

1. Metoda „ProcessProducts(ProductsController)“

Za metodu ProcessProducts koja se nalazi u ProductsController-u čija je „Cyclomatic Complexity“ 5 izvršavamo white box testiranje. White box testiranje je provedeno kroz sve pristupe i za svaki pristup je napisana dokumentacija i izvještaj.

1.1 Obuhvat iskaza/linija (*Statement/Line coverage*)

Za obuhvaćenost svih iskaza prilikom testiranja bila su potrebna dva testna slučaja i to:

Testni slučajevi:

```
1.    var listProducts = new List<Product>
{
    new Product
    {
        ProductID = 1,
        Name = "Product 1",
        ImageUrl = "url1",
        Price = 60,
        FlowerType = "Rose",
        Stock = 10,
        Category = "Flowers",
        Description = "Beautiful red rose",
        productType = "Type A"
    },
    new Product
    {
        ProductID = 2,
        Name = "Product 2",
        ImageUrl = "url2",
        Price = 20,
        FlowerType = "Lily",
        Stock = 15,
        Category = "Flowers",
        Description = "Elegant white lily",
        productType = "Type B"
    }
};
```

```
2.    var listProducts = null;
```

Nakon testiranja sa prethodno navedena dva testna slučaja postignuta je poptuna obuhvatnost iskaza/linija. To je takođe i najmanji broj testnih slučajeva da se to postigne.

1.2 Obuhvat grana/odluka (*Branch/Decision Line coverage*)

Za obuhvaćenost svih grana/odluka prilikom testiranja bila su potrebna dva testna slučaja i to:

Testni slučajevi:

```
1.  var listProducts = new List<Product>
{
    new Product
    {
        ProductID = 1,
        Name = "Product 1",
        ImageUrl = "url1",
        Price = 60,
        FlowerType = "Rose",
        Stock = 10,
        Category = "Flowers",
        Description = "Beautiful red rose",
        productType = "Type A"
    },
    new Product
    {
        ProductID = 2,
        Name = "Product 2",
        ImageUrl = "url2",
        Price = 20,
        FlowerType = "Lily",
        Stock = 15,
        Category = "Flowers",
        Description = "Elegant white lily",
        productType = "Type B"
    }
};
```

```
2.  var listProducts = null;
```

Nakon testiranja sa prethodno navedena dva testna slučaja postignuta je poptuna obuhvatnost grana/odluka. To je takođe i najmanji broj testnih slučajeva da se to postigne.

1.3 Obuhvat uslova (*Conditional coverage*)

Za obuhvaćenost grana/odluka prilikom testiranja bila su potrebna tri testna slučaja i to:

```
1.  var listProducts = new List<Product>
{
    new Product
    {
        ProductID = 1,
        Name = "Product 1",
        ImageUrl = "url1",
        Price = 60,
        FlowerType = "Rose",
        Stock = 10,
        Category = "Flowers",
        Description = "Beautiful red rose",
        productType = "Type A"
    },
    new Product
    {
        ProductID = 2,
        Name = "Product 2",
        ImageUrl = "url2",
        Price = 20,
        FlowerType = "Lily",
        Stock = 15,
        Category = "Flowers",
        Description = "Elegant white lily",
        productType = "Type B"
    }
};

2.  var listProducts = null;
3.  var listProducts = new List<Product>();
```

Nakon testiranja sa prethodno navedena tri testna slučaja postignuta je poptuna obuhvatnost grana/odluka u kodu.

