

CR5 - Több termékre vonatkozó kuponok bevezetése

1. Vicces bevezető

„Az összevonható almakuponok óriási sikert arattak! Ennek láttán a JóÁr áruház marketingosztálya újabb kuponok bevezetése mellett döntött! Itt az ideje az összevonható banánkuponoknak, sőt a több termékre is vonatkozó, úgynevezett X kuponoknak!”

2. Új kuponok

- **B5-MAX10**: banánra 5% kedvezmény, összevonható B* kuponokkal, MAX=10%.
 - **B5-MAX15**: banánra 5% kedvezmény, összevonható B* kuponokkal, MAX=15%.
 - **X5**: minden termékre 5%, nem összevonható más kuponnal, ha részben érvényesül → nem adható vissza.
 - **X10**: minden termékre 10%, nem összevonható más kuponnal, ha részben érvényesül → nem adható vissza.
 - **X5-MAX10**: minden termékre 5%, MAX=10% termékenként, összevonható minden kuponnal, ha részben érvényesül → nem adható vissza.
-

3. Szabályok

- Az X kuponokra minden eddigi szabály érvényes.
 - Új szabály: ha egy X kupon **akár részben** érvényesül, akkor **nem adható vissza**.
 - Nem lehet X kupon helyett A+ B kupont visszaadni (pl. X5 helyett A5+B5), mert azokat külön vissza kéne adni.
 - Külön kivétel: **X5 összevonható A5-MAX10-nel**, mivel az X5 lényegében tartalmaz egy A5 kupont.
-

4. Példák (tesztekből)

1. **B5-MAX10** egyedül → banán 5% kedvezmény.
2. **B5 + B5-MAX10** → banán 10% kedvezmény (MAX=10%).
3. **B5 + B5-MAX15 + B5-MAX10** → legszigorúbb MAX=10%, a nagyobb MAX (15) kupont vissza kell adni.
4. **X5** → minden termék 5%.
5. **X10 + más kuponok** → csak X10 érvényesül, a többi visszajár.
6. **X5 részben érvényesül** (pl. csak alma a kosárban) → X5 marad, más kupon visszajár.
7. **X5 + A5-MAX10** → kivételesen összevonhatók (alma 10%, banán 5%).
8. **X5-MAX10** egyedül → minden termék 5%, MAX=10%.
9. **X5-MAX10 + A5 + B5** → mindkét termék 10%-ig kedvezményes, nincs visszaadás.
10. **X5-MAX10 + A5 + A5** → alma oldal túllépné a MAX=10%-ot, egy A5 visszajár.
11. **X5-MAX10 + B5-MAX10 + B5** → banán 10%, nincs visszaadás.

12. **X5 részleges alkalmazás** (csak banán van, mellé A5) → banánra 5% érvényesül, A5 visszajár, X5 marad.

5. Fejlesztési feladat

„Az Ön feladata, hogy az eddigi megoldást a CR5-nek megfelelően módosítsa!

A következő alfejezet a CR6. A CR6-hoz csak akkor kezdjen hozzá, ha az összes előző CR teljesen kész! Javasoljuk, hogy használja az alfejezet végén lévő egységteszteket, illetve a hangulatvezérelt kódolás eszközeit a feladat megoldásához!”

