

## Kvízinstrukciók



### 1. kérdés

1 pont

Az alábbiak közül melyik adatszerkezetekben tehető meg az adatszerkezet elrontása nélkül, hogy a benne levő elemek tartalmát módosítsuk (anélkül, hogy erről értesítenénk az adatszerkezetet)?

- ☐ `HashSet<String>`
- ☒ `ArrayList<Integer>`
- ☒ `int[]`
- ☒ `HashMap<String, Integer>`



### 2. kérdés

1 pont

Mikor lehet `a == "xyz"` eredménye `true`?

- ☐ ha `a` forráskódba leírt literálból kap értéket
- ☐ ha `a` futási időben kap értéket
- ☐ soha
- ☒ ha `a` tartalma az `xyz` szöveg
- ☐ ha `a` osztályszintű változó, és kap kezdeti értéket



### 3. kérdés

1 pont

Helyes lehet-e a következő vizsgálat: `assertEquals(List.of(1, 2, 3), result)`, ahol `result` típusa `LinkedList<Integer>`, és sorban az `1`, `2`, `3` értékeket tartalmazza?

- ☐ nem, mert a tartalmazott elemek típusa különbözik

- ☐ nem, mert a vizsgált gyűjtemények típusa különbözik
- ☐ nem, mert az `assertEquals` nem megfelelően működik ezekre a típusokra
- ☒ igen



#### 4. kérdés

1 pont

Mikor lehet `a == 123` eredménye `true`, ahol `a` típusa `Integer`?

- ☐ ha `a` futási időben kap értéket
- ☐ ha `a` forráskódba leírt literálból kap értéket
- ☐ soha
- ☒ ha `a` értéke a `123` szám
- ☐ ha `a` osztályszintű változó, és kap kezdeti értéket



#### 5. kérdés

1 pont

Helyes lehet-e a következő vizsgálat: `assertEquals(new int[] { 1, 2, 3 }, result)`, ahol `result` típusa `int` elemeket tartalmazó tömb, és sorban az `1`, `2`, `3` értékeket tartalmazza?

- ☒ igen
- ☐ nem, mert az `assertEquals` nem megfelelően működik ezekre a típusokra
- ☐ nem, mert a vizsgált gyűjtemények típusa különbözik
- ☐ nem, mert a tartalmazott elemek típusa különbözik

Nincs menthető új adat. Utolsó ellenőrzés ekkor: 09:00

Kvíz beadása