1.4 Modifikovani uslov/odluka obuhvat (*Modified condition/decision coverage MCDC*)

Za obuhvaćenost modifikovanih uslova/odluka prilikom testiranja bila su potrebna tri testna slučaja i to:

```
1.  var listProducts = new List<Product>
{
    new Product
    {
        ProductID = 1,
        Name = "Product 1",
        ImageUrl = "url1",
        Price = 60,
        FlowerType = "Rose",
        Stock = 10,
        Category = "Flowers",
        Description = "Beautiful red rose",
        productType = "Type A"
    },
    new Product
    {
        ProductID = 2,
        Name = "Product 2",
        ImageUrl = "url2",
        Price = 20,
        FlowerType = "Lily",
        Stock = 15,
        Category = "Flowers",
        Description = "Elegant white lily",
        productType = "Type B"
    }
};

2.  var listProducts = null;
3.  var listProducts = new List<Product>();
```

Prikaz tabele za sledeći uslov:

`if (products != null && products.Any())`

Uslov 1	Uslov 2	Ishod
True	True	True
False	False	False
True	False	False

Dok za sljedeći if imamo samo dva slučaja i to true ili false, ali oni su pokriveni sa prvim testnim slučajem:

```
if (product.Price > 50)
```

1.5 Obuhvat petlji (*Loop coverage*)

Prilikom obuhvata petlji potrebno je proći kroz sljedeće strategije:

1. Preskoči unutrašnjost (tijelo) petlje

- Kada je lista produkata prazna.
- ```
var listProducts = new List<Product>();
```

### 2. Uradi jedan prolaz kroz petlju

- Kada lista produkata sadrži samo jedan produkt.
- ```
var listProducts = new List<Product>
{
    new Product
    {
        ProductID = 1,
        Name = "Product 1",
        ImageUrl = "url1",
        Price = 60,
        FlowerType = "Rose",
        Stock = 10,
        Category = "Flowers",
        Description = "Beautiful red rose",
        productType = "Type A"
    }
};
```

3. Uraditi dva prolaza kroz petlju

- Kada lista produkata sadrži dva elementa(produkta).
- ```
var listProducts = new List<Product>
{
 new Product
 {
 ProductID = 1,
 Name = "Product 1",
 ImageUrl = "url1",
 Price = 60,
 FlowerType = "Rose",
 Stock = 10,
 Category = "Flowers",
 Description = "Beautiful red rose",
 productType = "Type A"
 },
 new Product
 {
 ProductID = 2,
 Name = "Product 2",
 ImageUrl = "url2",
 Price = 20,
```

```

 FlowerType = "Lily",
 Stock = 15,
 Category = "Flowers",
 Description = "Elegant white lily",
 productType = "Type B"
 }
};

```

#### 4. Uraditi slučajan broj prolaza kroz petlju

```

- var listProducts = new List<Product>
{
 new Product
 {
 ProductID = 1,
 Name = "Product 1",
 ImageUrl = "url1",
 Price = 60,
 FlowerType = "Rose",
 Stock = 10,
 Category = "Flowers",
 Description = "Beautiful red rose",
 productType = "Type A"
 },
 new Product
 {
 ProductID = 2,
 Name = "Product 2",
 ImageUrl = "url2",
 Price = 20,
 FlowerType = "Lily",
 Stock = 15,
 Category = "Flowers",
 Description = "Elegant white lily",
 productType = "Type B"
 },
 new Product
 {
 ProductID = 3,
 Name = "Product 2",
 ImageUrl = "url2",
 Price = 20,
 FlowerType = "Lily",
 Stock = 15,
 Category = "Flowers",
 Description = "Elegant white lily",
 productType = "Type B"
 },
 new Product
 {
 ProductID = 4,
 Name = "Product 2",
 ImageUrl = "url2",
 Price = 20,
 FlowerType = "Lily",
 Stock = 15,
 Category = "Flowers",
 Description = "Elegant white lily",
 productType = "Type B"
 }
}

