

MGL7460 — Réalisation et maintenance de logiciels
Examen final (Automne 2016)

Durée: 3 heures **Documentation autorisée:** Toute documentation personnelle.

Nom: _____

Code permanent:

1	2	3	4	5	6	Total
/5	/10	/10	/5	/10	/10	/50

-
- Vous devez répondre directement sur le questionnaire.
 - Tant la clarté, la justesse que la pertinence des explications seront évaluées — donc si vous écrivez des choses *fausses* ou «*qui n'ont pas rapport*», vous pourrez être pénalisés.
 - Pour certaines questions, vous aurez à écrire du code. Sauf pour certaines questions spécifiques — questions 1, 3a (2^e partie) et 4 — vous pouvez utiliser le langage de votre choix, y compris du pseudocode.
 - Si vous utilisez un autre langage que Ruby, **vous n'avez pas** à redéfinir les classes et méthodes fournies (celles qui vous ont été transmises par courriel) ; vous pouvez simplement supposer que des classes avec des attributs et méthodes équivalents sont disponibles et déjà définies.
-

1. Ruby et MiniTest (5 pts)

Complétez les cas de test suivants, en MiniTest, pour qu'ils s'exécutent avec succès.

```
it "test 1" do
  [0, 1, 2].map { |x| x + 1 }
  .reduce(0) { |t, x| t + 2*x }
  .must_equal                                     #?
end
```

```
it "test 2" do
  def foo( x )
    yield( x > 10 )
  end
```

```
  r = []
  foo(10) { |x| r << x }
  foo(20) { |x| r << x }

  r.must_equal                                   #?
end
```

```
it "test 3" do
  x = "x"
  x.stub :+, "" do
    (x + "AB").must_equal                       #?
    ("AB" + x).must_equal                       #?
  end
  (x + "AB").must_equal                         #?
end
```

```
it "test 4" do
  m = MiniTest::Mock.new
  m.expect( :foo, "foo", [Fixnum, String] )
  m.expect( :bar, "bar", ["foo"] )
```

#?

```
  m.verify
end
```

2. Thèmes variés (10 pts)

Répondez à trois (3) questions parmi les suivantes. Vous pouvez aussi répondre à une quatrième (4^e) question et possiblement obtenir **des points bonus**.

Indiquez clairement à quelle sous-question vous répondez en encerclant l'item approprié au début des espaces pour les réponses.

[3.5] a) Décrivez de quelle façon et avec quel genre d'outils on peut exécuter, de façon automatique, des tests d'acceptation pour une application avec une interface Web.

[3.5] b) Pour certains auteurs, la maintenance est une phase distincte de la phase de développement. Par contre, d'autres considèrent que ce n'est pas le cas. Discutez brièvement des deux points de vue.

[3.5] c) Comment peut-on mesurer la complexité d'un composant logiciel et quels sont les liens entre complexité et tests?

[3.5] d) Selon Martin Feathers, du **code légataire** (*legacy code*) est «*simply code without tests*». Qu'est-ce que Feathers veut dire par cette affirmation?

[3.5] e) Lorsqu'on traite de qualité des logiciels, on peut distinguer entre «qualité interne» et «qualité externe» :¹ «*External quality mean all the properties of the software as a product that users can experience and enjoy [;] Internal quality mean all the properties of the software as seen by the developers.*»

Expliquez ce qu'est la «dette technique» et indiquez à quel aspect de la qualité — interne ou externe — cette notion est liée.

¹<http://wiki.c2.com/?InternalAndExternalQuality>

Sous-question a b c d e

Sous-question a b c d e

Sous-question a b c d e

Sous-question a b c d e

3. Tests unitaires (10 pts)

Cette question porte sur les classes présentées dans l'annexe **A Contrôleurs pour une machine à café simple**. On veut définir des tests unitaires pour la méthode `traiter_piece`, la méthode centrale de la classe `ControleurMachineCafe`.

- a. De quelle façon doit-on procéder pour tester cette méthode? Si des modifications doivent être apportées à une/des classe/s, expliquez brièvement pourquoi et indiquez clairement les modifications requises en fournissant le code modifié.

- b. Écrivez un cas de test dans lequel un usager insère une pièce de 1.00 \$ alors que le café coûte 0.50 \$. Assurez-vous de vérifier que tous les messages appropriés sont transmis **aux autres contrôleurs**.

