

Programowanie Obiektowe

Kyrylo Horiunov

kirillgorunov294@gmail.com

144135

10.09.2023

<https://github.com/Myrzej4/ZarzadzaniaESklepem.git>

- Klasy: Cart, Customer, Product, DiscountProduct, Program, DataService, UserService
- Enumy: LoginOption, MenuOption

```
public class Cart
{
    private List<DiscountProduct> cart = new List<DiscountProduct>();

    // Adding a Product to the Cart
    1 reference
    public void AddProduct(DiscountProduct product)
    {
        cart.Add(product);
    }

    // Displaying the contents of the shopping cart
    1 reference
    public void DisplayCart()
    {
        Console.WriteLine("Cart contents:");
        foreach (var product in cart)
        {
            Console.WriteLine($"{product.Name, -14} | \tPrice: {product.Price,5:C} | \tDiscount: {product.Discount*100}%");
        }
    }

    // Calculating the total cost of products in the cart
    1 reference
    public decimal? CalculateTotal()
    {
        decimal? total = 0;
        foreach (var product in cart)
        {
            total += product.Price - (product.Price * product.Discount);
        }
        return total;
    }
}
```

```

public class Product
{
    private int? productId;
    private string? name;
    private decimal? price;

    3 references
    public int? ProductId
    {
        get { return productId; }
        set { productId = value; }
    }

    3 references
    public string? Name
    {
        get { return name; }
        set { name = value; }
    }

    5 references
    public decimal? Price
    {
        get { return price; }
        set { price = value; }
    }

    /// <summary>
    /// Ctor
    /// </summary>
    /// <param name="productId">Id of Product</param>
    /// <param name="name">Name of product</param>
    /// <param name="price">Product's price</param>
    1 reference
    public Product(int productId, string name, decimal price)
    {
        ProductId = productId;
        Name = name;
        Price = price;
    }
}

```

```

public class DiscountProduct : Product
{
    4 references
    public decimal Discount { get; set; }
    /// <summary>
    /// Ctor
    /// </summary>
    /// <param name="productId"></param>
    /// <param name="name"></param>
    /// <param name="price"></param>
    /// <param name="discount"></param>
    0 references
    public DiscountProduct(int productId, string name, decimal price, decimal discount) : base(productId, name, price)
    {
        Discount = discount;
    }
}

```

```

13 references
public class Customer
{
    private int? customerId;
    private string? name;
    private string? password;
    private Cart cart;

    1 reference
    public int? CustomerId
    {
        get { return customerId; }
        set { customerId = value; }
    }

    7 references
    public string? Name
    {
        get { return name; }
        set { name = value; }
    }

    2 references
    public string? Password
    {
        get { return password; }
        set { password = value; }
    }

    4 references
    public Cart Cart
    {
        get { return cart; }
        private set { cart = value; }
    }

    /// <summary>
    /// Ctor
    /// </summary>
    /// <param name="customerId">ID of Customer</param>
    /// <param name="name">Customer's name</param>
    /// <param name="password">Customer's password</param>
    /// <param name="cart"></param>
    1 reference
    public Customer(int customerId, string name, string password, Cart cart)
    {
        CustomerId = customerId;
        Name = name;
        Password = password;
        Cart = cart;
    }
}

```

```

namespace ZarzadzaniaESklepem.Enum
{
    5 references
    public enum LoginOption
    {
        Exit = 0,
        LogIn = 1,
        Register = 2,
    }
}

```

```

namespace ZarzadzaniaESklepem.Enum
{
    7 references
    public enum MenuOption
    {
        Exit=0,
        ShowAvailableProduct= 1,
        AddToCart = 2,
        PlaceOrder = 3,
        LogOut = 4,
    }
}