```

```
};
```

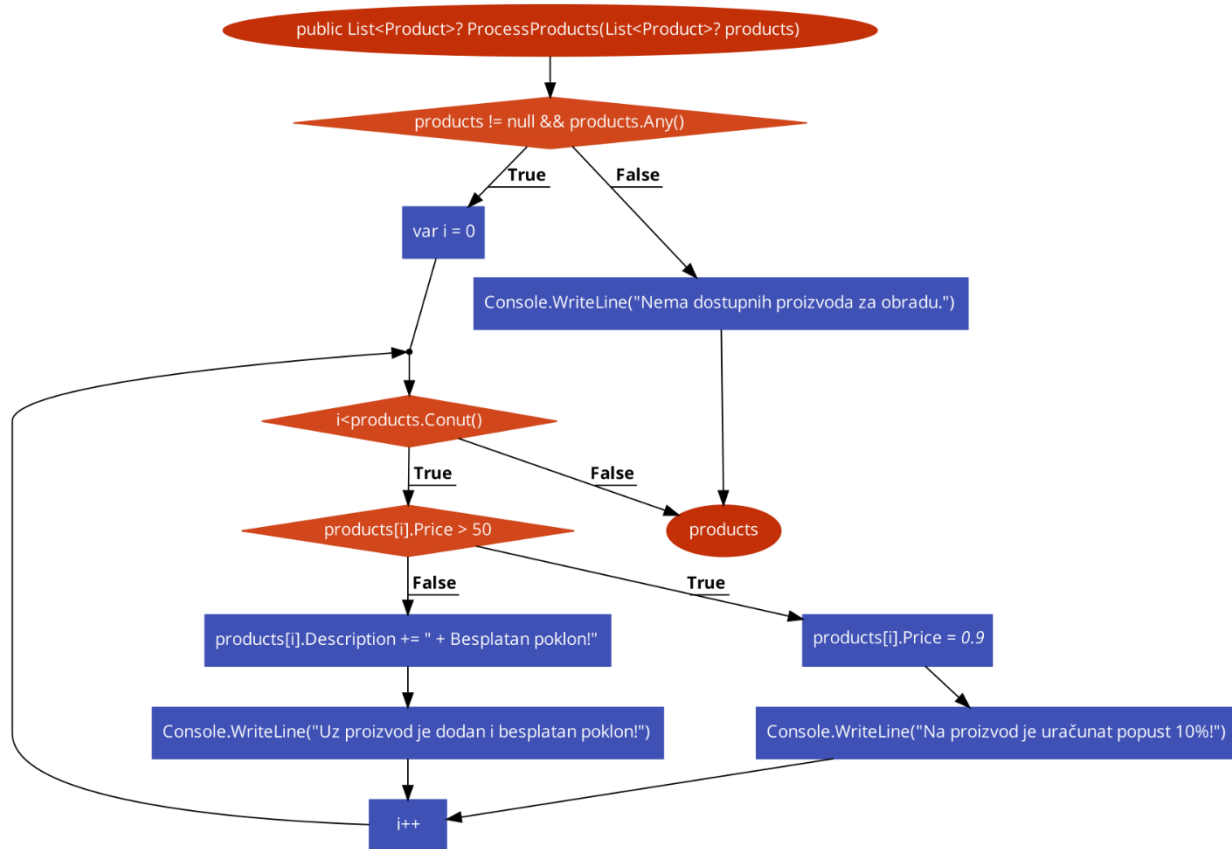
### 5. Uradi n, n-1, n+1 prolaza kroz petlju n znači maksimalni broj prolaza kroz petlju

- Nema smisla sa maksimalnim n probati za lisProducts.Count() (može biti beskonačan broj ponavljanja/produkata, prekoračenje memorije). Moguća greška u programu.

Nakon provođenja ovih principa zaključak je da je potrebno 4 testna slučaja koja su navedena prethodno, s tim da nije moguće testirati kada je n maksimalan broj pa je moguća greška u programu.

### 1.6 Obuhvat puteva (path coverage)

Dijagram toka modula je prikazan na slici ispod:



| Broj puta | Put |
|-----------|-----|
|-----------|-----|

|   |                       |
|---|-----------------------|
| 1 | 1-2-4-5               |
| 2 | 1-2-3-6-5             |
| 3 | 1-2-3-6-7-9-10-11-6-5 |
| 4 | 1-2-3-6-7-8-12-11-6-5 |