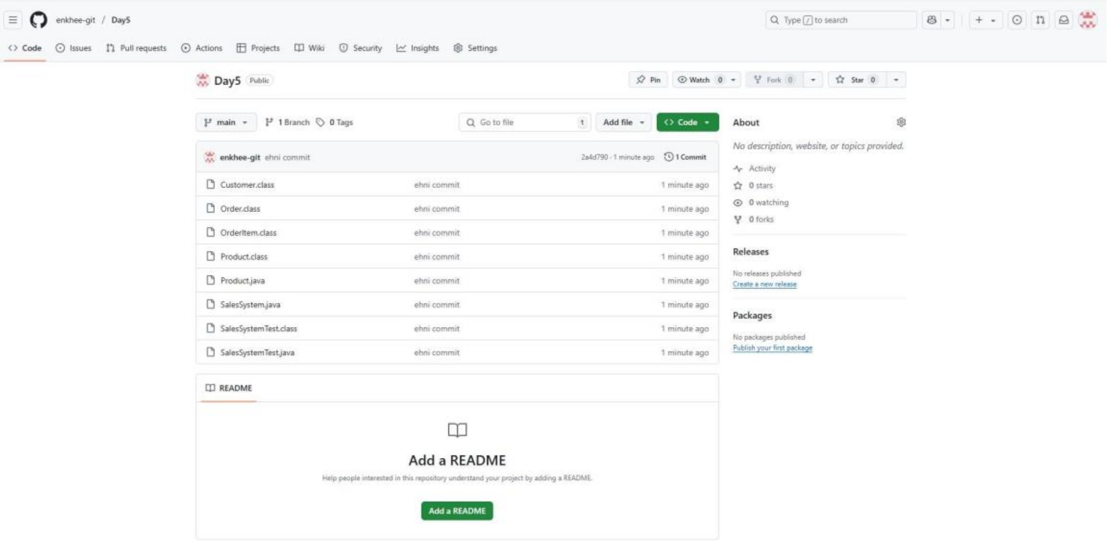
B232270141 Э.Энхбат
B232270805 Г.Үлэмж
B232270051 Х.Шинэжаргал
DAy5



enkhee-git ehni commit 2a4d790 · 1 minute ago 1 Commit

Customer.class	ehni commit	1 minute ago
Order.class	ehni commit	1 minute ago
OrderItem.class	ehni commit	1 minute ago
Product.class	ehni commit	1 minute ago
Product.java	ehni commit	1 minute ago
SalesSystem.java	ehni commit	1 minute ago
SalesSystemTest.class	ehni commit	1 minute ago
SalesSystemTest.java	ehni commit	1 minute ago

