Documentation technique du package « outils de controles »

Ce package possède une documentation propre car c'est un package indépendant. En effet, les fonctionnalités de ce package sont utiles pour beaucoup de programmes.

Le package « outils de controles » permet d'avoir des moyens de haut niveaux pour contrôler et vérifier que certaines conditions sont remplies, dans l'objectif d'éviter que votre programme ne lève pas des erreurs en cours de processus.
Par exemple, en contrôlant les arguments à l'appel de la méthode, votre programme ne commence pas à modifier l'objet alors qu'il risque de pas pouvoir finir les modifications attendues.

Ce package contient plusieurs classes qui vérifient que des objets sont valides :
● Verificateur qui vérifie qu'un objet non conteneur respecte certains critères (type, minimum, maximum, …)
● VerificateurStr qui vérifie qu'une chaîne respecte certains critères (longueur minimum, maximum, correspondance avec une expression régulière). Hérite de Verificateur.
● VerificateurConteneurs qui vérifie que les objets conteneurs sont composés d'éléments valides. (Pour chaque élément attendu, il est fourni un ou des identifiants ainsi qu'un objet Verificateur.)
● VerificateurListes qui vérifie qu'une liste est bien valide : on contrôle le nombre d'éléments de la liste et les éléments de la liste. Hérite de VerificateurConteneurs.
● VerificateurDict qui vérifie qu'un dictionnaire est bien conforme vis-à-vis de l'objet attendu. Hérite de VerificateurConteneurs. *(classe en développement)*
● VerificateurTableaux qui vérifie que des tableaux (c'est à dire une liste contenant d'autres listes, qui forme un tableau bi-dimensionnel) sont bien valides. Hérite de VerificateurListes.
● VerificateurArguments qui vérifie que les arguments fournis sont bien valides. Hérite de VerificateurConteneurs.

Ce package contient aussi un module avec des décorateurs. Parmi eux, on trouve le décorateur « controle_types » qui vérifie que les arguments passées à des fonctions ou méthodes sont du bon type. Un autre décorateur (« controle_arguments ») permet que les arguments vérifient tous les critères de validités définis par les objets vérificateurs.

Pour plus d'informations sur l'utilisation de ce package, voir le fichier « Guide d'utilisation du package outils de controles ».

Caractéristiques techniques de la version 2.5.0 :

Nom de la caractéristique :	Package	Fichiers de tests
Taille nécessaire à l'installation	248 Ko	380 Ko
Systèmes d'exploitation	Linux/Unix, Mac OS X et	Linux/Unix, Mac OS X et

compatibles (en théorie)	Windows 8 et 10	Windows 8 et 10
Systèmes d'exploitation compatibles (testés)	Windows 10 (64 bits)	Windows 10 (64 bits)
Langage de programmation	Python (version 3.9.2), interprété	Python (version 3.9.2), interprété

Remarques :
Pour que la version python fonctionne, il faut avoir téléchargé Python (version 3.9.2 ou ultérieure). Le package peut probablement fonctionner sous Python 3.8.6, mais rien n'est garanti.
Pour que les fichiers de tests s'exécutent, il faut avoir téléchargé le module unittest de Python (module de le bibliothèque standard).

Historique des versions :

N° de la version	Date de sortie	Modifications apportées
V 1.0.0	16/12/20	Version primitive adaptée d'un exemple de la PEP 318. « controle_type » : vérifie que les arguments passés en paramètre d'une fonction/méthode sont bien du type attendu. Un objet d'une classe qui hérite de la classe attendue est valide. (utilisation de la fonction « isinstance » et non « type »). Un argument peut être de différents types : mettre dans les paramètres du décorateur « (type1, type2) ».
V 1.1.1	17/12/20	Ajout de la documentation (ajout d'une docstring pour chaque fonction). Corrections de bugs et ajout de tests unitaires.
V 1.2.0	16/01/21	Ajout d'une nouvelle classe : ArgumentsAttendus. Cette classe permet de lister les arguments attendus, ainsi que les caractéristiques qu'ils doivent avoir. Ainsi, certaines méthodes de cette classe prennent en paramètre un argument reçu et vérifient la validité de l'argument. Le décorateur controle_types peut prendre en paramètre un objet ArgumentsAttendus pour effectuer les vérification de types.
V 2.0.0	25/04/21	Ajout de la classe Verificateur (pour le contrôle des objets non conteneur). En plus du contrôle des types, cette classe permet d'ajouter des fonctionnalités : minimum et maximum. Suppression de la classe ArgumentsArttendus. Elle est remplacée par une classe générique VerificateurConteneurs et une classe VerificateurArguments. Ajout d'une classe VerificateurListes (contrôle des listes et tuples).
V 2.1.0	09/05/21	Nouveau décorateur : @controle_arguments. Permet de vérifier que les arguments sont bien valides (ne vérifie pas uniquement les types). Ajout de fonctionnalités pour la classe VerificateurConteneurs et ses classes filles : contrôle du type du conteneur, et de sa longueur (minimum et maximum). Amélioration des tests unitaires (notamment pour VerificateurListes).
V 2.2.0	20/06/21	Ajout de la classe VerificateurTableaux. Permet de vérifier la validité d'objets contenus dans un tableau, c'est à dire une liste contenant d'autres listes, qui forme un tableau bi-dimensionnel. Vérifie le nom du tableau, les en-têtes et le contenu grâce aux vérificateurs définis pour les différentes colonnes.
V 2.2.1	24/06/21	Ajout de tests unitaires pour cette classe dans la version stable.

