

CR4 – Összevonható kuponok MAX kedvezménnyel

A JóÁr áruház marketingesei nem pihennek!

A vásárlói hűség növelése érdekében új kuponokat vezetnek be: **MAX kuponokat**, amelyek egymással összevonhatók, de csak meghatározott maximum kedvezményig!

Ezek a kuponok elsősorban **alma** vásárlására vonatkoznak, de a jövőben más termékekre is várható hasonló típus. A rendszernek mostantól képesnek kell lennie ezek feldolgozására.

Új kupon típusok

A5-MAX10

- **5% kedvezmény** az alma teljes árából
- **Összevonható** más A*, illetve A*-MAX* kuponokkal
- A **maximális kedvezmény** az almára: **10%**

A5-MAX15

- **5% kedvezmény** az alma teljes árából
- **Összevonható** más A*, illetve A*-MAX* kuponokkal
- A **maximális kedvezmény** az almára: **15%**

Ha **több MAX kupon is szerepel**, a rendszer a **legkisebb megengedett MAX százalékot** veszi figyelembe. Például, ha van egy A5-MAX10 és egy A5-MAX15, akkor **legfeljebb 10% kedvezmény** adható összesen. Ha több kupon is van, akkor az összevonhatósághoz legalább egy MAX kupon kell, ugyanakkor mindegy, hogy ez a MAX kupon hányadik a sorban.

Használati szabályok

1. **Összevonás megengedett**, de csak az **azonos termék típusára** vonatkozó kuponok között (pl. alma).
 2. A rendszer az **egyes kuponokat sorban dolgozza fel**.
 3. A kuponok összeadódnak (pl. A5 + A5-MAX15 = 10%), de **nem haladhatják meg a legszigorúbb MAX értéket**.
 4. Ha egy másik akció **önmagában jobb kedvezményt ad**, mint a kuponok összesen, a **kuponokat nem használjuk fel**.
 5. A rendszer visszaadja azokat a kuponokat, amelyeket nem tudott felhasználni.
 6. Az **összevonás érvényesítéséhez legalább egy MAX típusú kuponnak szerepelnie kell**, és mindegy, hogy ez a MAX kupon hol helyezkedik a kuponok közt.
 7. Ha több kupon is visszaadható, és ezek között van MAX típusú kupon, akkor a rendszer a MAX típusút adja vissza, mégpedig a legnagyobb MAX értékkel rendelkező kupont.
-

Példák

1. Egy MAX kupon egyedül

Kosár: 2 kg alma

Időszak: Normál

Kuponok: A5-MAX10

Ár: $2 \times 500 = 1000$ Ft

Mennyiségi kedvezmény: nincs

MAX: 10%, kuponkedvezmény: 5% → **engedélyezett**

Fizetendő: 950 Ft

Visszaadott kuponok: []

2. Két A5 kupon + A5-MAX10 → MAX 10% elérve

Kosár: 1 kg alma

Időszak: Normál

Kuponok: A5, A5, A5-MAX10

Ár: 500 Ft

Első A5 → -5%

Második A5 → újabb -5%, ez így már -10%, de még nem is biztos, hogy összevonhatók

A5-MAX10: MAX: 10%, újabb 5% kuponkedvezmény: összesen már 15%, de a max 10% miatt a három kupon csak 10%-ot ad, de ennek a 10% az eléréséhez elég lett volna csak egy A5 az A5-MAX10 mellett, ezért egy A5 visszajár

Fizetendő: 450 Ft

Visszaadott kuponok: [A5]

3. A5, A5-MAX15 és A5-MAX10 → csak 10% érvényes

Kosár: 1 kg alma

Időszak: Normál

Kuponok: A5, A5-MAX15, A5-MAX10

Ár: 500 Ft

MAX értékek: 15% és 10% → legszigorúbb: 10%

A5 → 5% kedvezmény

A5-MAX15 → 5% kedvezmény, MAX kedvezmény 15%

A5-MAX10 → 5% kedvezmény, MAX kedvezmény = $\text{Math.min}(15\%, 10\%)$

Fizetendő: 450 Ft

Visszaadott kuponok: [A5-MAX15], mert ennek az 5% nem kellett a 10% eléréséhez, és ez „értékesebb” kupon.