Add a README

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

Activity

0 stars

0 watching

0 forks

Releases

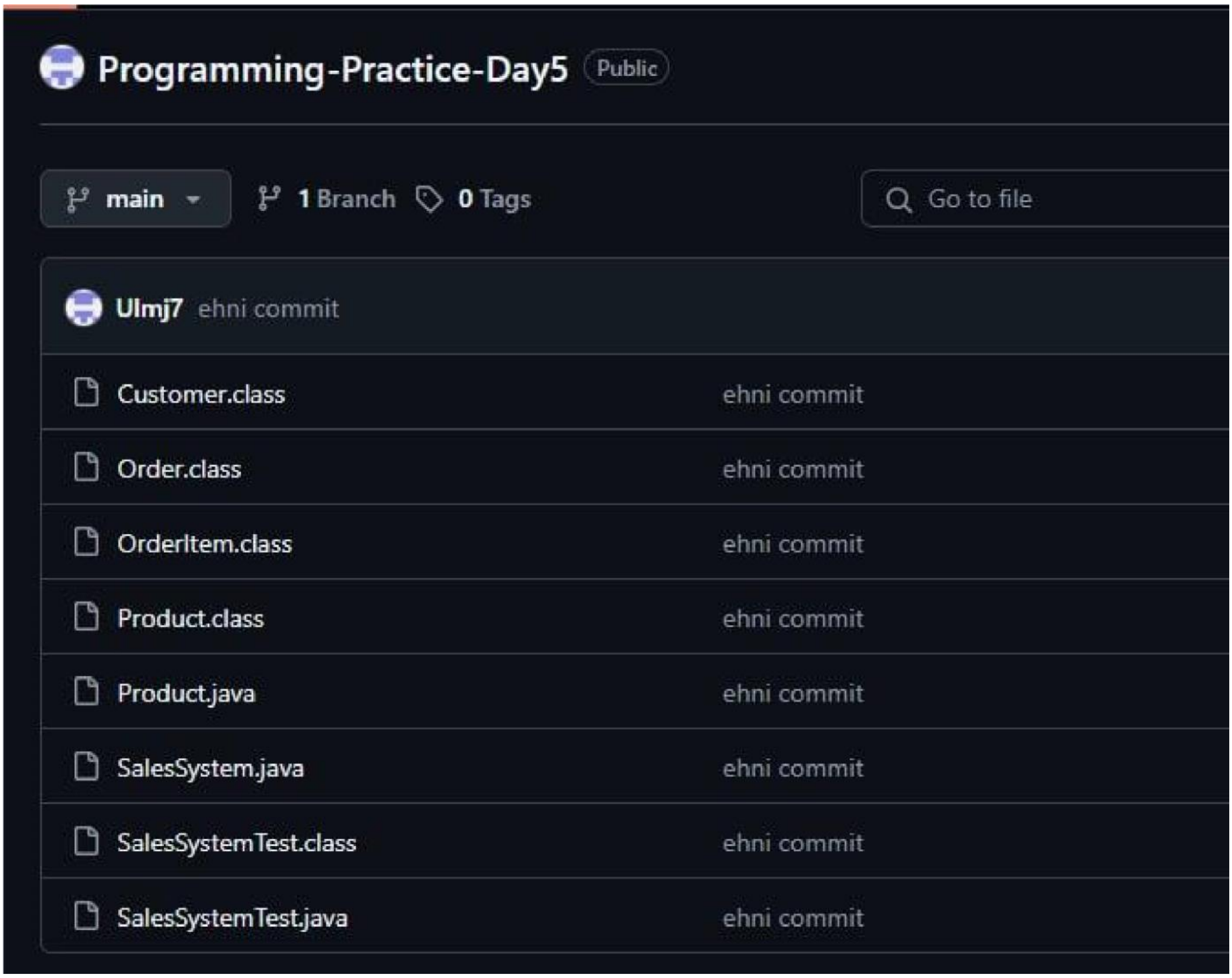
No releases published

Create a new release

Packages

No packages published

Publish your first package



Programming-Practice-Day5

Ulmj7 ehni commit

Customer.class	ehni commit
Order.class	ehni commit
OrderItem.class	ehni commit
Product.class	ehni commit
Product.java	ehni commit
SalesSystem.java	ehni commit
SalesSystemTest.class	ehni commit
SalesSystemTest.java	ehni commit

Producy.java

```
public class Product {
    private String name;
    private double price;
    private int stockQuantity;

    public Product(String name, double price, int stockQuantity) {
        if (!name.matches("[a-zA-Za-яА-Я\\s]+")) { throw new
        IllegalArgumentException("Нэр зөвхөн үсэг байж
болно.");
        }
        if (price <= 0) { throw new IllegalArgumentException("Үнэ
            эерэг тоо байх
ёстой.");
        }
        if (stockQuantity < 0) { throw new
            IllegalArgumentException("Үлдэгдэл сөрөг байж
болохгүй.");
        }
    this.name = name; this.price = price;
    this.stockQuantity = stockQuantity;
}

    public String getName() { return name; } public double
    getPrice() { return price; } public int
    getStockQuantity() { return stockQuantity; }
    public void reduceStock(int qty) { if (qty < 0 || qty >
        stockQuantity) { throw new
        IllegalArgumentException("Буруу тоо ширхэг
байна.");
        }
        stockQuantity -= qty;
    }
    public void increaseStock(int qty) {
        if (qty < 0) {
            throw new IllegalArgumentException("Нэмэгдэх тоо эерэг байх
ёстой.");
        }
        stockQuantity += qty;
    }
    public String toString() { return name + " - " + price + "₮
        (" + stockQuantity + "
ширхэг)";
    }
}
```

