

Tests unitaires de méthodes de la classe Evenement

setId :

Test unitaire de : *public CodeErreur setId(int unId);*

Initialisation : *this.id = 0;*

Résultat :

- null si la définition de l'id est un succès
- CodeErreur.ID_NEGATIF si unId est inférieur ou égal à 0

Initialisation this.id	Paramètre unId	Modification this.id	Résultat CodeErreur
0	1	1	null
0	3	3	null
0	0	0	ID_NEGATIF
0	-1	0	ID_NEGATIF
0	-3	0	ID_NEGATIF

setNom :

Test unitaire de : `public CodeErreur setNom(String unNom);`

Initialisation : `this.nom = null;`

Résultat :

- null si la définition du nom est un succès
- NOM_ESPACE_EN_TROP si unNom commence ou termine par un espace
- NOM_TROP_LONG si unNom dépasse la taille maximum
- NOM_NULL si unNom est null
- NOM_VIDE si unNom est une chaîne vide

Initialisation this.nom	Paramètre unNom	Modification this.nom	Résultat CodeErreur
null	"Repas de Noël"	"Repas de Noël"	null
null	" espaces avant"	null	NOM_ESPACE_EN_TROP
null	"espaces après "	null	NOM_ESPACE_EN_TROP
null	"..." (String de plus de 50 caractères)	null	NOM_TROP_LONG
null	null	null	NOM_NULL
null	" "	null	NOM_VIDE

setDescriptif :

Test unitaire de : `public CodeErreur setDescriptif(String unDescriptif);`

Initialisation : `this.descriptif = null;`

Résultat :

- null si la définition du nom est un succès
- DESCRIPTIF_ESPACE_EN_TROP si unDescriptif commence ou termine par un espace
- DESCRIPTIF_TROP_LONG si unDescriptif dépasse la taille maximum
- DESCRIPTIF_NULL si unDescriptif est null

Initialisation this.descriptif	Paramètre unDescriptif	Modification this.descriptif	Résultat CodeErreur
null	“Organisation d’un repas de Noël pour l’année 2022. Au programme un repas et différentes activités.”	“Organisation d’un repas de Noël pour l’année 2022. Au programme un repas et différentes activités.”	null
null	“”	“”	null
null	“ Un espaces avant”	null	DESCRIPTIF_ESPACE_EN_TROP
null	“Un espaces après “	null	DESCRIPTIF_ESPACE_EN_TROP
null	“...” (String de plus de 500 caractères)	null	DESCRIPTIF_TROP_LONG
null	null	null	DESCRIPTIF_NULL

setDate :

Test unitaire de : `public CodeErreur setDate(Date uneDate);`

Initialisation : `this.date = null;`

Résultat :

- null si la définition de la date est un succès
- DATE_NULL si uneDate est null
- DATE_PASSEE si uneDate est une date déjà passée

Initialisation this.descriptif	Paramètre unDescriptif	Modification this.descriptif	Résultat CodeErreur
null	uneDate = Date(2022-1900 11, 25, 19, 30);	this.date=Date(2022-1900 11, 25, 19, 30);	null
null	/*Date déjà passée*/ uneDate = Date(2021-1900 11, 25, 19, 30);	null	DATE_PASSEE
null	null	null	DATE_NULL

setLieu :

Test unitaire de : `public CodeErreur setLieu(String unLieu);`

Initialisation : `this.lieu = null;`

Résultat :

- null si la définition du lieu est un succès
- LIEU_ESPACE_EN_TROP si unLieu commence ou termine par un espace
- LIEU_TROP_LONG si unLieu dépasse la taille maximum
- LIEU_NULL si unLieu est null
- LIEU_VIDE si unLieu est une chaîne vide

Initialisation this.lieu	Paramètre unLieu	Modification this.lieu	Résultat CodeErreur
null	"Salle des fêtes, rue de la Paix, 29200 Brest"	"Salle des fêtes, rue de la Paix, 29200 Brest"	null
null	" espaces avant"	null	LIEU_ESPACE_EN_TROP
null	"espaces après "	null	LIEU_ESPACE_EN_TROP
null	"..." (String de plus de 100 caractères)	null	LIEU_TROP_LONG
null	null	null	LIEU_NULL
null	""	null	LIEU_VIDE

setNbMaxPersonnes :

