p>gestionComplete : vérification avec assertEquals que l'objet retourné par la fonction ensembleMembres et l'attribut .ensembleMembres du gestionnaires sont les mêmes.</p>
testPresident	<p>Récupère l'attribut .président de chaque gestionnaire</p>	<p>gestionVideTest : vérification avec assertNull que l'objet retourner par la fonction president() est bien null</p> <p>gestionComplete : vérification avec assertEquals que l'objet retourné par la fonction president() et l'attribut .president du gestionnaire sont bien les mêmes</p>

Classe GestionEvenement

Méthode de Test	Actions	Attendu
testAjoutEvenement	<p>Ajout dans un gestionnaire vide d'un évènement.</p> <p>Ajout dans un gestionnaire non vide d'un événement correct, d'un au même moment qu'un déjà présent, d'un événement incorrect et d'un événement déjà présent.</p>	<p>1 évènement est bien créé dans la liste vide</p> <p>1 évènement est créé dans la liste non vide et les 3 autres ne sont pas créés</p>
testSuppressionEvenement	<p>Suppression d'un événement,</p> <p>Suppression du même élément un deuxième fois</p>	<p>1 evenement supprimer de la liste</p> <p>Rien ne se passe car l'élément n'existe pas dans la liste</p>
testAvenirEvenement	<p>récupération des évènement à venir de deux gestion d'évènement différents</p> <p>et on compte si la nouvelle liste créée contient le bon nombre d'élément</p>	<p>Un seul évènement à venir existe dans le premier gestionnaire</p> <p>et zéro événement dans le deuxième gestionnaire</p>
testInscriptionEvenement	<p>Inscription d'un nouveau membre à un évènement</p> <p>Inscription d'un membre déjà inscrit à l'évènement</p> <p>Inscription d'un nouveau membre à un événement ne faisant pas partie du gestionnaire</p>	<p>Le membre est bien inscrit à l'évènement</p> <p>Le membre n'est pas inscrit</p> <p>Le membre n'est pas inscrit</p>

testAnnulerEvenement	<p>Annulation d'un événement présent dans le gestionnaire pour un membre inscrit</p> <p>Annulation d'un événement du gestionnaire pour un membre non inscrit</p>	<p>1 est supprimer du gestionnaire</p> <p>Et ensuite la fonction ne fait rien</p>
----------------------	--	---

Classe GestionAssociation

Méthode de Test	Actions	Attendu
testSauvegarderetChargerDonnees	Créer un nouveau événement et aussi un nouveau membre et les sauvegarder dans un fichier "Test.dat". Puis on le charge avec la fonction chargerDonnees et les stocke dans les attributs : myEnsembleEvenements et myEnsembleMembres	<p>Les événements et les membres sont bien stocké dans le fichier "Test.dat";</p> <p>myEnsembleEvenements non null</p> <p>myEnsembleMembres non null</p>
testGestionnaireEvenements	recupérer le résultat de fonction GestionnaireEvenements, et le comparer avec le gestionEvTest	le résultat de fonction gestionnaireEvenements et gestionEvTest sont equals.
testGestionnaireMembre	recupérer le résultat de fonction GestionnaireMembre, et le comparer avec gestionMemTest.	le résultat de fonction gestionnaireMembre et gestionmemTest sont equals

Classe Membre

Méthode de Test	Actions	Attendu
testSetlistEvenement	Ajouter un ensemble d'événement par la fonction setlistEvenement.	L'ensemble d'événement n'est pas vide.

testSetEvenement	Ajouter un événement dans 'membredemi' par la fonction setEvenement.	L'ensemble d'événement de 'membredemi' n'est pas vide.
testEnsembleEvenementsAvenir	Ajouter un événement avec la date à venir dans l'ensemble d'événement avec la fonction setevenement et puis utiliser la fonction ensembleEvenementAvenir pour récupérer l'ensemble des événements à venir.	L'ensemble des événements à venir n'est pas vide.
testDefinirInfoPer	Renouveler l'information personnelle de 'membrecomplet' avec l'information personnelle de 'membredemi', et puis comparer les deux information personnelle.	Les deux information personnelle sont equals.

Classe Evenement

Méthode de Test	Actions	Attendu
testCreationDate	Intancie avec creation date une variable et avec la fonction .of une autre identique Vérification de le bonne création et si oui ou non les dates sont identique	Les deux dates ne sont pas nulles Et son égales
testChevauchement	Test le chevauchement de trois évènement avec deux qui se chevauche et un à un moment et un endroit différent	Les évènements 1 et 2 ne se chevauche pas Les évènements 2 et 3 se chevauche

		Les évènement 3 et 2 ne se chevauche pas
testaddParticipant	Test avec l'ajout d'un membre nul, création d'un membre, ajout de ce membre, vérification de son bon ajout, deuxième ajout de même participant et vérification du non ajout de ce participant	<p>La fonction d'ajout retourne false</p> <p>Le premier membre est bien ajouté</p> <p>Le deuxième membre n'est pas ajouté</p>
TestdelParticipant	Test avec l'ajout d'un membre nul, création d'un nouveau membre, ajout puis suppression de ce membre, on vérifie la bonne suppression, deuxième suppression du même participant	<p>La fonction d'ajout retourne false</p> <p>Le premier membre est bien ajouté puis supprimé</p> <p>La deuxième suppression retourne faux</p>