CR2 – Kuponok kezelése

A JóÁr áruház megállapodott a JóKupon kupongyártó céggel, hogy kuponokkal bővíti marketingstratégiáját. A cél az, hogy a vásárlók egyedi kedvezményeket vehessenek igénybe, ezzel növelve a márkahűséget és az eladásokat. A rendszernek mostantól képesnek kell lennie **különböző típusú kuponok** feldolgozására is.

Metódus módosítás

A `getKosárÁr` metódus szignatúrája mostantól három paramétert fogad:

```
ÁrInfo getKosárÁr(Kosár kosár, Időszak időszak, List<String> kuponok)
```

A visszatérési típus most már nem `double`, hanem egy új típus, `ÁrInfo`, amely tartalmazza:

- a **fizetendő összeget** (lebegőpontos szám, forintban),
- valamint a **fel nem használt kuponok listáját**.

Kupon típusok

Az első körben a rendszernek az alábbi **alap kuponokat** kell tudnia kezelni:

Termékkedvezmény kuponok:

- `A5`: 5% kedvezmény az alma teljes árából. Nem összevonható más A* kuponnal.
- `A10`: 10% kedvezmény az alma teljes árából. Nem összevonható más A* kuponnal.
- `B5`: 5% kedvezmény a banán teljes árából. Nem összevonható más B* kuponnal.
- `B10`: 10% kedvezmény a banán teljes árából. Nem összevonható más B* kuponnal.

Ingyen mennyiség kuponok:

- `A-FREE1`: maximum 1 kg alma ingyenes (akkor is érvényes, ha kevesebb, mint 1 kg almát veszünk, ekkor az alma mennyiségét 0-ra csökkenti). Nem összevonható más A* kuponnal, de mennyiségi kedvezményt nem zár ki, illetve más termékre vonatkozó kupont sem zár ki.
- `B-FREE1`: maximum 1 kg banán ingyenes (akkor is érvényes, ha kevesebb, mint 1 kg banánt veszünk, ekkor a banán mennyiségét 0-ra csökkenti). Nem összevonható más B* kuponnal, de mennyiségi kedvezményt nem zár ki, illetve más termékre vonatkozó kupont sem zár ki.

Ezek a kuponok **egyszer használhatóak**. Nem összevonhatók más **azonos termékre vonatkozó kuponokkal**, de időszakos kedvezményekkel kombinálhatóak. A rendszernek a kuponokat **a megadott sorrendben** kell feldolgoznia.

Kuponhasználat szabályai

- A rendszer **sorrendben halad** a beadott kuponlistán.
- Ha egy kupon **nem alkalmazható** (pl. nem lenne kedvezőbb, mint egy meglévő akció, lásd eddigi mennyiségi kedvezmények, vagy már alkalmaztunk egy másik, nem összevonható kupont), akkor azt **nem használja fel**.
- A fel nem használt kuponokat **vissza kell adni** a vásárlónak (ezek kerülnek bele az `ÁrInfo` típus megfelelő listájába).
- **Nem összevonható kuponok esetén** csak az első alkalmazható kupont szabad figyelembe venni az adott termékre, más termékek kuponjaival összevonható.
- **Ha az érvényes mennyiségi kedvezmény nagyobb, vagy ugyanakkor kedvezményt ad, mint amit a feldolgozás alatt álló kupon adna, akkor azt a kupont vissza kell adni.**
- **A FREE kuponok nem zárják ki más kedvezmény adását, hiszen, bár a FREE kuponok nem összevonhatók, de ez csak más kuponokra vonatkozik, az időszakos akciókkal együtt adható.**
- **A FREE kuponok csökkentik a termék mennyiség, ami után fizetnünk kell; ha ez a csökkentett mennyiség még elég nagy, hogy mennyiségi kedvezmény járjon rá, akkor jár a mennyiségi kedvezmény.**

Példák

1. Egyszerű kuponhasználat:

Kosár: 2 kg banán
Időszak: Normál
Kuponok: `B10`

- Normál esetben 2 kg banán → 900 Ft
- Normál akció szerint 2 kg banán után **10% kedvezmény** jár
- A kupon `B10` is 10%-ot adna, de **nem előnyösebb**, ezért **nem használható fel**
- **Fizetendő:** 810 Ft → **5 forintra kerekítve → 810 Ft**
- **Visszaadott kuponok:** [`B10`]

---

2. Érvényes kupon:

Kosár: 1 kg alma
Időszak: Normál
Kuponok: `A10`

- Alma ára: 500 Ft
- Nincs akció
- `A10` kupont fel lehet használni → **10% kedvezmény**
- **Fizetendő:** 450 Ft → **5 forintra kerekítve → 450 Ft**
- **Visszaadott kuponok:** `[]`

---

3. Kupon érvénytelen sorrend miatt:

Kosár: 1 kg alma
Időszak: Normál
Kuponok: `A5`, `A10`

- `A5` felhasználható → 5% kedvezmény
- Ezután `A10` **már nem alkalmazható**, mert nem összevonható
- **Fizetendő:** 475 Ft → **5 forintra kerekítve → 475 Ft**
- **Visszaadott kuponok:** [`A10`]

