

[Tableau de bord](#) / [Mes cours](#) / [LOG3430 - Méthodes de test et de validation du logiciel](#) / C05 - Automatisation des tests
/ [Activité liée à la vidéo sur l'automatisation des tests](#)

Commencé le vendredi 30 septembre 2022, 16:45

État Terminé

Terminé le vendredi 30 septembre 2022, 16:48

Temps mis 3 min 16 s

Note Pas encore évalué

Question 1

Correct

Note de 0,20
sur 0,20

Associez le type de test à son niveau d'automatisabilité correspondant.

Test pyTest	toujours automatisé	✓
Test de performance	automatisable	✓
Test JUnit	toujours automatisé	✓
Test en boîte noire	automatisable	✓
Test en boîte blanche	automatisable	✓
Test de mutation	automatisable	✓

Votre réponse est correcte.

La réponse correcte est :

Test pyTest → toujours automatisé,

Test de performance → automatisable,

Test JUnit → toujours automatisé,

Test en boîte noire → automatisable,

Test en boîte blanche → automatisable,

Test de mutation → automatisable

Question 2

Correct

Note de 0,20
sur 0,20

Lequel des éléments suivants est un inconvénient de l'automatisation des tests ?

- ☐ a. Les tests automatisés sont une perte d'argent
- ☒ b. Les tests automatisés peuvent être difficiles à créer ✓ Effectivement, certains résultats de test sont difficile à vérifiés automatiquement
- ☐ c. Les tests automatisés prennent plus de temps que les tests manuels
- ☐ d. Les tests automatisés sont généralement sujets aux erreurs.

Votre réponse est correcte.

La réponse correcte est :

Les tests automatisés peuvent être difficiles à créer

Question **3**

Terminé

Noté sur 0,60

En utilisant le code que nous avons vu pendant la classe 3 (voir ci-dessous, des getters sont ajoutés), créez deux tests unitaires.

```
class CCoinBox:
    monnaie_totale = 0
    monnaie_courante = 0
    vente_permise = False

    def __init__(self):
        self.reset()

    def ajouter_25c(self):
        self.monnaie_courante = self.monnaie_courante + 1
        if self.monnaie_courante > 1:
            self.vente_permise = True
            print("Une pièce a été ajoutée")

    def vente(self):
        if self.vente_permise:
            self.monnaie_totale = self.monnaie_totale + 2
            self.monnaie_courante = self.monnaie_courante - 2
            print("Vente! Voici votre article ...")
            if self.monnaie_courante < 2:
                self.vente_permise = False

    def reset(self):
        self.monnaie_totale = 0
        self.monnaie_courante = 0
        self.vente_permise = False
        print("Machine reset")

    def retourne_monnaie(self):
        self.monnaie_courante = 0
        self.vente_permise = False
        print("Voici votre monnaie")

    def get_monnaie_totale(self):
        return self.monnaie_totale

    def get_monnaie_courante(self):
        return self.monnaie_courante

    def get_vente_permise(self):
        return self.vente_permise
```

Nous continuerons cette question en classe.

*notez que des points seront attribués à cette question en fonction de l'effort des tentatives (si vous avez essayer ou non) et non de l'exactitude, mais vous devez être familier avec le code en classe.

```
import unittest
```

```
class CCoinBoxTest:
```

```
    def setUp(self):
```

```
        self.c1=CCoinBox()
```

```
    def test_retourne_monnaie(self):
```

```
        self.c1.retourne_monnaie()
```

```
        self.assertEqual(self.c1.monnaie_courante,0)
        self.assertFalse(self.c1.vente_permise)
def test_reset(self):
    self.c1.reset()
    self.assertEqual(self.c1.monnaie_totale,0)
    self.assertEqual(self.c1.monnaie_courante,0)
    self.assertFalse(self.c1.vente_permise)
def tearDown(self):
    self.c1.dispose()
```

◀ Vidéo C5 - Automatisation des tests

Aller à...

Diapos de la vidéo ▶