

Ecommerce შეკვეთის დამუშავების სისტემის სიმულაცია

ელექტრონული კომერციის პლატფორმაზე მომხმარებლის მიერ შეკვეთის გაკეთების შემდეგ სისტემა ასრულებს რამდენიმე ოპერაციას შეკვეთის დასამუშავებლად. ზოგიერთი ოპერაცია შეიძლება პარალელურად შესრულდეს, ზოგი კი წინა ნაბიჯების დასრულებაზეა დამოკიდებული. თქვენი ამოცანაა, შექმნათ C# `async/await` და `Task` კონცეფციების გამოყენებით სიმულაციური სისტემა შეკვეთების დამუშავებისათვის.

1: შეკვეთის ნაბიჯების განსაზღვრა

თითოეული შეკვეთა გადის სამ ძირითად ეტაპს:

- გადახდის ვალიდაცია:** იმიტირებს გადახდის წარმატების შემოწმებას. შესრულების დრო: **1-2 წამი.**
- საწყობის შემოწმება:** იმიტირებს, შეკვეთილი ნივთები არის თუ არა საწყობი. შესრულების დრო: **1.5-2.5 წამი.**
- მიწოდების მომზადება:** იწყება მხოლოდ მას შემდეგ, რაც გადახდის ვალიდაცია და საწყობის შემოწმება წარმატებით დასრულდება. შესრულების დრო: **2-3 წამი.**

2: შეკვეთის დამუშავების ლოგიკა

- შექმნით **Order** კლასი, რომელსაც ექნება:
 - Order ID** (უნიკალური იდენტიფიკატორი).
 - Order Status** (მაგალითად: "Processing", "Failed", "Shipped").
- იმპლემენტაცია შემდეგი მეთოდებისათვის:
 - ValidatePayment(int orderId):**
 - იმიტირებს გადახდის შემოწმებას შემთხვევითი **1-2 წამის** დაყოვნებით.
 - ლოგავს შედეგს: Order <ID>: Payment validated.
 - აბრუნებს `bool` მნიშვნელობას წარმატების აღსანიშნად (მარტივად, აქ გადახდა ყოველთვის წარმატებულია).
 - CheckInventory(int orderId):**
 - იმიტირებს საწყობის შემოწმებას შემთხვევითი **1.5-2.5 წამის** დაყოვნებით.
 - ლოგავს შედეგს: Order <ID>: Inventory checked.
 - აბრუნებს `bool` მნიშვნელობას წარმატების აღსანიშნად.

- **PrepareShipment(int orderId):**

- იმიტირებს მიწოდების მომზადებას შემთხვევითი **2-3 წამის** დაყოვნებით.
- ლოგავს შედეგს: Order <ID>: Shipment prepared.

- **ProcessOrder(int orderId):**

- აერთიანებს ზემოთ ჩამოთვლილ ნაბიჯებს:
 - გადახდის ვალიდაცია და საწყობის შემოწმება უნდა შესრულდეს **პარალელურად**.
 - თუ ორივე წარმატებულია, იწყება მიწოდების მომზადება.
 - თუ რომელიმე ნაბიჯი ჩავარდება, შეკვეთის სტატუსი ხდება "**Failed**".
- გამოიყენეთ **Task.WhenAll**, რომ გადახდის ვალიდაცია და საწყობის შემოწმება ერთად შესრულდეს.
- შექმენით **5 შეკვეთის სიმულაცია**. თითოეული შეკვეთისთვის ასინქრონულად გაუშვით **ProcessOrder** მეთოდი.