



POLYTECHNIQUE  
MONTRÉAL

## Questionnaire Contrôle Périodique 4

**LOG3430**

Sigle du cours

Identification de l'étudiant(e)		
Nom :	Prénom :	
Signature :	Matricule :	Groupe :

Sigle et titre du cours		Groupe	Trimestre
LOG3430 - Méthodes de test et de validation du logiciel		Tous	20163
Professeur		Local	Téléphone
Giuliano Antoniol		L-2204	
Jour	Date	Durée	Heures
Mercredi	2 Novembre 2016	1 heure	
Documentation		Calculatrice	
<input type="checkbox"/> Aucune <input checked="" type="checkbox"/> Toute <input checked="" type="checkbox"/> Voir directives particulières		<input type="checkbox"/> Aucune <input checked="" type="checkbox"/> Toutes <input type="checkbox"/> Non programmable	Les cellulaires, agendas électroniques ou téléavertisseurs sont interdits.
Directives particulières			
Toute documentation est permise, ainsi que les calculatrices, tablettes et ordinateurs à l'exception toutefois de tout dispositif connecté à Internet.			

<b>Important</b>	Cet examen contient <input type="text" value="2"/> exercices et <input type="text" value="9"/> questions sur un total de <input type="text" value="10"/> pages (excluant cette page)
	La pondération de cet examen est de <input type="text" value="5"/> %
	Vous devez répondre sur : <input checked="" type="checkbox"/> le questionnaire <input type="checkbox"/> le cahier <input type="checkbox"/> les deux
	Vous devez remettre le questionnaire : <input checked="" type="checkbox"/> oui <input type="checkbox"/> non

L'étudiant doit honorer l'engagement pris lors de la signature du code de conduite.

## Exercice 1 – 8 points

Considérez le code suivant :

```
class Book{
    String title;
    String author;
    double price;
    protected int pages;

    Book(){} // 1
    Book(String title, String author, double newPrice, int numpages){ //2
        title = title;
        author = author;
        price = newPrice;
        pages=numpages;
    }
    public String getTitle(){ return title;} //3
    public String getAuthor(){return author;} //4
    public double getPrice(){ return price; } //5
    public int getPages(){ return pages;} //6
    public void setPrice(double newPrice){price = newPrice; } //7
}

class UsedBook extends Book{
    int status ;
    int year ;
    public int getStatus(){ return status;} //8
    public void setStatus(int n){ status=n;} //9
    public int getYear(){return year;} //10
    public void setYear(int n){ year=n;} // 11
    public void addUsedCopy(String t, String a, int np, int s, int y){ //12
        title=t;
        author=a;
        pages=np;
        status=s;
    }
    public void changePrice(double p){setPrice(p);} //13
    public void giftBook(){price=0.0;} //14
    public void showStatus(){ // 15
        if (status == 0)
            System.out.println("Like new\n");
        else if (status == 1)
            System.out.println("Good condition\n");
        else if (status == 2)
            System.out.println("Used\n");
        else
            System.out.println("Damaged copy\n");
    }
}
```

Pour les classes Book et UsedBook il faut concevoir les test unitaire.

1.1) (2 point) Selon le critère de Harrold McGregor quelles sont les méthodes du Book qui doivent être testées de nouveaux ?

Answer :

All methods must be re-tested

- 1.1.1 Book(), Book(...)
- 1.1.2 getTitle
- 1.1.3 getAuthor
- 1.1.4 getPrice
- 1.1.5 getPages
- 1.1.6 setPrice

Justification : addUsedCopy changes title, author and number of pages, plus we can assume there was an error and the programmer forgot to set year to the parameter y. Notice that changePrice and giftBook also alter price.

1.2 (1 point) Si on utilise le critère de Harrold McGregor quelles sont les méthodes du UsedBook qui doivent être testées?

Answer :

Here Harrold McGregor has actually nothing to do ; this is the first time we test UsedBook and thus all methods must be tested.