SalesSystem.java

```
import java.util.*; // Зөв List интерфейсийг оруулна

class Product { private
    String name; private
    double price; private int
    stockQuantity;
    public Product(String name, double price, int
        stockQuantity) { this.name = name; this.price =
        price;
        this.stockQuantity = stockQuantity; }
    public String getName() { return name; }

    public double getPrice() { return price; }
    public int getStockQuantity() { return stockQuantity; }


    public void reduceStock(int    y) {
        qt stockQuantity -= qty;
    }


   public void increaseStock(int   qty) {
       stockQuantity += qty;
    }
    public String toString() { return name + " - " + price + "₮
        (" + stockQuantity + "
ширхэг)";
    }
}
class Customer { private String name,
    email, phone;

    public Customer(String name, String email, String phone) {
    this.name = name; this.email = email; this.phone = phone;
    }
    public String getName() { return name; }
}
class OrderItem { private
    Product product; private
    int quantity;

    public OrderItem(Product product, int quantity) {
    this.product = product; this.quantity = quantity;
    }
    public String toString() { return product.getName() +
        " - " + quantity + "ш";
    }
}
```

```
class Order { private Customer
    customer; private
    List<OrderItem> items;

    public Order(Customer customer) { this.customer =
        customer; this.items = new ArrayList<>();
    }
public boolean addItem(Product product, int quantity) { if
    (quantity <= product.getStockQuantity()) {
    items.add(new OrderItem(product, quantity));
    product.reduceStock(quantity); return true;
    } else {
        return false;
    }
}

    public String getOrderDetails() {
        StringBuilder sb = new StringBuilder(); sb.append("Захиалагч:
").append(customer.getName()).append("\n"); for (OrderItem item : items)
        { sb.append(" - ").append(item).append("\n");
        } return sb.toString();
    }
}
```

SalesSystemTest.java

```
import org.junit.jupiter.api.Test; import static

org.junit.jupiter.api.Assertions.*; import

java.util.List;

public class SalesSystemTest {
@Test
public void testProductCreation() {
    Product p = new Product("Физик", 18000, 10);
    assertEquals("Физик", p.getName());
    assertEquals(18000, p.getPrice()); assertEquals(10,
    p.getStockQuantity());
}
@Test
public void testReduceAndIncreaseStock() {
    Product p = new Product("Математик", 20000, 5);
    p.reduceStock(2); assertEquals(3,
    p.getStockQuantity());
    p.increaseStock(4); assertEquals(7,
    p.getStockQuantity());
}
    @Test
    public void testAddItemToOrderSuccessfully() {
        Product p = new Product("Гун ухаан", 15000, 10);
        Customer c = new Customer("Баатар", "b@example.com",
"99112233");
        Order order = new Order(c);
    boolean added = order.addItem(p, 3);
    assertTrue(added); assertEquals(7,
    p.getStockQuantity());
}
@Test
public void testAddItemToOrderFailsWhenInsufficientStock() {
    Product p = new Product("Математик", 20000, 2);
        Customer c = new Customer("Сараа", "s@example.com", "99114455");
    Order order = new Order(c);
    boolean added = order.addItem(p, 5); assertFalse(added);
    assertEquals(2, p.getStockQuantity()); }


    @Test

public void testOrderDetailsFormatting() {
    Product p = new Product("Физик", 18000, 5);
    Customer c = new Customer("Болд", "b@example.com", "99001122");
    Order order = new Order(c); order.addItem(p, 2);

        String details = order.getOrderDetails();
    assertTrue(details.contains("Захиалагч: Болд"));
    assertTrue(details.contains("Физик - 2ш")); }
}
```