---

4. Jobb sorrend, jobb eredmény:

Kosár: 1 kg alma
Időszak: Normál
Kuponok: `A10`, `A5`

- `A10` felhasználható → 10% kedvezmény
- `A5` **már nem alkalmazható**
- **Fizetendő:** 450 Ft → **5 forintra kerekítve → 450 Ft**
- **Visszaadott kuponok:** [`A5`]

---

5. Ingyen termék kupon:

Kosár: 0.5 kg alma
Időszak: Normál
Kuponok: `A-FREE1`

- 0.5 kg alma mennyiségét csökkenti 0 kg almára
- Ár: 0 Ft → **5 forintra kerekítve → 0 Ft**
- **Visszaadott kuponok:** `[]`

---

6. Ingyen termék kupon mennyiségi kedvezménnyel kombinálva:

Kosár: 3 kg banán
Időszak: Normál
Kuponok: `B-FREE1`

- 1 kg ingyen (kupon alapján), marad 2 kg fizetendő
- 2 kg-ra **jár a 10% mennyiségi kedvezmény** (normál időszak)
- 2 kg banán ára: 900 Ft → kedvezménnyel 810 Ft
- **Fizetendő:** 810 Ft → **5 forintra kerekítve → 810 Ft**
- **Visszaadott kuponok:** `[]`

Megjegyzés: Bár a kupon **nem összevonható**, ez csak más kuponokra vonatkozik – **az időszakos akciókkal együtt érvényes** lehet.

---

7. Kuponok külön termékre – mindkét kupon érvényes:

Kosár: 1 kg alma, 1 kg banán
Időszak: Normál
Kuponok: `A5`, `B5`

- `A5` → alma ára 500 Ft → 5% kedvezmény → 475 Ft
- B5 alkalmazható, mert másik termékre vonatkozik, mint az A5.
- `B5` → banán ára 450 Ft → 5% kedvezmény → 427,5 Ft
- **Fizetendő:** 475 + 427,5 → **902,5 Ft → 5 forintra kerekítve → 905 Ft**
- **Visszaadott kuponok:** []

---

8. Ingyen kupon kiszorít jobb százalékos kupont

Kosár: 1 kg alma
Időszak: Normál
Kuponok: A10, A-FREE1

Lépések:

- A10: adna 10% kedvezményt → 500 Ft → 450 Ft
- A-FREE1: lenullázza az alma árát → 0 Ft, de nem összevonható
- A10 az első, felhasználjuk → A-FREE1 már **nem alkalmazható**

Fizetendő: 450 Ft
Visszaadott kuponok: [A-FREE1]

---

9. Ingyen kupon előbb, drágább végül

Kosár: 1 kg alma
Időszak: Normál
Kuponok: A-FREE1, A10

Lépések:

- A-FREE1 lenullázza az alma árát → 0 Ft
- A10 nem alkalmazható, mert nem összevonható

Fizetendő: 0 Ft
Visszaadott kuponok: [A10]

---

10. Ingyen kupon kevesebb mennyiségre, majd maradékra százalékos kupon – nem megengedett

Kosár: 1.5 kg banán
Időszak: Normál
Kuponok: B-FREE1, B10

Lépések:

- B-FREE1 → 1 kg ingyenes → marad 0.5 kg
- B10 **nem alkalmazható**, mert már volt egy B* kupon

Fizetendő: 0.5 kg × 450 Ft = 225 Ft
→ nincs akció 0.5 kg-ra
Fizetendő: 225 Ft
Visszaadott kuponok: [B10]

---

11. Kupon nem vonható össze akcióval (rosszabb feltétel miatt)

Kosár: 3 kg banán
Időszak: Normál
Kuponok: B5

Lépések:

- Normál időszakban 10% kedvezmény jár 3 kg banánra → 1350 Ft → 1215 Ft
- B5 csak 5% kedvezményt adna → nem előnyösebb
  Fizetendő: 1215 Ft → kerekítve: 1215 Ft
  Visszaadott kuponok: [B5]

---

12. Ingyenes mennyiség nem jár akció, mert mennyiség levonása után kiesik

Kosár: 2.2 kg alma
Időszak: Normál
Kuponok: A-FREE1

Lépések:

- A-FREE1 levon 1 kg → marad 1.2 kg
- Alma 2 kg-tól kaphat kedvezményt → most már **nem jár**
- Ár: 1.2 kg × 500 Ft = 600 Ft
  Fizetendő: 600 Ft
  Visszaadott kuponok: []

---

13. Ugyanarra a termékre több kupon különböző típusból – csak az első érvényes

Kosár: 1 kg banán
Időszak: Normál
Kuponok: B5, B-FREE1

Lépések:

- B5 felhasználható → 5% kedvezmény → 427.5 Ft
- B-FREE1 **nem alkalmazható** (B* kupont már használtunk)
  Fizetendő: 427.5 Ft → kerekítve: 430 Ft
  Visszaadott kuponok: [B-FREE1]

---

14. Olyan kupon, amely nem vonatkozik a kosárban lévő termékre

Kosár: 1 kg alma
Időszak: Normál
Kuponok: B10

Lépések:

- B10 nem vonatkozik almára
  Fizetendő: 500 Ft
  Visszaadott kuponok: [B10]

Fejlesztési követelmények

- A `getKosárÁr` metódus működését a kuponok befolyásolják.
- A kupon típusokat könnyen **bővíthető struktúrában** kell kezelni (pl. osztályhierarchia, típusazonosító alapján értelmezhető szabály).
- A rendszer minden kuponnál dönt:
  - felhasználja → alkalmazza a kedvezményt,
  - nem tudja felhasználni → visszaadja.

