

Auteurs:

Gloria IGIRANEZA, Walid EL OUAZIZ, Taha KALDA

TestEtudiant

Méthode de test	Actions	Attendu
testConstructeur	Vérifie que le constructeur initialise correctement un étudiant.	L'information personnelle, le mot de passe, et le numéro de l'étudiant sont correctement initialisés.
testGetUes	Vérifie que les UE sont vides a l'initialisation.	La liste des UE est vide.
testGetTd	Vérifie que Td est initialisé a 1.	La valeur de Td est -1.
testGetTp	Vérifie que Tp est initialisé a 1.	La valeur de Tp est -1.
testGetMessages	Vérifie que la liste des messages est vide a l'initialisation	La liste des messages est vide.
testRecevoirMessage	Vérifie que l'étudiant reçoit correctement un message	Le message est correctement ajouté à la liste des messages.
testMarquerMessageLu	Vérifie que l'étudiant peut marquer un message comme lu	Le message est correctement marqué comme lu, et la liste des messages non lus est vide.
testSetTd	Modifie le groupe de TD de l'étudiant à 10	La valeur de Td est correctement modifiée à 10.
testSetTp	Modifie le groupe de TP de l'étudiant à 10	La valeur de Tp est correctement modifiée à 10.

TestGestionEtudiant

Méthode de test	Actions	Attendu
testInscriptionOk()	Vérifier si l'inscription d'un nouvel étudiant est réussie.	Vérifie que le numéro d'étudiant attribué est positif et que l'étudiant est ajouté à l'ensemble des étudiants.
testConnexionEtDeconnexion())	On instancie une nouvelle UE et on appelle choisirOption	Une exception NonConnecteException doit être levée car aucun étudiant n'est connecté.

<code>testConnexionAvecIdentifiantsInexistants()</code>	Teste la connexion avec des identifiants inexistants.	Vérifie que la connexion échoue avec des identifiants inexistants.
<code>testChoisirOptionAvecConnexion()</code>	Teste la fonctionnalité de choix d'option par un étudiant connecté.	Vérifie que l'étudiant peut choisir une option et que cette option est ensuite listée parmi les enseignements suivis par l'étudiant.
<code>testChoisirOptionSansConnexion()</code>	Teste le comportement de la méthode choisirOption lorsque l'étudiant n'est pas connecté.	Vérifie qu'une NonConnecteException est levée dans ce cas.
<code>testChoisirOptionDejaChoisie()</code>	Teste le cas où un étudiant connecté tente de choisir une option déjà choisie.	Vérifie que la méthode retourne false dans ce cas.
<code>testChoisirOptionAvecUeNonDisponible()</code>	Teste la tentative de choisir une option avec une UE non disponible.	Vérifie que la méthode retourne false dans ce cas.
<code>testChoisirOptionInterditePourMemeOption()</code>	Teste la tentative de choisir plusieurs fois la même option.	Vérifie que la méthode retourne false dans ce cas.
<code>testChoisirOptionAvecUeMaxOptionsAtteint()</code>	Teste la tentative de choisir plusieurs options différentes jusqu'à atteindre la limite.	Vérifie que la méthode retourne false après avoir atteint la limite d'options.
<code>testChoisirOptionAvecCapaciteMaximaleAtteinte()</code>	Teste la tentative de choisir une option dont la capacité maximale est déjà atteinte.	Vérifie que la tentative de choix échoue comme prévu.
<code>testGetNumeroGroupeTravauxDiriges()</code>	Teste si l'exception NonConnecteException est levée lors de l'accès au numéro de TD sans connexion.	Vérifie que l'exception est levée.
<code>testGetNumeroGroupeTravauxPratiques()</code>	Teste si l'exception NonConnecteException est levée lors de l'accès au numéro de TP sans connexion.	Vérifie que l'exception est levée.
<code>testEnseignementsSuivis()</code>	Vérifie si l'étudiant est connecté pour accéder aux différentes UEs.	Vérifie que l'exception de type NonConnecteException est levée si l'étudiant est déconnecté.