Test unitaire de : `public CodeErreur setNbMaxPersonnes(in unNbMaxPersonnes);`

Initialisation : `this.nbMaxPersonnes = 0;`

Résultat :

- null si la définition du nombre de personnes maximum est un succès
- NB_MAX_PERSONNES_TROP_PETIT si unNbMaxPersonnes est inférieur ou égal à 1

Initialisation this.nbMaxPersonnes	Paramètre unNbMaxPersonnes	Modification this.nbMaxPersonnes	Résultat CodeErreur
0	30	30	null
0	1	1	null
0	0	0	NB_MAX_PERSONNES_TROP_PETIT
0	-2	0	NB_MAX_PERSONNES_TROP_PETIT

setType :

Test unitaire de : `public CodeErreur setType(TypeEvenement unType);`

Initialisation : `this.type = null;`

Résultat :

- null si la définition du type est un succès
- TYPE_NULL si unType est null

Initialisation this.type	Paramètre unType	Modification this.type	Résultat CodeErreur
null	TypeEvenement.REPAS	TypeEvenement.REPAS	null
null	null	null	TYPE_NULL

setImage :

Test unitaire de : `public CodeErreur setImage(String uneImage);`

Initialisation : `this.image = null;`

Résultat :

- null si la définition de l'image est un succès

Initialisation this.image	Paramètre uneImage	Modification this.image	Résultat CodeErreur
null	"https://www.google.com/image/chat.gif"	"https://www.google.com/image/chat.gif"	null
null	""	""	null
null	null	null	null

Tests unitaires de méthodes de la classe Membre

setPseudo

Test unitaire de : `public CodeErreur setPseudo(String unPseudo)`

Initialisation : `this.pseudo = null;`

Résultat :

- null en cas de succès
- PSEUDO_TROP_LONG si le pseudo est trop long
- CodeErreur.PSEUDO_NULL si le pseudo est null
- CodeErreur.PSEUDO_VIDE si le pseudo est vide
- CodeErreur.PSEUDO_INVALIDE si le pseudo contient : espace,<,>|,.,*,*,?/,\\

Initialisation this.pseudo	Paramètre unPseudo	Modification this.pseudo	Résultat CodeErreur
null	Test	Test	null
null	123	123	null
null	test-	test-	null
null	testé	testé	null
null	null	null	PSEUDO_NULL
null	(vide)	null	PSEUDO_VIDE
null	espace entre	null	PSEUDO_INVALIDE
null	(pseudo trop long (>30 car)	null	PSEUDO_INVALIDE
null	[<> .~*?/\]	null	PSEUDO_INVALIDE
null	(espace)	null	PSEUDO_INVALIDE

setNom

Test unitaire de : `public CodeErreur setNom(String unNom)`

Initialisation : `this.nom = null;`

Résultat :

- null en cas de succès
- CodeErreur.NOM_TROP_GRAND si le nom est trop grand
- CodeErreur.NOM_NULL si le nom est null
- CodeErreur.NOM_VIDE si le nom est vide
- CodeErreur.NOM_HORS_ALPHABET si le nom contient des caractères autres que les lettres de l'alphabet

Initialisation this.nom	Paramètre unNom	Modification this.nom	Résultat CodeErreur
null	Pierre	Pierre	null
null	JeanPierre	Jean-Pierre	null
null	Nom à 30 caractères	Nom à 30 caractères	null
null	null	null	NOM_NULL
null	Nom à 31 caractères	null	NOM_TROP_GRAND
null	(vide)	null	NOM_VIDE
null	Jean Pierre	null	NOM_HORS_ALPHABET
null	Méric	null	NOM_HORS_ALPHABET
null	123	null	NOM_HORS_ALPHABET
null	(avec n'importe quel caractère spécial, sauf espace)	null	NOM_HORS_ALPHABET
null	(espace)	null	NOM_HORS_ALPHABET

setPrenom

Test unitaire de : `public CodeErreur setPrenom(String unPrenom)`

Initialisation : `this.prenom = null;`

Résultat :