6. Unit tesztek

```
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 AruhazCR5Tesztek {
    static Aruhaz target;
    static Idoszak normal;

    @BeforeAll
    public static void initAruhaz() {
        target = new Aruhaz();
        normal = new Idoszak("Normál");
        // CR0/CR1-ből ismert Normál időszak beállításai
        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);
    }

    // --- Segéd: 5 Ft-os kerekítés (CR0 szabály)
    private double kerekites5re(double osszeg) {
        double maradek = osszeg % 10.0;
        double egeszResz = Math.floor(osszeg / 10.0) * 10.0;
        if (maradek < 2.5) return egeszResz;
        if (maradek < 5.0) return egeszResz + 5.0;
        if (maradek < 7.5) return egeszResz + 5.0;
        return egeszResz + 10.0;
    }

    // 1) B5-MAX10 egyedül banánra -> 5% kedvezmény (mint A5-MAX10
    // analógja)
    @Test
    void teszt_cr5_pelda1_b5max10_egyedul() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 1.0))); //
        // nincs mennyiségi akció
        // Bruttó: 450; 5% kedvezmény = 427.5 -> kerekítve 430
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("B5-MAX10"));
```

```

        assertEquals(kerekites5re(450.0 * 0.95), ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }

    // 2) B5 + B5-MAX10 -> 10% banán kedvezmény (MAX 10% elérve)
    @Test
    void teszt_cr5_pelda2_b5_es_b5max10_max10_elerve() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 1.0)));
        // 450 * 0.90 = 405 -> kerekítve 405
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("B5", "B5-MAX10"));
        assertEquals(kerekites5re(450.0 * 0.90), ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }

    // 3) B5 + B5-MAX15 + B5-MAX10 -> legszigorúbb MAX (10%), a "nagyobb MAX" kupont adjuk vissza
    @Test
    void teszt_cr5_pelda3_ket_max_kozul_10_a_szigorubb_es_max15_visszajar() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 1.0)));
        // MAX10 és MAX15 együtt -> MAX=10%; szükséges kedvezmény: 10% -> 450*0.90
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("B5", "B5-MAX15", "B5-MAX10"));
        assertEquals(kerekites5re(450.0 * 0.90), ar.getAr(), 0.001);
        // A logika CR4 mintájára: a "értékesebb" (nagyobb MAX) kupont adjuk vissza
        assertEquals(List.of("B5-MAX15"), ar.getFelNemHasznaltKuponok());
    }

    // 4) X5 egyedül (globális 5%) - két termék a kosárban
    @Test
    void teszt_cr5_pelda4_x5_egyedul_ket_termek() {
        Kosar kosar = new Kosar(List.of(
            new Tetel(Termek.ALMA, 1.0),
            new Tetel(Termek.BANAN, 1.0)
        ));
        // Összesen: 500 + 450 = 950; 5% kedvezmény: 902.5 -> kerekítve 905
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("X5"));
        assertEquals(kerekites5re(950.0 * 0.95), ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }

    // 5) X10 + (más kuponok) -> X10 nem összevonható, X10 alkalmazódik, a többiek visszajárnak
    @Test
    void teszt_cr5_pelda5_x10_egyeb_kuponokkal_nem_osszevonhato() {
        Kosar kosar = new Kosar(List.of(
            new Tetel(Termek.ALMA, 2.0), // 1000 Ft (normál: 2 kg -> még nincs almára akció)
            new Tetel(Termek.BANAN, 1.0) // 450 Ft
        ));
        // Összesen: 1450; X10 -> 1305 -> kerekítve 1305
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("X10", "A5", "B5", "A5-MAX10", "B5-MAX10"));
        assertEquals(kerekites5re(1450.0 * 0.90), ar.getAr(), 0.001);
        // Minden más kupon visszajár, X10 nem (alkalmazódott)
        assertEquals(List.of("A5", "B5", "A5-MAX10", "B5-MAX10"), ar.getFelNemHasznaltKuponok());
    }
}