testListeTousMessages	Tentative d'obtention de la liste de tous les messages sans connexion.	Exception de type NonConnecteException levée.
testListeMessagesNonLus	Tentative d'obtention de la liste des messages non lus sans connexion.	Exception de type NonConnecteException levée.
testInscriptionFinalisee	Tentative de vérification si l'inscription est finalisée sans connexion .	Exception de type NonConnecteException levée.
testInscriptionAvecNomsLongsEtCaracteresSpeciaux	Inscription d'un nouvel étudiant avec des noms très longs et des caractères spéciaux	Attend que l'inscription réussisse avec un identifiant d'étudiant positif.
testGetInformationPersonnelle	Tente d'obtenir des informations personnelles sans être connecté.	Exception de type NonConnecteException levée.
testEnseignementsSuivisCo	Connecte un étudiant et vérifie la liste des cours suivis.	Attend que la liste soit vide pour un nouvel étudiant.
testListeTousMessagesCo	Connecte un étudiant et vérifie la liste de tous les messages.	Attend que la liste contienne au moins un message.
testListeMessagesNonLusCo	Connecte un étudiant et vérifie la liste des messages non lus.	Attend que la liste contienne au moins un message non lu.
testInscriptionFinaliseeCo	Connecte un étudiant et vérifie si l'inscription est finalisée.	Attend que le résultat soit faux pour un nouvel étudiant
testInscriptionMdpVide	Tente de s'inscrire avec un mot de passe vide.	Attend que l'inscription échoue avec une valeur de retour de -1.
testInscriptionMdpNull	Tente de s'inscrire avec un mot de passe nul.	Attend un numéro d'étudiant égal à -1, indiquant un échec d'inscription.
testGetInformationPersonnelleCo	Connecte un étudiant et tente d'obtenir ses informations personnelles.	Attend que les informations personnelles obtenues correspondent aux informations initiales de l'étudiant.
testNombreOptionsAvecConnexion	Connecte un étudiant et crée une nouvelle formation avec un nombre défini d'options. Vérifie le nombre d'options	Attend que le nombre d'options obtenu corresponde au nombre défini dans la formation.

testNombreOptionsSansConne xion	obtenues par l'étudiant. Tente d'obtenir le nombre d'options sans être connecté.	Attend une levée de l'exception NonConnecteException.
testNombreOptionsApresModi fication	Connecte un étudiant, crée une nouvelle formation, modifie le nombre d'options dans la formation, et vérifie le nombre d'options obtenues par l'étudiant.	Attend que le nombre d'options obtenu reflète correctement les changements effectués dans la formation.
testNombreOptionsNegatif	Connecte un étudiant, crée une nouvelle formation avec un nombre d'options défini comme négatif, et vérifie le nombre d'options obtenues par l'étudiant.	Attend que le nombre d'options obtenu soit égal à -1, indiquant une définition incorrecte du nombre d'options.

TestFormation

Méthode de test	Actions	Attendu
testConstructeur	Vérifie la création correcte de la formation.	Le nom de la formation doit être "Informatique".
testNomResponsable	Vérifie la récupération correcte du nom du responsable.	Le nom du responsable doit être "John Doe"
testEmailResponsable	Vérifie la récupération correcte de l'email du responsable.	L'email du responsable doit être " john.doe@formation.fr ".
testTpInitial	Vérifie la valeur initiale de TP.	La valeur initiale de TP doit être 0.
testTdInitial	Vérifie la valeur initiale de TD.	La valeur initiale de TD doit être 0.
testTpmaxInitial	Vérifie la valeur initiale de TPmax.	La valeur initiale de TPmax doit être 0.
testTdmaxInitial	Vérifie la valeur initiale de TDmax.	La valeur initiale de TDmax doit être 0.
testOptminInitial	Vérifie la valeur initiale de Optmin.	La valeur initiale de Optmin doit être -1.
testUesNonNul	Vérifie que la liste des UE n'est pas nulle.	La liste des UE ne doit pas être nulle.
testUesVide	Vérifie que la liste des UE est vide initialement.	La liste des UE doit être vide initialement.
testSetTp	Modifie la valeur de TP et vérifie la modification.	La valeur de TP doit être modifiée à 2.
testSetTd	Modifie la valeur de TD et vérifie la modification.	La valeur de TD doit être modifiée à 3.
testSetTpmax	Modifie la valeur de TPmax et vérifie la modification.	La valeur de TPmax doit être modifiée à 30.
testSetTdmax	Modifie la valeur de TDmax et vérifie la modification.	La valeur de TDmax doit être modifiée à 40.
testSetOptmin	Modifie la valeur de Optmin et vérifie la modification.	La valeur de Optmin doit être modifiée à 1.