- null en cas de succès
- CodeErreur.PRENOM_TROP_GRAND si le nom est trop grand
- CodeErreur.PRENOM_NULL si le nom est null
- CodeErreur.PRENOM_VIDE si le nom est vide
- CodeErreur.PRENOM_HORS_ALPHABET si le nom contient des caractères autres que les lettres de l'alphabet

Initialisation this.prenom	Paramètre unPrenom	Modification this.prenom	Résultat CodeErreur
null	Pierre	Pierre	null
null	JeanPierre	Jean-Pierre	null
null	Nom à 30 caractères	Nom à 30 caractères	null
null	null	null	PRENOM_NULL
null	Jean Pierre	null	PRENOM_HORS_ALPHABET
null	Méric	null	PRENOM_HORS_ALPHABET
null	123	null	PRENOM_HORS_ALPHABET
null	Nom à 31 caractères	null	PRENOM_TROP_GRAND
null	(avec n'importe quel caractère spécial, sauf espace)	null	PRENOM_HORS_ALPHABET
null	(vide)	null	PRENOM_VIDE
null	(espace)	null	PRENOM_HORS_ALPHABET

setLieuNaissance

Test unitaire de : `public CodeErreur setLieuNaissance(String unLieuNaissance)`

Initialisation : `this.lieuNaissance = null;`

Résultat :

- null si la définition du lieu de naissance est un succès
- CodeErreur.LIEU_NULL si le lieu de naissance est null
- CodeErreur.LIEU_NOM_INCORRECT si le nom du lieu de naissance comporte un caractère spécial
- CodeErreur.LIEU_TAILLE_INCORRECTE si le lieu de naissance ne comprend pas le nombre de caractères requis

Initialisation this.lieuNaissance	Paramètre unLieuNaissance	Modification this.lieuNaissance	Résultat CodeErreur
null	Brest	Brest	null
null	Saint-Vincent-Sur-Jard	Saint-Vincent-Sur-Jard	null
null	Nom de 50 caractères	Nom à 50 caractères	null
null	Méric	null	null
null	null	null	LIEU_NULL
null	Jean Pierre	null	LIEU_NOM_INCORRECT
null	123	null	LIEU_NOM_INCORRECT
null	Nom à 51 caractères	null	LIEU_TAILLE_INCORRECTE
null	(vide)	null	LIEU_TAILLE_INCORRECTE
null	(espace)	null	LIEU_NOM_INCORRECT

setDateNaissance

Test unitaire de : `public CodeErreur setDateNaissance(String uneDateNaissance)`

Initialisation : `this.dateNaissance = null;`

Résultat :

- null si la définition de la date de naissance est un succès
- Code_Erreur.DATE_NULL si la date est null
- CodeErreur.DATE_IMPOSSIBLE si la date de naissance est plus grande que la date courante
- CodeErreur.DATE_AGE_IMPOSSIBLE si l'âge du membre est supérieur à 130 ans

Initialisation this.dateNaissance	Paramètre unDateNaissance	Modification this.dateNaissance	Résultat CodeErreur
null	uneDate = Date(2001-1900 1, 1, 1, 1);	this.date=Date(2001 -1900 1, 1, 1, 1);	null
null	/*Date plus grande que la date actuelle */ uneDate = Date(2050-1900 1, 1, 1, 1);	null	DATE_IMPOSSIBLE
null	/*Age > 130 ans*/ uneDate = Date(1890-1900 1, 1, 1, 1);	null	DATE_AGE_IMPOSSIBLE
null	null	null	DATE_NULL

setVille

Test unitaire de : `public CodeErreur setVille(String uneVille)`

Initialisation : `this.ville = null;`

Résultat :

- null si la définition de la ville est un succès
- CodeErreur.VILLE_NOM_NULL si la ville est null
- CodeErreur.VILLE_NOM_INCORRECT si le nom de la ville comporte un caractère spécial
- CodeErreur.VILLE_TAILLE_INCORRECTE si la ville ne comprend pas le nombre de caractères requis

