

## CR3 – A végösszeg-csökkentő kupon bevezetése

**A marketingeseknek van egy jó ötlete!**

Úgy gondolják, hogy ideje igazán nagylelkűnek lenni:

**bevezetésre kerül a KUPON-2000-ULTRAMAX** nevű kupon, amely 2000 forintot von le a végösszezből – igen, a teljes végösszezből, függetlenül attól, hogy milyen termékek kerültek a kosárba, vagy milyen időszakban járunk.

**Reméljük, felfutnak az eladások!**

---

### Új kupon típusa

**KUPON-2000-ULTRAMAX**

- **2000 forintot** von le a **teljes végösszezből**
- **Bármilyen más kuponnal összevonható**
- **Semmilyen korlátozás nem vonatkozik rá**
- Ha a kedvezmény **negatív végösszeghez vezetne**, akkor a végösszeg **0 forint** lesz
- Több KUPON-2000-ULTRAMAX kupon is használható **egymás után**

**Fontos szabály:**

A KUPON-2000-ULTRAMAX típusú kuponokat **a rendszer mindig csak a legvégén alkalmazza, függetlenül attól, milyen sorrendben kaptuk őket.**

Ez biztosítja, hogy más kuponok – különösen a **nem összevonható, termékspecifikus** kedvezmények – előbb érvényesülhessenek, és ne vesszen el a hatásuk egy 0 forintos részösszeg miatt.

---

### Példák

#### 1. Csak egy KUPON-2000-ULTRAMAX

**Kosár:** 3 kg alma

**Időszak:** Normál

**Kuponok:** KUPON-2000-ULTRAMAX

- Alma ára:  $3 \times 500 = 1500$  Ft
  - Nincs kedvezmény
  - Kupon levonás: -2000 Ft  $\rightarrow$  negatív  $\rightarrow$  **0 Ft**
  - **Fizetendő:** 0 Ft
  - **Visszaadott kuponok:** []
- 

#### 2. A10 kupon + ULTRAMAX

**Kosár:** 4,1 kg alma

**Időszak:** Normál

**Kuponok:** A10, KUPON-2000-ULTRAMAX

- Alma ára:  $4,1 \times 500 = 2050$  Ft
  - Nincs mennyiségi kedvezmény
  - A10  $\rightarrow 10\% \rightarrow 1845$  Ft
  - KUPON-2000-ULTRAMAX:  $1845 - 2000 = \text{negatív} \rightarrow \mathbf{0}$  Ft
  - **Fizetendő:** 0 Ft
  - **Visszaadott kuponok:** []
- 

## 2/B. Ugyanez fordított kupon sorrenddel

**Kosár:** 4,1 kg alma

**Időszak:** Normál

**Kuponok:** KUPON-2000-ULTRAMAX, A10

- A rendszer a KUPON-2000-ULTRAMAX kupont **a végén** alkalmazza
  - Ezért az A10 kupon **érvényesül** előbb
  - Végösszeg:  $2050 \rightarrow 1845 \rightarrow -2000 = \mathbf{0}$  Ft
  - **Fizetendő:** 0 Ft
  - **Visszaadott kuponok:** []
- 

## 3. Két KUPON-2000-ULTRAMAX kupon egymás után

**Kosár:** 7 kg banán

**Időszak:** Normál

**Kuponok:** KUPON-2000-ULTRAMAX, KUPON-2000-ULTRAMAX

- Banán ára:  $7 \times 450 = 3150$  Ft
  - Mennyiségi kedvezmény:  $10\% \rightarrow 2835$  Ft
  - Kupon 1:  $2835 - 2000 = 835$  Ft
  - Kupon 2:  $835 - 2000 \rightarrow \text{negatív} \rightarrow \mathbf{0}$  Ft
  - **Fizetendő:** 0 Ft
  - **Visszaadott kuponok:** []
- 

