Questions

Bookmark this page

equals-ish

1/1 point (graded)

Consider the latest implementation of `Duration` in the reading, reprinted here for convenience:

```
public class Duration {
    private final int mins;
    private final int secs;
    // rep invariant:
    //    mins >= 0, secs >= 0
    // abstraction function:
    //    represents a span of time of mins minutes and secs seconds

    /** Make a duration lasting for m minutes and s seconds. */
    public Duration(int m, int s) {
        mins = m; secs = s;
    }
    /** @return length of this duration in seconds */
    public long getLength() {
        return mins*60 + secs;
    }

    private static final int CLOCK_SKEW = 5; // seconds

    @Override
    public boolean equals (Object thatObject) {
        if (!(thatObject instanceof Duration)) return false;
        Duration thatDuration = (Duration) thatObject;
        return Math.abs(this.getLength() - thatDuration.getLength()) <= CLOCK_SKEW;
    }
}
```

Suppose these Duration objects are created:

```
Duration d_0_60 = new Duration(0, 60);
Duration d_1_00 = new Duration(1, 0);
Duration d_0_57 = new Duration(0, 57);
Duration d_1_03 = new Duration(1, 3);
```

Which of the following expressions return true? Check all that apply.

- [x] d_0_60.equals(d_1_00)
- [x] d_1_00.equals(d_0_60)
- [x] d_1_00.equals(d_1_00)
- [x] d_0_57.equals(d_1_00)
- [ ] d_0_57.equals(d_1_03)
- [x] d_0_60.equals(d_1_03)

✔

Explanation

The equals method compares the total lengths of the intervals in seconds, and allows them to differ by up to 5 seconds (CLOCK_SKEW) while still testing equal. So all the objects are equal to each other except for `d_0_57` and `d_1_03`, which differ by 6 seconds.

Submit | Show Answer

Answers are displayed within the problem

skewed up

1/1 point (graded)

Which properties of an equivalence relation are violated by this equals() method? Ignore null references. Check all that apply.

- [ ] recursivity
- [ ] reflexivity
- [ ] sensitivity
- [ ] symmetry
- [x] transitivity

✔

Explanation
This equals() violates transitivity: `d_0_57` equals `d_1_00`, and `d_1_00` equals `d_1_03`, but `d_0_57` does not equal `d_1_03`.

Submit | Show Answer

Answers are displayed within the problem

buggy equality

1/1 point (graded)

Suppose you want to show that an equality operation is buggy because it isn't reflexive. How many objects do you need for a counterexample to reflexivity?

- ○ none
- ◉ 1 object
- ○ 2 objects
- ○ 3 objects
- ○ all the objects in the type

✔

Explanation
If you show that `x.equals(x)` returns `false` for some particular object `x`, then you have a counterexample to reflexivity.

Submit | Show Answer

Answers are displayed within the problem

Previous Next