Questions

Bookmark this page

structural induction

1/1 point (graded)

Recall this data type from the first exercise in this reading:

```
/** Represents an immutable right triangle. */
class RightTriangle {
    private double[] sides;

    // sides[0] and sides[1] are the two legs,
    // and sides[2] is the hypotenuse, so declare it to avoid having a
    // magic number in the code:
    public static final int HYPOTENUSE = 2;

    /** Make a right triangle.
     * @param legA, legB  the two legs of the triangle
     * @param hypotenuse    the hypotenuse of the triangle.
     *        Requires hypotenuse^2 = legA^2 + legB^2
     *           (within the error tolerance of double arithmetic)
     */
    public RightTriangle(double legA, double legB, double hypotenuse) {
        this.sides = new double[] { legA, legB, hypotenuse };
    }

    /** Get all the sides of the triangle.
     *  @return three-element array with the triangle's side lengths
     */
    public double[] getAllSides() {
        return sides;
    }

    /** @return length of the triangle's hypotenuse */
    public double getHypotenuse() {
        return sides[HYPOTENUSE];
    }

    /** @param factor to multiply the sides by
     *  @return a triangle made from this triangle by
     *  multiplies all side lengths by factor.
     */
    public RightTriangle scale(double factor) {
        return new RightTriangle (sides[0]*factor, sides[1]*factor, sides[2]*factor);
    }

    /** @return a regular triangle made from this triangle.
     *  A regular right triangle is one in which
     *  both legs have the same length.
     */
    public RightTriangle regularize() {
        double bigLeg = Math.max(side[0], side[1]);
        return new RightTriangle (bigLeg, bigLeg, side[2]);
    }

}
```

This datatype has an important invariant: the relationship between the legs and hypotenuse, as stated in the Pythagorean theorem.

Assuming the client obeys the contracts when using RightTriangle, which of the following statements are true about this invariant? Check all that apply.

- [x] The creator RightTriangle() establishes the invariant if the client obeys all contracts.
- [ ] The observer getAllSides() preserves the invariant if the client obeys all contracts.

- [x] The observer getHypotenuse() preserves the invariant if the client obeys all contracts.
- [x] The producer scale() preserves the invariant if the client obeys all contracts.
- [ ] The producer regularize() preserves the invariant if the client obeys all contracts.

✔

Explanation

RightTriangle: yes. If the client obeys the contract of the constructor, particularly its precondition, then the three sides of the triangle stored in the sides array satisfy the Pythagorean theorem invariant, as desired.
getAllSides: no. This method causes a rep exposure, so the client may inadvertently change values in the returned array and destroy the invariant as a result, even while obeying all the specs as written.
getHypotenuse: yes. This method doesn't mutate anything, and doesn't expose the rep, so if the invariant was true before getHypotenuse() is called, then it continues to be true afterward.
scale: yes. This method *does* mutate the triangle, but it scales all the sides by the same amount, so the result is still a right triangle satisfying the Pythagorean theorem.
regularize: no. This method mutates the triangle, but changes only the legs of the triangle and doesn't recalculate a new hypotenuse. This is likely to break the Pythagorean invariant.

Submit

Show Answer

Answers are displayed within the problem

Previous Next