Initialisation this.ville	Paramètre uneVille	Modification this.ville	Résultat CodeErreur
null	Brest	Brest	null
null	Saint-Vincent-Sur-Jard	Saint-Vincent-Sur-Jard	null
null	Nom de 50 caractères	Nom à 50 caractères	null
null	Méric	null	null
null	null	null	VILLE_NOM_NULL
null	Jean Pierre	null	VILLE_NOM_INCORRECT
null	123	null	VILLE_NOM_INCORRECT
null	Nom à 51 caractères	null	VILLE_TAILLE_INCORRECTE
null	(vide)	null	VILLE_TAILLE_INCORRECTE
null	(espace)	null	VILLE_NOM_INCORRECT

setMail

Test unitaire de : `public CodeErreur setMail(String unMail)`

Initialisation : `this.mail = null;`

Résultat :

- null si la définition du mail est un succès
- CodeErreur.MAIL_NULL le mail est null
- CodeErreur.MAIL_VIDE si le mail est vide
- CodeErreur.MAIL_INVALIDE si le mail ne correspond pas à une adresse mail

Initialisation this.mail	Paramètre unMail	Modification this.mail	Résultat CodeErreur
null	tof.bob@example.com	tof.bob@example.com	null
null	tof@example.me.org	tof@example.me.org	null
null	tof@example.com	tof@example.com	null
null	null	null	MAIL_NULL
null	(vide)	null	MAIL_VIDE
null	tof@.com	null	MAIL_INVALIDE
null	tof@exemple	null	MAIL_INVALIDE
null	tof.example.com	null	MAIL_INVALIDE
null	tof..bob@example.com	null	MAIL_INVALIDE
null	@example.me.org	null	MAIL_INVALIDE
null	tof#@exemple.me.org	null	MAIL_INVALIDE

setMotDePasse

Test unitaire de : `public CodeErreur setMotDePasse(String unMotDePasse)`

Initialisation : `this.motDePasse = null;`

Résultat :

- null si la définition du mot de passe est un succès
- CodeErreur.MDP_NULL si le mot de passe est null
- CodeErreur.MDP_VIDE si le mot de passe n'est composé d'aucun caractère
- CodeErreur.MDP_TAILLE_INCORRECTE si le mot de passe est trop court ou trop long

Initialisation this.motDePasse	Paramètre unMotDePasse	Modification this.motDePasse	Résultat CodeErreur
null	motDePasse	null	null
null	123456789	null	null
null	1234567\	null	null
null	123456\"	null	null
null	(6 caractères)	null	null
null	(30 caractères)	null	null
null	(caractères spéciaux)	null	null
null	null	null	MDP_NULL
null	(vide)	null	MDP_NULL
null	(5 caractères)	null	MDP_TAILLE_INCORRECTE
null	(31 caractères)	null	MDP_TAILLE_INCORRECTE

Tests unitaires de méthodes de la classe Gestion

ajouterEvenement :

Test unitaire de : `public ArrayList<CodeErreur> ajouterEvenement(String unNom, String unDescriptif, String unImage, Date uneDate, String unLieu, int unNbMaxPersonnes, TypeEvenement unType)`

Résultat :

- null : ajout réussi
- ArrayList<CodeErreur> : erreur de création de l'événement

Initialisation this.listeEvenements	Paramètres (unNom, unDescriptif, unImage, uneDate, unLieu, unNbMaxPersonnes, unType)	Modification this.listeEvenements	Résultat ArrayList<CodeErreur>
[]	("Un événement", "Description de l'événement", "url", "27/06/22 15:30", "adresse", 500, TypeEvenement.REPAS)	[(1, "Un événement", "Description de l'événement", "url", "27/06/22 15:30", "adresse", 500, TypeEvenement.REPAS)]	null
[(1, "Un événement", "Description de l'événement", "url", "27/06/22 15:30", "adresse", 500, TypeEvenement.REPAS)]	("ag", "assemblée générale", "url", "08/06/22 12:45", "bureau", 20, TypeEvenement.AG)	[(1, "Un événement", "Description de l'événement", "url", "27/06/22 15:30", "adresse", 500, TypeEvenement.REPAS), (2, "ag", "assemblée générale", "url", "08/06/22", "bureau", 20, TypeEvenement.AG)]	null
[]	("Un événement", "Description de l'événement", "url", "27/06/22 15:30", "adresse", -90, TypeEvenement.REPAS)	[]	[CodeErreur.NB_MAX_PERSONNES_TROP_PETIT]
[]	("", "Description de l'événement", "url", "27/06/22 15:30", "adresse", 500, TypeEvenement.REPAS)	[]	[CodeErreur.NOM_VIDE]
[]	("", "Description de l'événement", "url", null, "adresse", 500, TypeEvenement.REPAS)	[]	[CodeErreur.NOM_VIDE, CodeErreur.DATE_NULL]

supprimerEvenement :