```

```

0 references
static void Main()
{
    while (true)
    {
        Console.WriteLine("Menu:");
        Console.WriteLine("1. Enter profile");
        Console.WriteLine("2. Register new profile");
        Console.WriteLine("0. Exit");
        Console.Write("Choose an option: ");

        string? choiceAsString = Console.ReadLine();
        try
        {
            if (choiceAsString == null)
            {
                throw new ArgumentNullException(nameof(choiceAsString));
            }

            LoginOption choice = (LoginOption)Int32.Parse(choiceAsString);
            switch (choice)
            {
                case LoginOption.Login:
                    // Login profile
                    Console.Clear();
                    Login();
                    Console.WriteLine("Press any key to continue");
                    Console.ReadKey();
                    Console.Clear();
                    break;

                case LoginOption.Register:
                    // Register new profile
                    Console.Clear();
                    Register();
                    Console.WriteLine("Press any key to continue");
                    Console.ReadKey();
                    Console.Clear();
                    break;

                case LoginOption.Exit:
                    // Exit app
                    return;
            }
        }
        catch
        {
            Console.WriteLine("Invalid choice. Try again.");
        }
    }

    /// <summary>
    /// Application main screen.
    /// </summary>

    static void MainMenu(Customer customer)
    {
        CultureInfo.CurrentCulture = new CultureInfo("pl-PL");
        Cart cart = new Cart();
        Console.WriteLine("Welcome, {0}", customer.Name);
        while (true)
        {
            Console.WriteLine();
            Console.WriteLine("Main menu:");
            Console.WriteLine("1. Show available products");
            Console.WriteLine("2. Add to cart");
            Console.WriteLine("3. Place Order");
            Console.WriteLine("4. Sign out");
            Console.WriteLine("0. Exit");
            Console.Write("Choose an option: ");

            string? choiceAsString = Console.ReadLine();
            try
            {
                if (choiceAsString == null)
                {
                    throw new ArgumentNullException(nameof(choiceAsString));
                }
                MenuOption choice = (MenuOption)Int32.Parse(choiceAsString);

                switch (choice)
                {
                    case MenuOption.ShowAvailableProduct:

```

```

3 references
public class DataService
{
    private List<DiscountProduct> products;
    private string productsFilePath = "products.json";

    2 references
    public DataService()
    {
        LoadProducts();
    }

    /// <summary>
    /// 
    /// </summary>
    /// <param name="product"></param>
    0 references
    public void AddProduct(DiscountProduct product)
    {
        products.Add(product);
        SaveProducts();
    }

    /// <summary>
    /// Returns list of Products
    /// </summary>
    /// <returns></returns>
    2 references
    public List<DiscountProduct> GetProducts()
    {
        return products;
    }

    /// <summary>
    /// Load products from file
    /// </summary>
    1 reference
    private void LoadProducts()
    {
        if (File.Exists("products.json"))
        {
            string json = File.ReadAllText("products.json");
            var deserializedProducts = JsonConvert.DeserializeObject<List<DiscountProduct>>(json);
            if (deserializedProducts != null)
            {
                products = deserializedProducts;
            }
        }
        else
        {
            products = new List<DiscountProduct>();
        }
    }

    1 reference
    private void SaveProducts()
    {
        string json = JsonConvert.SerializeObject(products, Newtonsoft.Json.Formatting.Indented);
        File.WriteAllText(productsFilePath, json);
    }
}

```

```

3 references
public class UserService
{
    private List<Customer> customers;
    private string customersFilePath = "customers.json";

    2 references
    public UserService()
    {
        LoadCustomers();
    }

    1 reference
    public bool AddCustomer(Customer customer, string username)
    {
        if (customers == null)
        {
            customers = new List<Customer>(); // Initializing a list if it hasn't been initialized
        }
        if (!customers.Any(c => c.Name == username))
        {
            customers.Add(customer);
            SaveCustomers();
            return true;
        }
        else
        {
            Console.WriteLine("User with this name already exists.");
            return false;
        }
    }

    /// <summary>
    /// Get customers list
    /// </summary>
    /// <returns></returns>
    1 reference
    public List<Customer> GetCustomers()
    {
        return customers;
    }

    /// <summary>
    /// Load customers from file
    /// </summary>
    1 reference
    private void LoadCustomers()
    {
        if (File.Exists(customersFilePath))
        {
            string json = File.ReadAllText(customersFilePath);
            var deserializedCustomers = JsonConvert.DeserializeObject<List<Customer>>(json);
            if (deserializedCustomers != null)
            {
                customers = deserializedCustomers;
            }
        }
        else
        {
            customers = new List<Customer>();
        }
    }

    /// <summary>
    /// Save customers to file
    /// </summary>
    2 references
    public void SaveCustomers()