testListeUe	Ajoute des UE à la liste et vérifie si elles sont correctement récupérées.	La liste des UE doit contenir les UE ajoutées.
-------------	--	--

TestInformationPersonnelle

Méthode de test	Actions	Attendu
testAge25Bastique	Positionne un âge de 25 ans pour l'information basique.	L'âge de l'information basique doit être égal à 25.
testAgeNegatifBastique	Positionne un âge négatif pour l'information basique.	L'âge de l'information basique ne doit pas être égal à -20.
testAgeNegatifCompleet	Positionne un âge négatif pour l'information complète.	L'âge de l'information complète doit rester le même qu'avant.
testAdresseNonNull	Vérifie que l'adresse n'est pas nulle lors de la création de l'information.	Les adresses de l'information basique et complète ne doivent pas être nulles.
testSetterAdresseNull	Positionne une adresse nulle sur une information existante.	L'adresse de l'information complète ne doit pas être nulle après la tentative de modification.
testConstructeur	Vérifie que les paramètres des constructeurs sont correctement gérés.	Le nom et le prénom doivent être corrects, l'adresse doit être non nulle et l'âge doit être positif.
TestConstructeurException.	Vérifie que le constructeur lance correctement une exception lorsque le nom ou le prénom est vide.	Le constructeur doit lever une exception InformationPersonnelleException lorsque le nom est vide.

TestSauvegarde

Méthode de test	Actions	Attendu
-----------------	---------	---------

testSauvegarderEtChargerDonnees	Teste la sauvegarde et le chargement des données.	Les instances de GestionFormation et GestionEtudiant avant et après la sauvegarde doivent être égales.
testChargerDonneesAvecFichierInexistant	Teste le chargement des données avec un fichier inexistant.	Une exception de type IOException doit être levée lors du chargement des données.

TestUniteEnseignement

Méthode de test	Actions	Attendu
testConstructeur	Teste le constructeur de l'unité d'enseignement.	Les propriétés de l'unité d'enseignement (nom, nom du responsable, type) sont correctes.
testCapaciteInitiale	Vérifie que la capacité initiale est correcte.	La capacité initiale est -1.
testTypeInitiale	Vérifie que le type initial est correct.	Le type initial est UniteEnseignement.TypeUe.OBG.
testSetOptionnelCapacitePositive	Teste le changement en optionnel avec une capacité positive.	Le type devient UniteEnseignement.TypeUe.OPT et la capacité est correcte.
testSetOptionnelCapaciteNegative	Teste le changement en optionnel avec une capacité négative.	Le type devient UniteEnseignement.TypeUe.OPT et la capacité est -1.
testToStringSansCapacite	Teste la représentation textuelle sans capacité.	La représentation textuelle est correcte sans la capacité.
testToStringAvecCapacite	Teste la représentation textuelle avec capacité.	La représentation textuelle est correcte avec la capacité.
testEquals	Teste l'égalité entre deux unités d'enseignement.	Deux unités d'enseignement avec les mêmes noms et responsables sont considérées comme égales.

Suite de TestUniteEnseignement

testEqualsObjetDifferent	Teste l'égalité avec un objet de type différent.	Deux objets de types différents ne sont pas considérés comme égaux.
testEqualsNomDifferent	Teste l'égalité avec un objet de type différent.	Deux objets de types différents ne sont pas considérés comme

testEqualsNomResponsableDifférents	Teste l'égalité avec des noms de responsables différents.	égaux Deux unités d'enseignement avec des noms de responsables différents ne sont pas considérées comme égales.
testHashCode	Crée une autre unité d'enseignement (autreUe) avec le même nom et le même responsable que l'unité d'enseignement (ue) actuelle. Compare les codes de hachage des deux unités d'enseignement.	Attend que les codes de hachage des deux unités d'enseignement soient égaux.