```

```

// 6) X5 már részben érvényesül -> nem adható vissza
// Kosárban csak alma van; X5 alkalmazódik almára, a B5 nem (nincs
banán) -> B5 visszajár, X5 marad
@Test
void teszt_cr5_pelda6_x5_reszben_ervenyesul_nem_adhato_vissza() {
    Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
    // Alma 500; X5 -> 475 -> kerekítve 475
    ArInfo ar = target.getKosarAr(kosar, normal, List.of("X5", "B5"));
    assertEquals(kerekites5re(500.0 * 0.95), ar.getAr(), 0.001);
    assertEquals(List.of("B5"), ar.getFelNemHasznaltKuponok());
}

// 7) Kivétel: X5 összevonható A5-MAX10-nel (X5 "tartalmaz" A5-öt) ->
alma 10%, banán 5%
@Test
void teszt_cr5_pelda7_x5_es_a5max10_osszevonhato_kivetel() {
    Kosar kosar = new Kosar(List.of(
        new Tetel(Termek.ALMA, 1.0),
        new Tetel(Termek.BANAN, 1.0)
    ));
    // Alma:  $500 * (1 - 0.05 \text{ [X5-as A5]} - 0.05 \text{ [A5-MAX10]}) = 500 * 0.90$ 
= 450
    // Banán:  $450 * (1 - 0.05 \text{ [X5-as B5]}) = 427.5$ 
    // Összesen: 877.5 -> kerekítve 880
    ArInfo ar = target.getKosarAr(kosar, normal, List.of("X5", "A5-
MAX10"));
    double expected = kerekites5re(500.0 * 0.90 + 450.0 * 0.95);
    assertEquals(expected, ar.getAr(), 0.001);
    assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
}

// 8) X5-MAX10 egyedül (globális 5%, MAX=10% termékenként) - két termék
@Test
void teszt_cr5_pelda8_x5max10_egyedul_ket_termek() {
    Kosar kosar = new Kosar(List.of(
        new Tetel(Termek.ALMA, 1.0),
        new Tetel(Termek.BANAN, 1.0)
    ));
    // Mindkettőre -5%:  $500*0.95 + 450*0.95 = 902.5$  -> 905
    ArInfo ar = target.getKosarAr(kosar, normal, List.of("X5-MAX10"));
    assertEquals(kerekites5re(500.0 * 0.95 + 450.0 * 0.95), ar.getAr(),
0.001);
    assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
}

// 9) X5-MAX10 + A5 + B5 -> mindkét termék 10%-ig mehet, nem lépjük túl
-> nincs visszaadás
@Test
void teszt_cr5_pelda9_x5max10_a5_b5_mindketto_10szazalekig() {
    Kosar kosar = new Kosar(List.of(
        new Tetel(Termek.ALMA, 1.0),
        new Tetel(Termek.BANAN, 1.0)
    ));
    // Alma: X5-MAX10 (5%) + A5 (5%) -> 10% összesen =>  $500*0.90 = 450$ 
    // Banán: X5-MAX10 (5%) + B5 (5%) -> 10% összesen =>  $450*0.90 = 405$ 
    // Összesen: 855 -> kerekítve 855
    ArInfo ar = target.getKosarAr(kosar, normal, List.of("X5-MAX10",
"A5", "B5"));
    assertEquals(kerekites5re(500.0 * 0.90 + 450.0 * 0.90), ar.getAr(),
0.001);
}

```

```

        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }

    // 10) X5-MAX10 + A5 + A5 -> alma oldalon túllépnénk a 10%-ot, egy A5
    visszajár
    @Test
    void teszt_cr5_pelda10_x5max10_ket_a5_eset_visszaadas() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        // X5-MAX10 (5%) + A5 (5%) => 10% elérve -> második A5 már nem fér
        bele -> visszajár
        // Ár: 500*0.90 = 450 -> kerekítve 450
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("X5-MAX10",
        "A5", "A5"));
        assertEquals(kerekites5re(500.0 * 0.90), ar.getAr(), 0.001);
        assertEquals(List.of("A5"), ar.getFelNemHasznaltKuponok());
    }

    // 11) X5-MAX10 + B5-MAX10 + B5, csak banán a kosárban -> 10%-ig mehet,
    nincs visszaadás
    @Test
    void teszt_cr5_pelda11_x5max10_b_oldali_max_kombinalas() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 1.0)));
        // B oldal: X5-MAX10 (5%) + B5 (5%) -> MAX10 teljesítve => 450*0.90
        = 405
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("X5-MAX10",
        "B5-MAX10", "B5"));
        assertEquals(kerekites5re(450.0 * 0.90), ar.getAr(), 0.001);
        assertEquals(List.of(), ar.getFelNemHasznaltKuponok());
    }

    // 12) X5 jelen van, de csak részlegesen alkalmazható (csak banán a
    kosárban),
    // mellé A5 kupon is szerepel -> X5 érvényesül banánra (mint B5), A5
    nem (nincs alma), A5 visszajár, X5 nem.
    @Test
    void teszt_cr5_pelda12_x5_reszleges_alkalmazas_es_visszaadas_szabaly()
    {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 1.0)));
        // 450*0.95 = 427.5 -> 430
        ArInfo ar = target.getKosarAr(kosar, normal, List.of("X5", "A5"));
        assertEquals(kerekites5re(450.0 * 0.95), ar.getAr(), 0.001);
        assertEquals(List.of("A5"), ar.getFelNemHasznaltKuponok());
    }
}

```