## 4. Érvényes kupon + KUPON-2000-ULTRAMAX

**Kosár:** 1.5 kg alma

**Időszak:** Normál

**Kuponok:** A10, KUPON-2000-ULTRAMAX

- Alma ára: 750 Ft
- A10  $\rightarrow 10\% \rightarrow 675$  Ft

- ULTRAMAX:  $675 - 2000 \rightarrow 0$  Ft
  - **Fizetendő:** 0 Ft
  - **Visszaadott kuponok:** []
- 

## 5. Két termék, két kupon, egyik érvényes

**Kosár:** 1 kg alma, 1 kg banán

**Időszak:** Normál

**Kuponok:** A5, KUPON-2000-ULTRAMAX

- Alma:  $500 \rightarrow A5 \rightarrow 475$  Ft
  - Banán: 450 Ft
  - Összesen: 925 Ft
  - ULTRAMAX:  $925 - 2000 \rightarrow 0$  Ft
  - **Fizetendő:** 0 Ft
  - **Visszaadott kuponok:** []
- 

## 6. Egyik kupon sem érvényes, csak ULTRAMAX

**Kosár:** 1 kg alma

**Időszak:** Normál

**Kuponok:** B10, KUPON-2000-ULTRAMAX

- B10: nem alkalmazható (nem banán)  $\rightarrow$  visszaadás
  - Alma: 500 Ft
  - ULTRAMAX:  $500 - 2000 \rightarrow 0$  Ft
  - **Fizetendő:** 0 Ft
  - **Visszaadott kuponok:** [B10]
- 

## 7. Fizetendő nem 0 Ft – nagyobb kosár, egy ULTRAMAX

**Kosár:** 10 kg alma

**Időszak:** Normál

**Kuponok:** KUPON-2000-ULTRAMAX

- Alma: 5000 Ft
  - Kedvezmény: 10%  $\rightarrow 4500$  Ft
  - ULTRAMAX:  $4500 - 2000 = 2500$  Ft
  - **Fizetendő:** 2500 Ft
  - **Visszaadott kuponok:** []
- 

## 8. Fizetendő nem 0 Ft – túl nagy kosár

**Kosár:** 15 kg banán

**Időszak:** Normál

**Kuponok:** KUPON-2000-ULTRAMAX

- Banán: 6750 Ft  $\rightarrow$  10% kedvezmény  $\rightarrow$  6075 Ft
  - ULTRAMAX:  $6075 - 2000 = \mathbf{4075\text{ Ft}}$
  - **Fizetendő:** 4075 Ft
  - **Visszaadott kuponok:** []
- 

## 9. Két ULTRAMAX kupon – nem 0 Ft, de közel

**Kosár:** 10 kg banán

**Időszak:** Normál

**Kuponok:** KUPON-2000-ULTRAMAX, KUPON-2000-ULTRAMAX

- Banán: 4500 Ft
  - 10% kedvezmény  $\rightarrow$  4050 Ft
  - Kupon 1:  $4050 - 2000 = 2050\text{ Ft}$
  - Kupon 2:  $2050 - 2000 = \mathbf{50\text{ Ft}}$
  - **Fizetendő:** 50 Ft
  - **Visszaadott kuponok:** []
- 

## 10. Kritikus sorrendprobléma – megoldva a szabállyal

**Kosár:** 2 kg alma, 2 kg banán

**Időszak:** Normál

**Kuponok:** KUPON-2000-ULTRAMAX, A5

- Alma: 1000 Ft
- Banán: 900 Ft  $\rightarrow$  10% mennyiségi kedvezmény  $\rightarrow$  810 Ft
- Összesen: 1900 Ft
- A5  $\rightarrow$  alma ára 1000  $\rightarrow$  950 Ft
- Végösszeg:  $950 + 810 = 1760\text{ Ft}$
- ULTRAMAX:  $1760 - 2000 \rightarrow \mathbf{0\text{ Ft}}$
- **Fizetendő:** 0 Ft
- **Visszaadott kuponok:** []

Megjegyzés: ha a KUPON-2000-ULTRAMAX kupont előbb alkalmaznánk, az A5 **érvénytelenné válna**. Ezért is **elengedhetetlen**, hogy az ULTRAMAX kupon mindig **utoljára** kerüljön feldolgozásra.