TestGestionFormation

Méthode de test	Actions	Attendu
testCreerFormation	Appelle la méthode <code>creerFormation</code> de l'objet <code>GestionFormation</code> .	Vérifie que les attributs de la formation sont correctement initialisés.
<u>testCreerFormationEchec</u>	Tente de créer une formation avec des paramètres vides.	Assure que les attributs de la formation ne sont pas vides après l'échec de la création.
testAjouterEnseignementObligatoire	Ajoute une UE obligatoire à la formation.	Vérifie que la taille de la liste des UE obligatoires a augmenté de 1.
testAjouterEnseignementObligatoireEchec	Tente d'ajouter une UE obligatoire déjà existante.	Assure que l'ajout échoue et que la taille de la liste des UE obligatoires reste inchangée.
testAjouterEnseignementOptionnel	Ajoute une UE optionnelle avec une capacité spécifiée	Vérifie que la taille de la liste des UE optionnelles a augmenté de 1.
testAjouterEnseignementOptionnelEchec	Tente d'ajouter une UE optionnelle déjà existante.	Assure que l'ajout échoue et que la taille de la liste des UE optionnelles reste inchangée.

testDefinirNombreOptions	Définit le nombre d'options requises.	Vérifie que le nombre d'options est correctement défini.
testDefinirNombreOptionsNegatif	Tente de définir un nombre d'options négatif.	Assure que le nombre d'options reste inchangé.
testDefinirNombreOptionsDejaDefini	Tente de redéfinir le nombre d'options déjà défini.	Assure que le nombre d'options reste inchangé.
testSetGetTailleGroupeDirige	Définit et récupère la taille d'un groupe de TD.	Vérifie que la taille du groupe de TD est correctement définie et récupérée.
testSetGetTailleGroupePratique	Définit et récupère la taille d'un groupe de TP.	Vérifie que la taille du groupe de TP est correctement définie et récupérée.
testSetTailleGroupeDirigeNegatif	Tente de définir une taille de groupe de TD négative.	Assure que la taille du groupe de TD reste inchangée.
testSetTailleGroupePratiqueNull	Tente de définir une taille de groupe de TP à 0.	Assure que la taille du groupe de TP reste inchangée.
testSetGetTailleGroupeDirigeDejaDefini	Tente de redéfinir la taille d'un groupe de TD déjà définie.	Assure que la taille du groupe de TD reste inchangée.
testAttribuerAutomatiqueMentGroupes	Appelle la méthode attribuerAutomatiqueMentGroupes pour attribuer automatiquement les groupes.	Vérifie que tous les étudiants ont un groupe de TD et de TP attribué.
testChangerGroupe	Change manuellement le groupe d'un étudiant.	Vérifie que le changement de groupe est effectué correctement.
testNombreGroupesTravauxDiriges	Appelle la méthode nombreGroupesTravauxDiriges .	Vérifie que le nombre de groupes de TD est initialement à 0.
testNombreGroupesTravauxPratiques	Appelle la méthode nombreGroupesTravauxPratiques .	Vérifie que le nombre de groupes de TP est initialement à 0.

testListeEtudiantsGroupeDirige	Appelle la méthode listeEtudiantsGroupeDirige avec un groupe inexistant.	Vérifie que la liste des étudiants du groupe de TD inexistant est nulle.
testListeEtudiantsGroupePratique	Appelle la méthode listeEtudiantsGroupePratique avec un groupe inexistant.	Vérifie que la liste des étudiants du groupe de TP inexistant est nulle.
testListeEtudiantsOption	Appelle la méthode listeEtudiantsOption avec une option inexistante.	Vérifie que la liste des étudiants de l'option inexistante est nulle.
testEnseignementsObligatoires	Ajoute une UE obligatoire et appelle la méthode enseignementsObligatoires .	Vérifie que la liste des UE obligatoires contient l'UE ajoutée.
testEnseignementsOptionnels	Ajoute une UE optionnelle et appelle la méthode enseignementsOptionnels .	Vérifie que la liste des UE optionnelles contient l'UE ajoutée.
testAjoutUeExistante	Tente d'ajouter une unité d'enseignement (UE) obligatoire à la gestion de la formation.	Attend un échec lors de la tentative d'ajout de la même UE obligatoire déjà existante dans la gestion de la formation.