Test unitaire de : `public int supprimerEvenement(int unId)`

Résultat :

- null : suppression réussie
- ArrayList<CodeErreur> : non présent dans la liste

Initialisation this.listeEvenements	Paramètre (unId)	Modification this.listeEvenements	Résultat ArrayList<CodeErreur>
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(2)	[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS)]	null
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(0)	[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	[CodeErreur.EVENEMENT_INTROUVABLE]
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(-1)	[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	[CodeErreur.EVENEMENT_INTROUVABLE]
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(3)	[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	[CodeErreur.EVENEMENT_INTROUVABLE]

modifierEvenement :

Test unitaire de : `public ArrayList<CodeErreur> modifierEvenement(int unId, String unNom, String unDescriptif, String unImage, Date uneDate, String unLieu, int unNbMaxPersonnes, TypeEvenement unType)`

Résultat :

- null : modifications réussies
- ArrayList<CodeErreur> : erreur de modification

Initialisation this.listeEvenements	Paramètres (unId, unNom, unDescriptif, unImage, uneDate, unLieu, unNbMaxPersonnes, unType)	Modification this.listeEvenements	Résultat ArrayList<CodeErreur>
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(1," nom ", "Description de l'événement", "url","27/06/22 15:30","adresse", 200 , TypeEvenement.REPAS)	[(1," nom ", "Description de l'événement", "url","27/06/22 15:30","adresse", 200 , TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	null
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(3 ," nom ", "Description de l'événement", "url","27/06/22 15:30","adresse", 200 , TypeEvenement.REPAS)	[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	[CodeErreur.EVENEMENT_INTROUVABLE]
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(-1 ," nom ", "Description de l'événement", "url","27/06/22 15:30","adresse", 200 , TypeEvenement.REPAS)	[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	[CodeErreur.EVENEMENT_INTROUVABLE]
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse", -500 , TypeEvenement.REPAS)	[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	[CodeErreur.NB_MAX_PERSONNES_TROP_PETIT]

getEvenement :

Test unitaire de : *public Evenement getEvenement(int unld)*

Résultat :

- Evenement : l'évènement est trouvé
- null : l'évènement n'est pas trouvé

Initialisation this.listeEvenements	Paramètre (unld)	Résultat Evenement
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(1)	(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS)
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(0)	null
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(-1)	null
[(1,"Un événement", "Description de l'événement", "url","27/06/22 15:30","adresse",500, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url","08/06/22","bureau",20, TypeEvenement.AG)]	(3)	null

ajouterMembre :

Test unitaire de : `public ArrayList<CodeErreur> ajouterMembre(String unPseudo, String unNom, String unPrenom, String unLieuNaissance, Date uneDateNaissance, String uneVille, String unMail, String unMotDePasse)`

Résultat :

- null : ajout réussi
- ArrayList<CodeErreur> : erreur de création de l'événement

Initialisation this.listeMembre	Paramètres (<i>unPseudo, unNom, unPrenom, unLieuNaissance, uneDateNaissance, uneVille, unMail, unMotDePasse</i>)	Modification this.listeMembres	Résultat ArrayList<CodeErreur>
[]	("p1", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")	("p1", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")	null
[("p1", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]	("p2", "nn", "pr", "ll", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")	[("p1", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("p2", "nn", "pr", "ll", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")]	null
[]	(null, null, null, null, null, null, null, null)	[]	[PSEUDO_NULL, NOM_NULL, PRENOM_NULL, LIEU_NULL, DATE_NULL, VILLE_NULL, MAIL_NULL, MDP_NULL]
[("p1", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]	("p2", "nn", "pr", "ll", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")	[]	[CodeErreur.PSEUDO_DEJA_EXISTANT]

supprimerMembre :

