Université Abou Beckr Belkaid
Master 2 Génie Logiciel
Année Universitaire 2019/2020

Examen Final

—

Validation et vérification logiciel

—

Remarques :

— Les documents ne sont pas autorisés ainsi que les appareils électroniques (PC, Tablette, téléphone, etc).
— La clarté des réponses et du code est très importante.

Partie A Questions de cours :

1 donner la définition des testes en boites noires.
2 Donner la définition du graphe de flot de données. Comment il est utilisé?

Partie B

Exercice 1 :

Un programme est donné comme suite :

```
begin
if (x<=0)then x:=-x
else x:=x-1;
if (x=-1)then x:=1
else x:=x+1;
end
```

Questions :

1 Donner le graphe de contrôle correspondant a ce programme.
2 Donner un jeux de testes permettant d'obtenir le critère "toutes les instructions".
3 Donner un jeux de testes permettant d'obtenir le critère "toutes les branches".
4 Est ce que les critères de couverture choisis sont pertinent? pourquoi?

Exercice 2 :

Dans cet exercice, nous souhaitons définir des jeux de testes en utilisons le critère de couverture toutes les conditions-décisions modifiées (MC/DC). Pour cela nous proposons le bout de code suivant :

```
if(( ((u == 0) ou (x>5)) && ((y<6) || (z == 0)) )){
//instructions
}else{
//instruction
}
```

Questions :

1 Donner le définition du critère de couverture MC/DC.
2 Utiliser la table de vérité pour définir des jeux de testes qui vérifie la couverture MC/DC.

Exercice 2 :

Une application de gestion de stock permet de définir un portefeuille (Portfolio) pour des clients. Le portefeuille du client contient la liste de son stock. Cette application utilise deux classes **Stock** et **Portfolio**. Ainsi que l'interface **StockService**.

```java
public class Stock {
   private String stockId;
   private String name;
   private int quantity;

   public Stock(String stockId, String name, int quantity){
      this.stockId = stockId;
      this.name = name;
      this.quantity = quantity;
   }

   public String getStockId() {
      return stockId;
   }

   public void setStockId(String stockId) {
      this.stockId = stockId;
   }

   public int getQuantity() {
      return quantity;
   }

   public String getTicker() {
      return name;
   }
}
```

.

```java
public class Portfolio {
   private StockService stockService;
   private List<Stock> stocks;

   public StockService getStockService() {
      return stockService;
   }

   public void setStockService(StockService stockService) {
      this.stockService = stockService;
   }

   public List<Stock> getStocks() {
      return stocks;
   }

   public void setStocks(List<Stock> stocks) {
      this.stocks = stocks;
   }

   public double getMarketValue(){
      double marketValue = 0.0;

      for(Stock stock:stocks){
         marketValue += stockService.getPrice(stock) * stock.getQuantity();
      }
      return marketValue;
   }
}
```

.

```java
public interface StockService {
   public double getPrice(Stock stock);
}
```

.

Questions :

1 Donner le diagramme de séquence lié à la méthode getMarketValue().

2 Nous souhaitons tester la classe **Portfolio**.

2.1 Quelles techniques utilisées pour faire les testes ? Justifier votre réponse.

2.2 Donner l'implémentation des testes de la classe **Portfolio**.

Bon courage et bonne continuation.