CR3 – Introducing the Final-Amount-Reducing Coupon

The marketing team has a great idea!
They think it is time to be truly generous:
a new coupon called **KUPON-2000-ULTRAMAX** is introduced, which deducts 2000 HUF from the final total – yes, from the entire total amount, regardless of what products are in the cart or what period we are in.

Hopefully, sales will skyrocket!

New Coupon Type

- **KUPON-2000-ULTRAMAX**
- Deducts 2000 HUF from the total final amount
- Can be combined with any other coupon
- No restrictions apply
- If the discount would lead to a negative final total, the total will be set to 0 HUF
- Multiple KUPON-2000-ULTRAMAX coupons can be used consecutively

Important rule:
The system always applies KUPON-2000-ULTRAMAX coupons at the very end, regardless of the order in which they were received.
This ensures that other coupons – especially non-combinable, product-specific discounts – are applied first and their effect is not lost because of a zero subtotal.

Examples

1. Only one KUPON-2000-ULTRAMAX

Cart: 3 kg apple
Period: Normal
Coupons: KUPON-2000-ULTRAMAX

Apple price: 3 × 500 = 1500 HUF
No discount
Coupon deduction: −2000 HUF → negative → 0 HUF
Payable: 0 HUF
Returned coupons: []

---

2. A10 coupon + ULTRAMAX

Cart: 4.1 kg apple
Period: Normal
Coupons: A10, KUPON-2000-ULTRAMAX

Apple price: 4.1 × 500 = 2050 HUF
No quantity discount
A10 → 10% → 1845 HUF
ULTRAMAX: 1845 − 2000 = negative → 0 HUF
Payable: 0 HUF
Returned coupons: []

2/B. Same with reversed coupon order

Cart: 4.1 kg apple
Period: Normal
Coupons: KUPON-2000-ULTRAMAX, A10

The system applies the ULTRAMAX coupon at the end.
So the A10 coupon applies first.
Final: 2050 → 1845 → −2000 = 0 HUF
Payable: 0 HUF
Returned coupons: []

---

3. Two ULTRAMAX coupons in a row

Cart: 7 kg banana
Period: Normal
Coupons: ULTRAMAX, ULTRAMAX

Banana price: 7 × 450 = 3150 HUF
Quantity discount: 10% → 2835 HUF
Coupon 1: 2835 − 2000 = 835 HUF
Coupon 2: 835 − 2000 = negative → 0 HUF
Payable: 0 HUF
Returned coupons: []

---

4. Valid coupon + ULTRAMAX

Cart: 1.5 kg apple
Period: Normal
Coupons: A10, ULTRAMAX

Apple price: 750 HUF
A10 → 10% → 675 HUF

ULTRAMAX: 675 − 2000 = 0 HUF
Payable: 0 HUF
Returned coupons: []

---

5. Two products, two coupons, one valid

Cart: 1 kg apple, 1 kg banana
Period: Normal
Coupons: A5, ULTRAMAX

Apple: 500 → A5 → 475 HUF
Banana: 450 HUF
Total: 925 HUF
ULTRAMAX: 925 − 2000 = 0 HUF
Payable: 0 HUF
Returned coupons: []

---

6. One invalid coupon, only ULTRAMAX applies

Cart: 1 kg apple
Period: Normal
Coupons: B10, ULTRAMAX

B10: not applicable (not banana) → returned
Apple: 500 HUF
ULTRAMAX: 500 − 2000 = 0 HUF
Payable: 0 HUF
Returned coupons: [B10]

---

7. Payable is not 0 HUF – larger cart, one ULTRAMAX

Cart: 10 kg apple
Period: Normal
Coupons: ULTRAMAX

Apple: 5000 HUF
Discount: 10% → 4500 HUF
ULTRAMAX: 4500 − 2000 = 2500 HUF
Payable: 2500 HUF
Returned coupons: []

---

8. Payable not 0 HUF – too large cart

Cart: 15 kg banana
Period: Normal
Coupons: ULTRAMAX

Banana: 6750 HUF → 10% discount → 6075 HUF
ULTRAMAX: 6075 − 2000 = 4075 HUF
Payable: 4075 HUF
Returned coupons: []

---

9. Two ULTRAMAX coupons – not zero, but close

Cart: 10 kg banana
Period: Normal
Coupons: ULTRAMAX, ULTRAMAX

Banana: 4500 HUF
10% discount → 4050 HUF
Coupon 1: 4050 − 2000 = 2050 HUF
Coupon 2: 2050 − 2000 = 50 HUF
Payable: 50 HUF
Returned coupons: []

---

10. Critical order problem – solved by the rule

Cart: 2 kg apple, 2 kg banana
Period: Normal
Coupons: ULTRAMAX, A5

Apple: 1000 HUF
Banana: 900 HUF → 10% discount → 810 HUF
Total: 1900 HUF
A5 → apple price 1000 → 950 HUF
Final: 950 + 810 = 1760 HUF
ULTRAMAX: 1760 − 2000 = 0 HUF
Payable: 0 HUF
Returned coupons: []

Note: If the ULTRAMAX coupon were applied first, the A5 would become invalid. Therefore, it is essential that ULTRAMAX is always applied last.