1.3 (1 point) Si on veut tester les classes avec la méthode des tranches des données, spécifiez dans le tableau suivant combien de setter et getter il faut ajouter pour chaque class :

Class	Setter	Getter
Book	setTitle, setPages, setAuthor → 3	0
UsedBook	0	0

1.4 (1 point) Pour la class Book identifiez les tranches des donnees, ajouter des getter si nécessaire

Attribute									
<b>title</b>	<b>Book()</b>	<b>Book(...)</b>	<b>setTitle</b>	<b>getTitle</b>					
<b>author</b>	<b>Book()</b>	<b>Book(...)</b>	<b>setAuthor</b>	<b>getAuthor</b>					
<b>price</b>	<b>Book()</b>	<b>Book(...)</b>	<b>setPrice</b>	<b>getPrice</b>					
<b>pages</b>	<b>Book()</b>	<b>Book(...)</b>	<b>setPages</b>	<b>getPages</b>					

1.5 (1 point) Pour la class UsedBook identifiez les « tranches des donnees », ajouter des getter si nécessaire

Attribute									
<b>status</b>	<b>setStatus</b>	<b>getStatus</b>	<b>showStatus</b>	<b>addUsedCopy</b>					
<b>year</b>	<b>setYear</b>	<b>getYear</b>	<b>addUsedCopy</b>						

1.6 (2 points) Pour la class UsedBook donnez le MADUM dans le tableau suivant (usez les numéros !):

MADUM

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>Title</b>	C	C	r									t			
<b>author</b>	C	C		r								t			
<b>price</b>	C	C			r		t						t	t	
<b>pages</b>	C	C				r						t			
<b>status</b>								r	t			t			o
<b>year</b>										r	t	t!			

Selon le MADUM quelles sont les méthodes du Book qui doivent être testées de nouveau ?

Answer :

We notice that methods 12, 13 and 14 basically transform all Book attribute thus again we have to retest all the tranches and thus all methods.

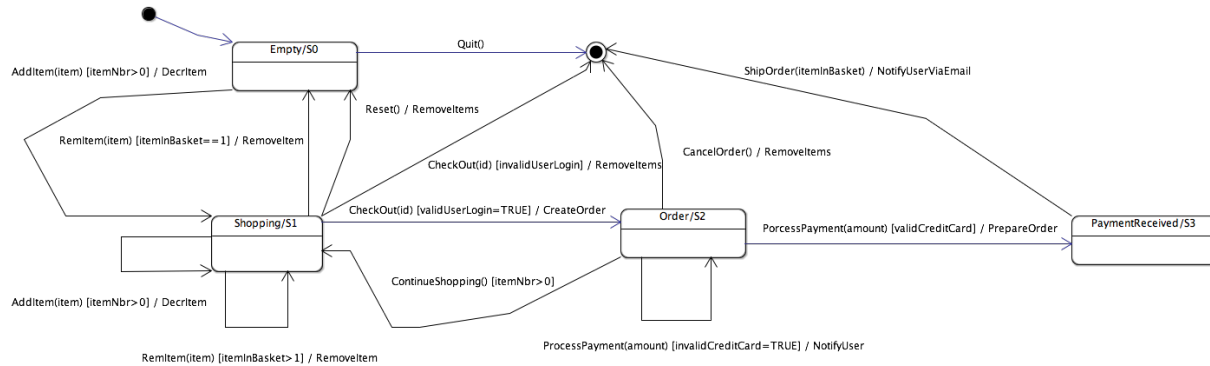
Est-ce que les réponses à la test selon le MADUM et le critère Harrold-McGregor sont en désaccord ? Justifie et explique la réponse.

Answer :

As one can expect the answer is the same all need to be retested.

