



# TP1

---

## Exercice 1

Nous sommes dans le cadre d'un projet Java qui gère un système de bonification pour des clients.

Chaque client dispose d'un compte, soit les classes Java Client et Compte ci-dessous illustrant le cas :

```
package domain.test.org;

public class Client {
    private String id;
    private String nom;
    private String prenom;
    private String adresse;
    private String tel;

    public Client(String id, String nom, String prenom, String adresse, String tel) {
        super();
        this.id = id;
        this.nom = nom;
        this.prenom = prenom;
        this.adresse = adresse;
        this.tel = tel;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getNom() {
        return nom;
    }
```

```
public void setNom(String nom) {
    this.nom = nom;
}

public String getPrenom() {
    return prenom;
}

public void setPrenom(String prenom) {
    this.prenom = prenom;
}

public String getAdresse() {
    return adresse;
}

public void setAdresse(String adresse) {
    this.adresse = adresse; }

public String getTel() {
    return tel;
}

public void setTel(String tel) {
    this.tel = tel;
}

package domain.test.org;

public class Compte {
    private String idClient;
    private int totalPoint;
    public Compte(String idClient, int totalPoint) {
        super();
        this.idClient = idClient;
        this.totalPoint = totalPoint;
    }

    public int addPoints (float montantVente)
```

```

{
int bonus=0;
if ((montantVente >= 10f) && (montantVente < 100f))
bonus = (int)(montantVente / 10);
else if ((montantVente >= 100f) && (montantVente < 1000f))
bonus = (int) montantVente / 15;
return bonus;
}

public boolean usePoints (float montantAchat)
{
if (this.totalPoint > 0){
this.totalPoint -= (montantAchat * 5);
return true;
}
else
return false;
}

public String checkCadeauType ()
if ((this.totalPoint >= 50) && (this.totalPoint < 100))
return "Platinum";
else if ((this.totalPoint >= 100) && (this.totalPoint < 200))
return "Silver";
else
return "Gold";
}

public int getTotalPoint() {
return totalPoint;
}

public void setTotalPoint(int totalPoint) {
this.totalPoint = totalPoint;
}

```

}}

### Question 1 :

Nous voulons procéder au test unitaire de la classe Compte, nous utiliserons le *framework Junit* pour cela.

- a. Identifier les cas de test à préparer pour chaque méthode de la classe Compte en indiquant dans chaque cas : les input et les résultats prévus.
- b. Créer la classe TestCompte de type JUnit Test Case.

Créer les tests pour les méthodes :

- **addPoints**
- **usePoints**
- **checkCadeauType**

#### Remarque :

**Utiliser assertEquals (expect, actual) : réaliser le test.**

**@Test:** indiquer qu'il s'agit d'un test.

### Question 2 :

Pour manipuler l'ensemble des clients, nous utiliserons une nouvelle classe :

Clients (avec s) dont ci-dessous l'implémentation.

```
public class Clients {  
    ArrayList<Client> Clients;  
  
    public Clients() {  
        super();  
        Client client;  
        Clients = new ArrayList<Client>() ;  
        client = new Client ("1", "Ben Foulén", "Foulén", "Rue des roses", "1111111111" ) ;  
        Clients.add(client );  
        client = new Client ("2", "Ben Foulén", "Foulén", "Rue des roses", "22222222" ) ;  
        Clients.add(client );  
        client = new Client ("3", "Ben Foulén", "Foulén", "Rue des roses", "33333333");  
        Clients.add(client );  
        public void add (Client client)  
        {
```

```

        this.Clients.add(client) ;
    }
    public boolean remove (Client client)
    {
        int i = Clients.indexOf (client);
        if (i != -1)
        {
            Clients.remove(i);
            return true;
        }
        else
            return false;
    }
    public Client getElementById (int i)
    {
        return this.Clients.get(i);
    }
    public Client getElementByName (String name)
    {
        int i=0;
        while (!this.getElementById(i).getNom().equals (name) )
        {
            i++; }
        if (i >= this.getSize()-1)
            return null;
        else
            return this.getElementById(i) ;
    }
    public ArrayList<Client> getAll ()
    {

```

```
return this.Clients;  
}  
public int getSize ()  
{  
return this.Clients.size(); }  
public void deleteAll ()  
{  
this.Clients.clear();  
}}
```

Créer une nouvelle classe TestClients permettant de tester toutes les méthodes de la classe Clients.