Test unitaire de : `public int supprimerMembre(String unPseudo)`

Résultat :

- null : suppression réussie
- ArrayList<CodeErreur> : non présent dans la liste

Initialisation this.listeMembres	Paramètre (unPseudo)	Modification this.listeMembre	Résultat ArrayList<CodeErreur>
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), (("GillesDu25", "nn", "pr", "l", 07/11/2001, "v", "m@m.m", "mmdpppppp"))]	GillesDu25	[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]	null
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]	Robert65	[]	null
[]	Robert65	[]	[MEMBRE_INTROUVABLE]
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]	GillesDu25	[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]	[MEMBRE_INTROUVABLE]

modifierMembre :

Test unitaire de : `public ArrayList<CodeErreur> modifierMembre(String ancienPseudo, String unPseudo, String unNom, String unPrenom, String unLieuNaissance, Date uneDateNaissance, String uneVille, String unMail, String unMotDePasse)`

Résultat :

- null : modifications réussies
- ArrayList<CodeErreur> : erreur de modification

Initialisation this.listeMembres	Paramètres (ancienPseudo, unPseudo, unNom, unPrenom, unLieuNaissance, uneDateNaissance, uneVille, unMail, unMotDePasse)	Modification this.listeMembres	Résultat ArrayList<CodeErreur>
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("GillesDu25", "nn", "pr", "l", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")]	("Robert65", "Rob", "n", "pr", "l", 01/01/2022, "v", "m@m.m", "mdpppppp")	[("Rob", "n", "pr", "l", 01/01/2022, "v", "m@m.m", "mdpppppp"), ("GillesDu25", "nn", "pr", "l", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")]	null
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("GillesDu25", "nn", "pr", "l", 07/11/2001, "v", "mm@m.m", "mmdpppppp")]	("Robert65", "Robert65", "nn", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")	[("Robert65", "nn", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("GillesDu25", "nn", "pr", "l", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")]	null
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("GillesDu25", "nn", "pr", "l", 07/11/2001, "v", "mm@m.m", "mmdpppppp")]	("Robert65", "Robert65", null, null, "l", 07/11/2001, "v", "m@m.m", "mdpppppp")	[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("GillesDu25", "nn", "pr", "l", 07/11/2001, "v", "mm@m.m", "mmdpppppp")]	[NOM_NULL, PRENOM_NULL]
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("GillesDu25", "nn", "pr", "l", 07/11/2001, "v", "mm@m.m", "mmdpppppp")]	("Rob", "Robert", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")	[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("GillesDu25", "nn", "pr", "l", 07/11/2001, "v", "mm@m.m", "mmdpppppp")]	[MEMBRE_INTROUVABLE]
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("GillesDu25", "nn", "pr", "l", 07/11/2001, "v", "mm@m.m", "mmdpppppp")]	("Robert65", "GillesDu25", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")	[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("GillesDu25", "nn", "pr", "l", 07/11/2001, "v", "mm@m.m", "mmdpppppp")]	[PSEUDO_DEJA_EXISTANT]

getMembre :

Test unitaire de : `public Membre getMembre(String unPseudo)`

Résultat :

- Membre : le membre est trouvé
- null : le membre n'est pas trouvé

Initialisation this.listeMembres	Paramètre (unPseudo)	Résultat Membre
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]	Robert65	[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]	GillesDu25	null
[("Robert65", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]	null	null