Megjegyzés

A fenti példák csak néhány gyakori kuponfajtát mutatnak be. A JóÁr áruház és a JóKupon cég együttműködése során **számos új kuponforma megjelenése várható** a jövőben – ezek kezeléséhez a rendszernek rugalmasnak kell lennie.

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

Hogy még pontosabb legyen a követelmény, megadjuk a példák alapján készült unit teszteket 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 AruhazCR2Tesztek {
    static Aruhaz target;
    static Idoszak normal;
    @BeforeAll
    static void init() {
        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_cr2_pelda1_b10_nem_alkalmazhato() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 2.0)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("B10"));
        assertEquals(810.0, info.getAr(), 0.001);
        assertEquals(List.of("B10"), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda2_a10_alkalmazhato() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("A10"));
        assertEquals(450.0, info.getAr(), 0.001);
        assertEquals(List.of(), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda3_rossz_sorrend() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("A5",
"A10"));
        assertEquals(475.0, info.getAr(), 0.001);
        assertEquals(List.of("A10"), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda4_jo_sorrend() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("A10",
"A5"));
        assertEquals(450.0, info.getAr(), 0.001);
        assertEquals(List.of("A5"), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda5_ingyen_alma() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 0.5)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("A-FREE1"));
        assertEquals(0.0, info.getAr(), 0.001);
        assertEquals(List.of(), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda6_free_es_mennyisegi() {
```

```
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 3.0)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("B-FREE1"));
        assertEquals(810.0, info.getAr(), 0.001);
        assertEquals(List.of(), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda7_ket_kulon_kupon() {
        Kosar kosar = new Kosar(List.of(
                new Tetel(Termek.ALMA, 1.0),
                new Tetel(Termek.BANAN, 1.0)
        ));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("A5",
"B5"));
        assertEquals(905.0, info.getAr(), 0.001);
        assertEquals(List.of(), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda8_free_kimarad_jo_kupon_marad() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("A10", "A-
FREE1"));
        assertEquals(450.0, info.getAr(), 0.001);
        assertEquals(List.of("A-FREE1"), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda9_free_elsokent_nem_enged_kupon_osszevonast() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("A-FREE1",
"A10"));
        assertEquals(0.0, info.getAr(), 0.001);
        assertEquals(List.of("A10"), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda10_free_utan_nem_alkalmazhato_szazalekos() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 1.5)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("B-FREE1",
"B10"));
        double expected = kerekites5re(0.5 * 450.0);  // nincs akció
        assertEquals(expected, info.getAr(), 0.001);
        assertEquals(List.of("B10"), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda11_kupon_nem_elozo_akcionak_jobbat() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 3.0)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("B5"));  //
már jár 10% akció
        double expected = kerekites5re(3.0 * 450.0 * 0.9);  // 10% akció
        assertEquals(expected, info.getAr(), 0.001);
        assertEquals(List.of("B5"), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda12_free_kiesik_akciobol() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 2.2)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("A-FREE1"));
        // 2.2 - 1.0 = 1.2 kg alma, nem kap mennyiségi kedvezményt
        double expected = kerekites5re(1.2 * 500.0);
        assertEquals(expected, info.getAr(), 0.001);
        assertEquals(List.of(), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda13_ket_kupon_egy_termekre() {
```

```
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.BANAN, 1.0)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("B5", "B-
FREE1"));
        double expected = kerekites5re(450.0 * 0.95);
        assertEquals(expected, info.getAr(), 0.001);
        assertEquals(List.of("B-FREE1"), info.getFelNemHasznaltKuponok());
    }
    @Test
    void teszt_cr2_pelda14_rossz_termek_kupon() {
        Kosar kosar = new Kosar(List.of(new Tetel(Termek.ALMA, 1.0)));
        ArInfo info = target.getKosarAr(kosar, normal, List.of("B10"));
        assertEquals(500.0, info.getAr(), 0.001);
        assertEquals(List.of("B10"), info.getFelNemHasznaltKuponok());
    }
    // Segédfüggvény a 0.5-re kerekítéshez - CR1 alapján
    private double kerekites5re(double osszeg) {
        double maradek = osszeg % 10.0;
        if (maradek < 2.5) return osszeg - maradek;
        if (maradek < 5.0) return osszeg - maradek + 5.0;
        if (maradek < 7.5) return osszeg - maradek + 5.0;
        return osszeg - maradek + 10.0;
    }
}
```