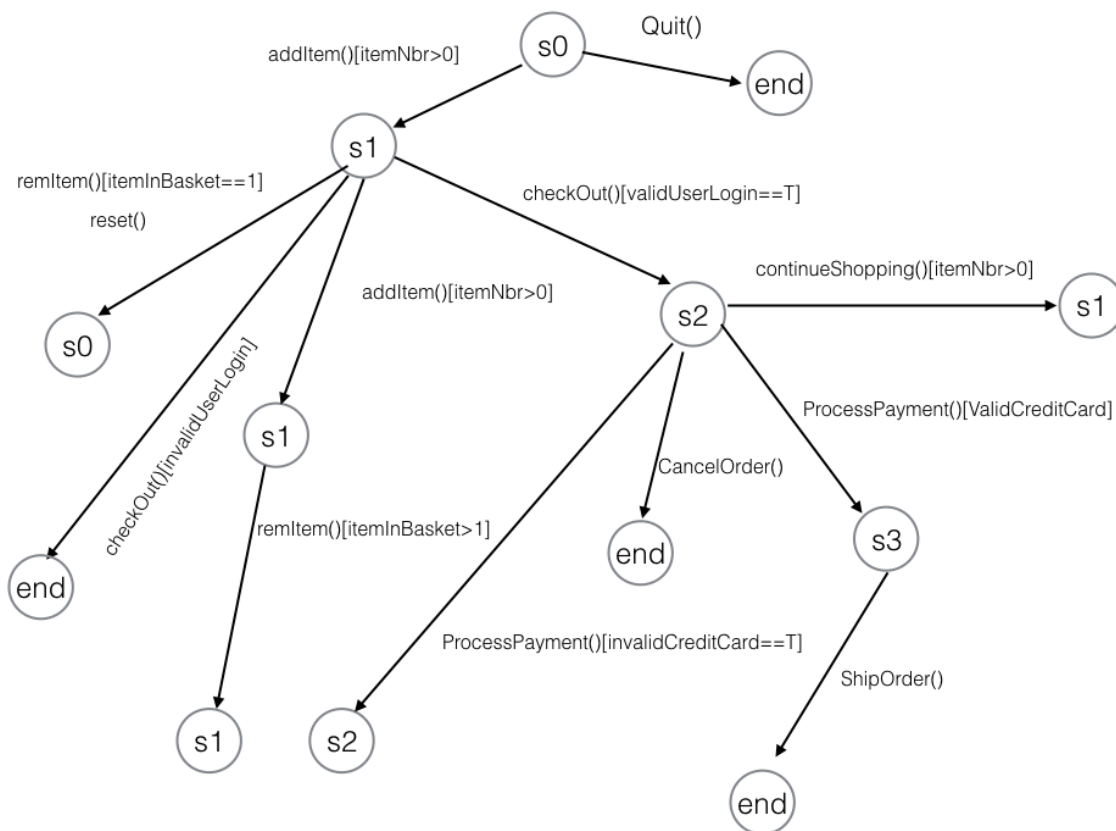
## Exercice 2 - 12 points

Pour le diagramme suivant :



**Q2.1)** Dérivez l'arbre de transition selon la convention S0=Empty, S1=Shopping, etc. (4 points)

Answer :



**Q2.2)** Dérivez les séquences de test. Pour chaque cas de test, il faut justifier et expliquer si des conditions spéciales sont nécessaires. (4 points)

Answer :

S0 Quit () END

S0 addItem [ItemNBR>0] S1 reset S0

S0 addItem [ItemNBR>0] S1 remItem [itemInBasket==1] S0

S0 addItem [ItemNBR>0] S1 checkOut [invalidUserLogin] END

S0 addItem [ItemNBR>0] S1 addItem S1

S0 addItem [ItemNBR>0] S1 addItem [ItemNBR>0] S1 remItem [itemInBasket>1] S1

S0 addItem [ItemNBR>0] S1 checkOut[validUserLogin==T] S2

ProcessPayment()[invalidCreditCard==T] S2

S0 addItem [ItemNBR>0] S1 checkOut[validUserLogin==T] S2 CancelOrder END

S0 addItem [ItemNBR>0] S1 checkOut[validUserLogin==T] S2 ProcessPayment()[validCreditCard==T]

S3 ShipOrder END

S0 addItem [ItemNBR>0] S1 checkOut[validUserLogin==T] S2 ContinueShopping S1

Notice only the essential guard conditions are reported. It is also important to notice that the 5<sup>th</sup> test case can be written as

S0 addItem [ItemNBR>0] S1 ( addItem [ItemNBR>0] S1)+ remItem [itemInBasket>1] S1

Where we add a new item at least once. The above sequences can be combined to obtain a lower number of test cases. For example

S0 addItem [ItemNBR>0] S1 ( addItem [ItemNBR>0] S1)+ remItem [itemInBasket>1]

S1vcheckOut[validUserLogin==T] S2 ContinueShopping S1 checkOut[validUserLogin==T] S2

ProcessPayment()[validCreditCard==T] S3 ShipOrder END

Joins together 3 simple sequences into a longer one.

**Q2.3)** Pour chaque cas de test, dérivez l'oracle : les séquences des états et les conditions requises pour traverser le chemin. Il n'est pas requis de fournir les valeurs des paramètres des méthodes. (4 point).

Answer :

Nothing to be done the oracle is already specified within the test sequences as the oracle is the status.