(**Bonus**) `traiter_piece` contient actuellement un *bogue*, lié à `@montant_recu` 😊

- a. Écrivez un cas de test qui montre la présence de ce bogue — donc un cas de test qui, dans l'état actuel du code, **va échouer!**
- b. Indiquez quelle(s) modification(s) il faut faire à `traiter_piece` pour corriger le bogue et assurer que votre cas de test s'exécute avec succès.

Note : Si vous répondez à cette question bonus, répondez **au verso** de cette page!

4. Utilisation d'APIs coulantes (5 pts)

Cette question porte sur les classes de l'annexe **B Modèle et classes pour des cours et programmes**. On veut construire des objets en utilisant les APIs coulantes appropriées, en s'assurant que les objets résultants sont correctement construits.

a. Définissez les deux cours suivants :

- (i) Un cours `mgl7777` de 3 crédits dont le sigle est "MGL7777", le titre est "Developpement" et qui n'a aucun préalable.
- (ii) Un cours `mgl8888` de 3 crédits dont le sigle est "MGL8888", le titre est "Realisation" et qui a un seul préalable, soit le cours `mgl7777`.

b. Définissez un programme d'études `micro_prog_gl` dont le code est 1010, le nom est "Micro-programme GL" et qui est composé des deux cours `mgl7777` et `mgl8888` définis précédemment (donc pour un total de 6 crédits).

5. Tests «unitaires» avec un outil de BDD (10 pts)

Cette question porte sur les classes de l'annexe **B Modèle et classes pour des cours et programmes**. On veut définir des tests «unitaires» pour la classe `Programme`, mais en utilisant un outil de BDD (sic) — on suppose que `Cours` est déjà testé.

- a. Donnez la spécification d'un scénario — exprimé en `gherkin` ou une notation semblable — qui crée le "Micro-programme GL" décrit à la question précédente et qui vérifie que ce programme est correctement créé et défini — avec le bon code, le bon nom, le bon nombre de crédits, les bons cours.

Feature: Creation d'un nouveau programme

En tant que responsable academique
Je veux pouvoir definir un nouveau programme d'etudes avec divers cours
De facon a ce que le programme puisse etre offert aux etudiants

Scenario: Je cree un micro-programme en GL qui comporte deux cours

- b. Donnez la définition des *steps* pour un (1) **Given** + un (1) **When ou Then**² — avec les *signatures* et *mises en oeuvre* des méthodes appropriées — qui vont rendre exécutable le scénario décrit en a. à l'aide d'un outil de BDD.

Vous pouvez utiliser le langage/outil de votre choix (cf. p. 1), y compris pour les assertions.

²Avec des points **bonus** possibles si vous donnez un **When et** un **Then!**

6. Pratiques pour une équipe de développement (10 pts)

Vous travaillez actuellement pour une firme de consultants. Pour votre prochaine affectation, vous allez être responsable d'une petite équipe de développeurs (5–6 personnes), laquelle va réaliser une application Web de commerce en ligne pour la compagnie *Foo Inc.* Une première version, minimale, de l'application devra être livrée dans quelques mois, puis des fonctionnalités additionnelles seront ajoutées, de façon incrémentale, dans les mois qui suivent.

Les personnes qui feront partie de cette équipe, des personnes déjà à l'emploi de *Foo Inc.*, ont travaillé ensemble auparavant, sur un projet d'un type différent, mais utilisant des technologies Web. Ces personnes ont déjà l'habitude d'utiliser un système de contrôle du code source (style `git`) et de développer des tests unitaires.

Par contre, lors du dernier projet, lorsqu'est venu le temps de livrer et déployer l'application, l'équipe a rencontré de nombreuses difficultés, qui ont retardé la livraison du produit final. De plus, la maintenance — surtout corrective et perfective — fut difficile, la qualité du code n'étant pas très bonne. Pour éviter de rencontrer les mêmes problèmes, ces personnes sont toutes disposées à essayer de nouvelles pratiques et de nouveaux outils.³

Quels seraient les principaux outils et pratiques que vous leur proposeriez de mettre en place pour assurer que le logiciel soit livré dans les délais prévus et pour assurer que le logiciel soit maintenable?

Vous devez décrire (brièvement) comment vous expliqueriez et justifieriez ces choix aux membres de l'équipe — donc en quoi consiste ces outils/pratiques et pourquoi vous les suggérez.

³Malheureusement, ce n'est pas ainsi dans toutes les organisations ☹