---

## Fejlesztési követelmények

- A rendszernek a KUPON-2000-ULTRAMAX típust **speciálisan kell kezelnie**, mivel:

- Összevonható **minden más kuponnal**
  - Nem vonatkozik rá semmilyen kedvezményplafon
  - **Mindig a legvégén** kell érvényesíteni
  - **Többször is használható** ugyanabban a vásárlásban
- A negatív végösszeg esetén a visszatérített érték legyen **0 Ft**

---

## Megjegyzés

Ez a kupon egy igazi „marketing bomba”, és ha sikeres lesz az akció, akkor a **JóÁr** áruház szinte biztosan újabb különleges kuponokat fog bevezetni.

Elképzeltető, hogy a **JóÁr** áruház vezetősége további CR-eket fog kérni a jövőben.

Hogy még pontosabb legyen a feladat megfogalmazása, megadjuk a példákhoz tartozó unit tesztet:

```
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.aruhaz.*;

class AruhazCR3Tesztek {
    static Aruhaz target;
    static Idoszak normal;
    @BeforeAll
    public static void initAruhaz() {
        target = new Aruhaz();
        normal = new Idoszak("Normál");
        normal.setEgysegAr(Termek.ALMA, 500.0);
        normal.setEgysegAr(Termek.BANAN, 450.0);
        normal.setKedvezmeny(Termek.ALMA, 5.0, 0.1);
        normal.setKedvezmeny(Termek.ALMA, 20.0, 0.15);
        normal.setKedvezmeny(Termek.BANAN, 2.0, 0.1);
        target.addIdoszak(normal);
    }
    @Test
    void teszt_cr3_pelda1_csakUltramax() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 3.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr3_pelda2_a10plusUltramax() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 4.1)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A10", "KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr3_pelda2b_Ultramaxplusa10() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 4.1)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("KUPON-2000-ULTRAMAX", "A10"));
    }
}
```

```

        assertEquals(0.0, ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr3_pelda3_ketUltramax() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 7.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("KUPON-2000-ULTRAMAX", "KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, ar.getAr(), 0.001);
    }
    @Test
    void teszt_cr3_pelda4_al0esUltramax() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.5)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A10", "KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, ar.getAr(), 0.001);
    }
    @Test
    void teszt_cr3_pelda5_ketTermek_egyKupon() {
        Kosar kosar = new Kosar(List.of(
            new Tetel(Termek.ALMA, 1.0),
            new Tetel(Termek.BANAN, 1.0)
        ));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A5", "KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, ar.getAr(), 0.001);
    }
    @Test
    void teszt_cr3_pelda6_b10Hasznalhatatlan() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("B10", "KUPON-2000-ULTRAMAX"));
        assertEquals(0.0, ar.getAr(), 0.001);
        assertEquals(List.of("B10"), ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr3_pelda7_nemNullaVegosszeg() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 10.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("KUPON-2000-ULTRAMAX"));
        assertEquals(2500.0, ar.getAr(), 0.001);
    }
    @Test
    void teszt_cr3_pelda8_banankupac() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 15.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("KUPON-2000-ULTRAMAX"));
        assertEquals(4075.0, ar.getAr(), 0.001);
    }
    @Test
    void teszt_cr3_pelda9_ketUltramax_nemNulla() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 10.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("KUPON-2000-ULTRAMAX", "KUPON-2000-ULTRAMAX"));
        assertEquals(50.0, ar.getAr(), 0.001);
    }
    @Test
    void teszt_cr3_pelda10_kritikusSorrend() {
        Kosar kosar = new Kosar(List.of(
            new Tetel(Termek.ALMA, 2.0),
            new Tetel(Termek.BANAN, 2.0)
        ));
    }

```

```
    ));  
    ArInfo ar = target.getKosarAr(kosar, normal, List.of("KUPON-2000-  
ULTRAMAX", "A5"));  
    assertEquals(0.0, ar.getAr(), 0.001);  
  }  
}
```