Questions

Bookmark this page

bulking up

1/1 point (graded)

When a specification is strengthened (check all that apply):

- [x] fewer implementations satisfy it
- [ ] more implementations satisfy it
- [ ] fewer clients can use it
- [x] more clients can use it
- [ ] none of the above

✓

Submit

Show Answer

✓ Correct (1/1 point)

strength is truth

1/1 point (graded)

Which of the following can be true about a pair of specifications *A* and *B*? Check all that apply.

- [x] 1. *A* can be stronger than *B* and have a weaker precondition
- [x] 2. *A* can be stronger than *B* and have the same precondition
- [ ] 3. *A* can be stronger than *B* and have a stronger precondition
- [ ] 4. *A* can be stronger than *B* and have an incomparable precondition
- [x] 5. *A* can be incomparable to *B*

✓

Explanation
If *A* has a stronger precondition, either it is weaker than *B*, or there is not a containment relationship (option 3).
If the preconditions are not comparable, then the specs will not be comparable either (4).

Submit

Show Answer

Answers are displayed within the problem

finding find1

1/1 point (graded)

Here are the *find* specifications from the reading:

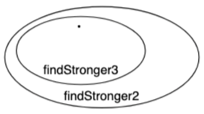
```
static int find1(int[] a, int val)
  requires: val occurs exactly once in a
  effects:  returns index i such that a[i] = val
```

```
static int findStronger2(int[] a, int val)
  requires: val occurs at least once in a
  effects:  returns index i such that a[i] = val
```

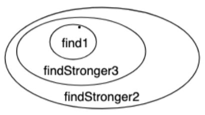
```
static int findStronger3(int[] a, int val)
  requires: val occurs at least once in a
  effects:  returns lowest index i such that a[i] = val
```

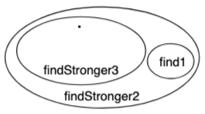
```
static int find4(int[] a, int val)
  requires: nothing
  effects:  returns index i such that a[i] = val,
            or -1 if no such i
```

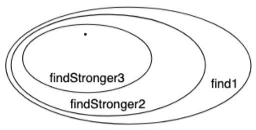
We already know that *findStronger3* is stronger than *findStronger2*, which is stronger than *find1*.

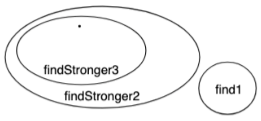


Where is *find1* on the diagram?





[✓] 



✓

Submit

Show Answer

✓ Correct (1/1 point)

finding find4

1/1 point (graded)

Let's determine where *find4* is on the diagram.

How does find1 compare to find4?

- find4 is weaker
- [✓ find4 is stronger]
- find4 and find1 are incomparable

✓

Explanation
find4 has a weaker precondition.
For inputs that satisfy find1's precondition, find4's postcondition is equal.
So find4 is stronger.

Submit

Show Answer

Answers are displayed within the problem

vs. findStronger2

1/1 point (graded)

How does findStronger2 compare to find4?

- find4 is weaker
- [✓ find4 is stronger]
- find4 and findStronger2 are incomparable

✓

Submit

Show Answer

✓ Correct (1/1 point)

vs. findStronger3

0/1 point (graded)

How does findStronger3 compare to find4?

- find4 is weaker
- [✓ find4 is stronger]
- find4 and findStronger3 are incomparable

✗

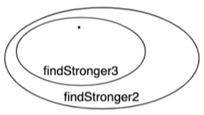
Submit

Show Answer

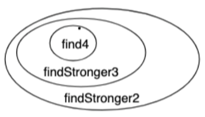
✗ Incorrect (0/1 point)

found

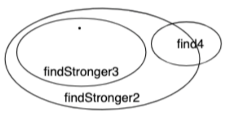
0/1 point (graded)

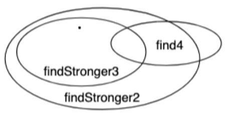


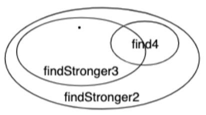
Where is find4 on the diagram?

[✓


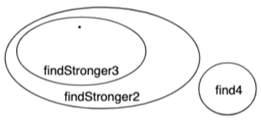
]







✓



✖

Explanation

Since find4 is stronger than findStronger2, it must be contained within that region of the space.
Then the question is its relationship to findStronger3.
There exist implementations that satisfy findStronger3 but not find4: for example, they do not return -1 when `val` is not in `a`, which is excluded by findStronger3's precondition.
There also exist implementations that satisfy find4 but not findStronger3: for example, they do not return the lowest index when `val` occurs multiple times.
And there exist implementations that satisfy both: they can handle the weaker precondition, and the stronger parts of each postcondition.
So find4 overlaps findStronger3, but neither contains the other.

Submit

Show Answer

Answers are displayed within the problem

Previous   Next