Tests d'intégration de Gestion partie Evenement

Appel	Retour
getListeEvenements()	{}
ajouterEvenement("Un événement", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30),"adresse",500, TypeEvenement.REPAS)	null
getListeEvenement()	[(1, "Un événement", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30),"adresse",500, TypeEvenement.REPAS)]
ajouterEvenement("", "Description de l'événement", "url", null, "adresse", 500, TypeEvenement.REPAS)	[NOM_VIDE, DATE_NULL]
getListeEvenement()	[(1, "Un événement", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30),"adresse",500, TypeEvenement.REPAS)]
ajouterEvenement("ag", "assemblée générale", "url", new Date(2022-1900,06,8,12,45),"bureau",20, TypeEvenement.AG)	null
getListeEvenement()	[(1, "Un événement", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30), "adresse", 500, TypeEvenement.REPAS), (2, "ag", "assemblée générale", "url", new Date(2022-1900,06,8,12,45), "bureau", 20, TypeEvenement.AG)]
modifierEvenement(1,"nom", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30), "adresse", 200 , TypeEvenement.REPAS)	null
getListeEvenement()	[(1, " nom ", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30), "adresse", 200 , TypeEvenement.REPAS), (2, "ag", "assemblée générale", "url", new Date(2022-1900,06,8,12,45), "bureau", 20, TypeEvenement.AG)]
modifierEvenement(3 ,"nom", "Description de l'événement", "url", new Date(2022-1900, 05,28 ,15,30), "adresse", 200, TypeEvenement.REPAS)	[EVENEMENT_INTROUVABLE]
getListeEvenement()	[(1,"nom", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30), "adresse", 200, TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url", new

	Date(2022-1900,06,8,12,45), "bureau" , 20, TypeEvenement.AG)]
modifierEvenement(1,"nom", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30), "adresse", -500 , TypeEvenement.REPAS)	[NB_MAX_PERSONNES_TROP_PETIT]
getListeEvenement()	[(1,"nom", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30), "adresse", 200 , TypeEvenement.REPAS), (2,"ag", "assemblée générale", "url", new Date(2022-1900,06,8,12,45), "bureau" , 20, TypeEvenement.AG)]
supprimerEvenement(2)	null
getListeEvenement()	[(1,"nom", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30), "adresse", 200, TypeEvenement.REPAS)]
supprimerEvenement(0)	[EVENEMENT_INTROUVABLE]
getListeEvenement()	[(1,"nom", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30), "adresse", 200, TypeEvenement.REPAS)]
getEvenement(1)	(1,"nom", "Description de l'événement", "url", new Date(2022-1900,06,27,15,30), "adresse", 200, TypeEvenement.REPAS)
getEvenement(2)	null

Tests d'intégration de Gestion partie Membre

Appel	Retour
getListeMembres()	{}
ajouterMembre("p1", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")	null
getListeMembres()	[("p1", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]
ajouterMembre("p1", "nn", "pr", "ll", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")	[PSEUDO_DEJA_EXISTANT]
getListeMembres()	[("p1", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp")]
ajouterMembre("p2", "nn", "pr", "ll", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")	null
getListeMembres()	[("p1", "n", "pr", "l", 07/11/2001, "v", "m@m.m", "mdpppppp"), ("p2", "nn", "pr", "ll", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")]
modifierMembre("p1", "Rob", "n", "pr", "l", 01/01/2022, "v", "m@m.m", "mdpppppp")	null
getListeMembres()	[("Rob", "n", "pr", "l", 01/01/2022, "v", "m@m.m", "mdpppppp"), ("p2", "nn", "pr", "ll", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")]
modifierMembre("p1", "p1", "nn", "pr", "l", 01/01/2022, "v", "m@m.m", "mdpppppp")	[MEMBRE_INTROUVABLE]
getListeMembres()	[("Rob", "n", "pr", "l", 01/01/2022, "v", "m@m.m", "mdpppppp"), ("p2", "nn", "pr", "ll", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")]
modifierMembre("Rob", "Rob", null, null, "l", 01/01/2022, "v", "m@m.m", "mdpppppp")	[NOM_NULL, PRENOM_NULL]
getListeMembres()	[("Rob", "n", "pr", "l", 01/01/2022, "v", "m@m.m", "mdpppppp"), ("p2", "nn", "pr", "ll", 07/11/2001, "vv", "mm@m.m", "mmdpppppp")]
supprimerMembre("p2")	null
getListeMembres()	[("Rob", "n", "pr", "l", 01/01/2022, "v", "m@m.m", "mdpppppp")]
supprimerMembre("p2")	[MEMBRE_INTROUVABLE]

getListeMembres()	[(“Rob”, “n”, “pr”, “l”, 01/01/2022, “v”, “m@m.m”, “mdpppppp”)]
getMembre(“Rob”)	(“Rob”, “n”, “pr”, “l”, 01/01/2022, “v”, “m@m.m”, “mdpppppp”)
getMembre(“p2”)	null