

Exercice 2 – TD 3 – Correction :

Ecrire un service web ShopCatalogWebService fournissant des opérations pour la gestion de produits d'un magasin. Une opération pour ajouter un produit, une autre pour récupérer la liste complète des produits et enfin une dernière opération pour récupérer un produit par un identifiant. Un produit est décrit par un identifiant (String), une catégorie (String), et un prix.

- 1 Ecrire les classes de ce service web, à savoir, Product, ShopCatalog (interface), ShopCatalogImpl, et ShopCatalogPublisher.
- 2 Changer depuis l'interface le nom du service web en CatalogMagasin.
- 3 Changer depuis l'interface le nom de l'opération 2 en *tousLesProduits*.
- 4 Changer depuis l'interface le nom du paramètre de l'opération 3 en *produitID* et le nom du résultat retourné en *produitRetourné*.
- 5 Pour tester ce service web, créer une classe cliente qui le consomme en appelant l'opération 2.

Interface :

```
@WebService(name="CatalogMagasin")
public interface ShopCatalog {
```

```
    @WebMethod
    public boolean addProduct(String id, String categ, double prix);
```

```
    @WebMethod(operationName="tousLesProduits")
    public ArrayList<Product> getProducts();
```

```
    @WebMethod
    @WebResult(name="produitRetourné")
    public Product getProductById(@WebParam(name="produitID") String id);
```

```
}
```

Classe Impl :

```
@WebService(endpointInterface = "org.soa.ws.tp3.ShopCatalog")
public class ShopCatalogImpl implements ShopCatalog{
```

```
    private ArrayList<Product> products;
```

```
    public ShopCatalogImpl() {
        products = new ArrayList<Product>();
        products.add(new Product("prodB1","Livre", 100));
        products.add(new Product("prodM2","Magazine", 200));
        products.add(new Product("prodM3","CD Music", 300));
    }
```

```
    @Override
    @WebMethod
    public boolean addProduct(String id, String categ, double prix) {
        Product p = new Product(id, categ, prix);
        return products.add(p);
    }
```

```
    @Override
    @WebMethod
```

```

public ArrayList<Product> getProducts() {
    return products;
}

@Override
@WebMethod
public Product getProductById(String id) {
    for(int i=0; i<products.size();i++)
        if(id.equals(products.get(i).getId()))
            return products.get(i);
    return null;
}
}

```

Classe Product:

```

public class Product {

    private String id;
    private String categ;
    private double prix;

    public Product(String id, String categ, double prix) {
        this.id = id;
        this.categ = categ;
        this.prix = prix;
    }

    // Getters et setters

}

```

Classe Client :

```

public class ShopCatalogWSClient {

    public static void main(String[] args) {

        ShopCatalogImplService service = new ShopCatalogImplService();
        CatalogMagasin catalog = service.getShopCatalogImplPort();

        List<Product> products = new ArrayList<Product>();
        products = catalog.tousLesProduits();
        for(int i=0; i<products.size(); i++)
            System.out.println("id = " + products.get(i).getId() + " Categ = "
                               + products.get(i).getCateg());
    }
}

```