4. MAX15 és MAX10 → csak 10% érvényes

Kosár: 1 kg alma

Időszak: Normál

Kuponok: A5-MAX15, A5-MAX10

Ár: 500 Ft

MAX értékek: 15% és 10% → legszigorúbb: 10%

A5-MAX15 → 5% kedvezmény, MAX kedvezmény 15%

A5-MAX10 → 5% kedvezmény, MAX kedvezmény = $\text{Math.min}(15\%, 10\%)$

Fizetendő: 450 Ft

Visszaadott kuponok: []

5. MAX15 és csak egy A5 → nem használjuk ki a maximumot

Kosár: 1 kg alma

Időszak: Normál

Kuponok: A5, A5-MAX15

Ár: 500 Ft

MAX: 15%

A5 → -5%

A5-MAX15 → újabb -5%, még van 5% lehetőség, de nincs más kupon

Fizetendő: 450 Ft

Visszaadott kuponok: []

6. Akció erősebb, kupon nem érvényes

Kosár: 2 kg alma

Időszak: Tavaszi (15% kedvezmény automatikusan)

Kuponok: A5, A5, A5-MAX15

Ár: 1200 Ft

Tavaszi kedvezmény: 15%

Kuponok összesen: 15% → **nem előnyösebb**

Fizetendő: 1020 Ft

Visszaadott kuponok: [A5, A5, A5-MAX15]

7. A5-MAX15 + A5 + A5 → max 15% elérve

Kosár: 1 kg alma

Időszak: Normál

Kuponok: A5-MAX15, A5, A5

Ár: 500 Ft

A5-MAX15, A5, A5 (sorrend mindegy): kupon kedvezmény 15%

Fizetendő: 425 Ft

Visszaadott kuponok: []

8. A5 + A5-MAX15 + A5 → max 15% elérve

Kosár: 1 kg alma

Időszak: Normál

Kuponok: A5, A5-MAX15, A5

Ár: 500 Ft

A5, A5-MAX15, A5, (sorrend mindegy): kupon kedvezmény 15%

Fizetendő: 425 Ft

Visszaadott kuponok: []

9. A5 + A5 + A5-MAX15 → max 15% elérve

Kosár: 1 kg alma

Időszak: Normál

Kuponok: A5, A5, A5-MAX15

Ár: 500 Ft

A5, A5, A5-MAX15 (sorrend mindegy): kupon kedvezmény 15%

Fizetendő: 425 Ft

Visszaadott kuponok: []

Fejlesztési követelmények

A rendszernek az alábbiakat kell támogatnia:

- A MAX kuponokat külön kell kezelni, és összevonhatósági logikával kell dolgozni.
 - A MAX kupon helye a kupon sorrendben mindegy.
 - A rendszernek nyomon kell követnie:
 - **Az adott termékre addig alkalmazott kedvezményt**
 - **Az adott MAX limitet**, és azt nem haladhatja meg
 - Ha egy kupon nem használható (pl. túllépné a MAX-ot), azt **vissza kell adni**.
-

Megjegyzés

A MAX kuponok bevezetése **összetett szabályokat** igényel, és jól demonstrálja a **kuponlogika bonyolultságát**.

Hogy még jobban érthető legyen a feladat, ezért a feladatban szereplő példákat megadjuk unit tesztek formájában is:

```
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 AruhazCR4Tesztek {
    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_cr4_pelda1_EgyMAX() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 2.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A5-MAX10"));
        assertEquals(950.0, ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr4_pelda2_KetA5esMAX10() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A5", "A5",
"A5-MAX10"));
        assertEquals(450.0, ar.getAr(), 0.001);
        assertEquals(List.of("A5"), ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr4_pelda3_MAX15esMAX10() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A5", "A5-
MAX15", "A5-MAX10"));
        assertEquals(450.0, ar.getAr(), 0.001);
        assertEquals(List.of("A5-MAX15"), ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr4_pelda4_MAX15esMAX10() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A5-MAX15",
"A5-MAX10"));
        assertEquals(450.0, ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr4_pelda5_MAX15_es_A5() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A5", "A5-
MAX15"));
        assertEquals(450.0, ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }
}
```

```

    }
    @Test
    void teszt_cr4_pelda6_Akcio_erosebb_mint_kupon() {
        Idoszak tavaszi = new Idoszak("Tavaszi");
        tavaszi.setEgysegAr(Termek.ALMA, 600.0);
        tavaszi.setKedvezmeny(Termek.ALMA, 2.0, 0.15);
        tavaszi.setKedvezmeny(Termek.ALMA, 5.0, 0.20);
        target.addIdoszak(tavaszi);
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 2.0)));
        ArInfo ar = target.getKosarAr(kosar, tavaszi, List.of("A5", "A5",
"A5-MAX15"));
        assertEquals(1020.0, ar.getAr(), 0.001);
        assertEquals(List.of("A5", "A5", "A5-MAX15"),
ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr4_pelda7_MAX15plus2A5() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A5-MAX15",
"A5", "A5"));
        assertEquals(425.0, ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr4_pelda8_A5_MAX15_A5() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A5", "A5-
MAX15", "A5"));
        assertEquals(425.0, ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr4_pelda9_2A5_MAX15() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("A5", "A5",
"A5-MAX15"));
        assertEquals(425.0, ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }
}

```