Development Requirements

The system must treat the KUPON-2000-ULTRAMAX type specially, because:

- It can be combined with all other coupons
- No discount ceiling applies
- Must always be applied last
- Can be used multiple times in the same purchase
- If the total would be negative, it must be set to 0 HUF

Note

This coupon is a true “marketing bomb,” and if the promotion succeeds, GoodPrice store will almost certainly introduce further special coupons.
It is likely that the management will request additional Change Requests (CRs) in the future.

To make the task even more precise, here are the unit tests for the examples:

Java Unit Tests

```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import java.util.List;
import org.store.*;

class StoreCR3Tests {
    static Store target;
    static Period normal;
    @BeforeAll
    public static void initStore() {
        target = new Store();
        normal = new Period("Normal");
        normal.setUnitPrice(Product.APPLE, 500.0);
        normal.setUnitPrice(Product.BANANA, 450.0);
        normal.setDiscount(Product.APPLE, 5.0, 0.1);
        normal.setDiscount(Product.APPLE, 20.0, 0.15);
        normal.setDiscount(Product.BANANA, 2.0, 0.1);
        target.addPeriod(normal);
    }
    @Test
    void test_cr3_example1_onlyUltramax() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 3.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("KUPON-
2000-ULTRAMAX"));
        assertEquals(0.0, price.getAmount(), 0.001);
        assertEquals(List.of(), price.getUnusedCoupons());
    }
    @Test
    void test_cr3_example2_a10PlusUltramax() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 4.1)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A10",
"KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, price.getAmount(), 0.001);
        assertEquals(List.of(), price.getUnusedCoupons());
    }
```

```
    @Test
    void test_cr3_example2b_ultramaxPlusA10() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 4.1)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("KUPON-
2000-ULTRAMAX", "A10"));
        assertEquals(0.0, price.getAmount(), 0.001);
        assertEquals(List.of(), price.getUnusedCoupons());
    }
    @Test
    void test_cr3_example3_twoUltramax() {
        Cart cart = new Cart(List.of(new Item(Product.BANANA, 7.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("KUPON-
2000-ULTRAMAX", "KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, price.getAmount(), 0.001);
    }
    @Test
    void test_cr3_example4_a10AndUltramax() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 1.5)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A10",
"KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, price.getAmount(), 0.001);
    }
    @Test
    void test_cr3_example5_twoProducts_oneCoupon() {
        Cart cart = new Cart(List.of(
                new Item(Product.APPLE, 1.0),
                new Item(Product.BANANA, 1.0)
        ));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A5",
"KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, price.getAmount(), 0.001);
    }
    @Test
    void test_cr3_example6_b10Unused() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 1.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("B10",
"KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, price.getAmount(), 0.001);
        assertEquals(List.of("B10"), price.getUnusedCoupons());
    }
    @Test
    void test_cr3_example7_notZeroFinal() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 10.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("KUPON-
2000-ULTRAMAX"));
        assertEquals(2500.0, price.getAmount(), 0.001);
    }
    @Test
    void test_cr3_example8_bigBananaCart() {
        Cart cart = new Cart(List.of(new Item(Product.BANANA, 15.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("KUPON-
2000-ULTRAMAX"));
        assertEquals(4075.0, price.getAmount(), 0.001);
    }
    @Test
    void test_cr3_example9_twoUltramax_notZero() {
        Cart cart = new Cart(List.of(new Item(Product.BANANA, 10.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("KUPON-
2000-ULTRAMAX", "KUPON-2000-ULTRAMAX"));
        assertEquals(50.0, price.getAmount(), 0.001);
    }
```

```
    @Test
    void test_cr3_example10_criticalOrder() {
        Cart cart = new Cart(List.of(
                new Item(Product.APPLE, 2.0),
                new Item(Product.BANANA, 2.0)
        ));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("KUPON-
2000-ULTRAMAX", "A5"));
        assertEquals(0.0, price.getAmount(), 0.001);
    }
}
```