V 2.2.2	29/06/21	Modification des chemins d'importation.
V 2.2.3	03/07/21	Le dossier contenant les tests a été déplacé, d'où une nouvelle modification des chemins d'importation.
V 2.2.4	05/07/21	Modification mineures pour le déploiement du package.
V 2.2.5	11/07/21	Le minimum des classes héritant de VerificateurConteneurs ne peut plus être None. Si on ne veut pas contrôler le minimum, minimum doit être à 0 car les conteneurs sans éléments sont alors considérés comme valides.
V 2.3.0	12/07/21	Ajout de la classe VerificateurStr pour contrôler les chaînes de caractères.
V 2.3.1	15/07/21	Modification de la valeur par défaut de l'attribut types pour la classe VerificateurArguments. Les types valides par défaut pour cette classe sont les listes, tuples et dictionnaires.
V 2.3.2	16/07/21	Modification de la valeur par défaut de l'attribut types pour la classe VerificateurListes. Le type valide par défaut pour cette classe est le type list.
V 2.4.0	16/07/21	Si l'objet contrôlé n'est pas d'un type valide, il peut être convertit dans un type valide. Pour cela, le paramètre conversion de la méthode de contrôle doit être vrai. Ajout de la méthode conversion qui convertissent un objet vers un type valide. N.B. : Si la conversion vers un type valide n'est pas possible un erreur est levée.
V 2.4.1	17/07/21	Amélioration de la conversion des tuples dans les classes VerificateurListes et VerificateurArguments. (On les transforme en listes avant la conversion des objets contenus puis on les retransforme en tuple.)
V 2.4.2	18/07/21	Les méthodes « conversion » ne doivent plus être utilisé directement par les utilisateurs. Ces méthodes doivent uniquement être utilisé par les méthodes des classes du package.
V 2.4.3	19/07/21	Correction du bug liés aux fonctionnalités de conversion pour la classe VerificateurArguments. Ajout de tests unitaires liés à ces fonctionnalités.
V 2.4.4	19/07/19	Correction de bugs liés à la fonctionnalité conversion dans la classe VerificateurTableaux.
V 2.4.5	19/07/19	Correction d'un bug concernant le décorateur controle_types. Ajout de tests unitaires liés à ces fonctionnalités.
V 2.4.6 à V 2.4.8	21/07/21 à 22/07/21	Modification du fichier setup.py pour ajouter des mots-clés dans PyPI et améliorer l'affichage du fichier README.md. Modification de ce fichier avec sa traduction en anglais et formatage Markdown.
V 2.4.9	24/07/21	L'attribut regex de la classe VerificateurStr est compilé pour que le contrôle des chaînes soit plus rapide. On peut fournir à l'attribut une regex déjà compilée.
V2.5.0	12/08/21	Ajout d'un décorateur pour contrôler le temps d'exécution d'une fonction/méthode.
V2.6.0	Planifiée	Ajout de la classe VerificateurDict pour contrôler les dictionnaires. Rajout de tests unitaires pour cette classe.

Signalement de bugs :

- Aucun pour le moment.

Pour signaler un bug, merci de faire un message à l'adresse c.b.e.python@gmail.com. Pour un meilleur traitement de votre message, nous vous conseillons de mettre comme objet « Signalement de bug - outils de controles » et de décrire le plus précisément possible le problème dans votre message. Pour nous aider à identifier le problème, veuillez, si possible, indiquer le texte d'erreur généré par python. De plus, nous vous remercions de préciser : le numéro de version du package, le numéro de version de l'interpréteur de Python que vous utilisez, votre système d'exploitation, le fichier qui lève l'erreur. Merci de votre coopération.

Propositions d'amélioration :

● Améliorations proposées pour la prochaine version 2.6.
● Pour les classes héritant de VerificateurConteneurs, rajouter des méthodes pour pouvoir modifier ou supprimer les listes de vérificateurs des objets contenus.

Crédits :

Développeur : Cyprien BONTRON

Contact : c.b.e.python@gmail.com

Pour tous renseignements ou problème de fonctionnement, faites un mail à l'adresse sus citées. Nous ne sommes pas tenu de répondre.

Remerciements : Python software foundation

Documentation de la version : 2.5.0

Mise à jour : Août 2021

Référence de la documentation : ODC0721It (Documentation pour Outils De Controle de août (08) 2021, version A, concernant les caractéristiques techniques du package)