```

Menu:

1. Enter profile
2. Register new profile
0. Exit

Choose an option:

Welcome, Bobby

Main menu:

1. Show available products
2. Add to cart
3. Place Order
4. Sign out
0. Exit

Choose an option:

Available products:

ID: 1	Name: Bread	Price: 2,49 zł	Discount: 10,0%
ID: 2	Name: Milk	Price: 1,99 zł	Discount: 5,00%
ID: 3	Name: Eggs	Price: 3,49 zł	Discount: 0%
ID: 4	Name: Cheese	Price: 6,99 zł	Discount: 15,00%
ID: 5	Name: Cereal	Price: 5,79 zł	Discount: 20,0%
ID: 6	Name: Yogurt	Price: 2,89 zł	Discount: 0%
ID: 7	Name: Apples	Price: 2,29 zł	Discount: 10,0%
ID: 8	Name: Bananas	Price: 5,79 zł	Discount: 0%
ID: 9	Name: Potatoes	Price: 1,19 zł	Discount: 0%
ID: 10	Name: Chicken	Price: 17,99 zł	Discount: 25,00%
ID: 11	Name: Orange Juice	Price: 3,99 zł	Discount: 20,0%
ID: 12	Name: Pasta	Price: 2,49 zł	Discount: 0,0%
ID: 13	Name: Tomatoes	Price: 2,69 zł	Discount: 0,0%
ID: 14	Name: Rice	Price: 4,29 zł	Discount: 5,00%
ID: 15	Name: Salmon	Price: 7,99 zł	Discount: 15,00%
ID: 16	Name: Butter	Price: 5,29 zł	Discount: 0,0%

Main menu:

1. Show available products
2. Add to cart
3. Place Order
4. Sign out
0. Exit

Choose an option: _

Order from a client Bobby:

Cart contents:

Product: Chicken	Price: 17,99 zł	Discount: 25,00%
Product: Rice	Price: 4,29 zł	Discount: 5,00%
Product: Apples	Price: 2,29 zł	Discount: 10,0%
Product: Milk	Price: 1,99 zł	Discount: 5,00%

Total order value: 21,52 zł

Main menu:

1. Show available products
2. Add to cart
3. Place Order
4. Sign out
0. Exit

Choose an option:


```
{
  "ProductId": 1,
  "Name": "Bread",
  "Price": 2.49,
  "Discount": 0.1
},
{
  "ProductId": 2,
  "Name": "Milk",
  "Price": 1.99,
  "Discount": 0.05
},
{
  "ProductId": 3,
  "Name": "Eggs",
  "Price": 3.49,
  "Discount": 0
},
{
  "ProductId": 4,
  "Name": "Cheese",
  "Price": 6.99,
  "Discount": 0.15
},
{
  "ProductId": 5,
  "Name": "Cereal",
  "Price": 5.79,
  "Discount": 0.2
},
{
  "ProductId": 6,
  "Name": "Yogurt",
  "Price": 2.89,
  "Discount": 0
},
{
  "ProductId": 7,
  "Name": "Apples",
  "Price": 2.29,
  "Discount": 0.1
},
{
  "ProductId": 8,
  "Name": "Bananas",
  "Price": 5.79,
  "Discount": 0
},
{
  "ProductId": 9,
  "Name": "Potatoes",
  "Price": 1.19,
  "Discount": 0
}
```