Questions | Reading 1: Static Checking | 6.005.1x Courseware

openlearninglibrary.mit.edu/courses/course-v1:MITx+6.005.1x+3T2016/courseware/Readings_Videos/01-Static-Checking

1. Course , current location
2. Progress

Questions

documenting assumptions

1/1 point (graded)

Consider the following simple Python function:

```
from math import sqrt
def funFactAbout(person):
  if sqrt(person.age) == person.age:
    print("The age of " + person.name + " is a perfect square: " +
str(person.age))
```

Which of the following are assumptions made by this code, which must be true in order for it to run without errors?

correct

Explanation

If person is not an object (or if it's `None`), then the code will fail as soon as it tries to refer to `person.age`.

If `person.age` is not a number, or if it's a negative number, then `sqrt()` will fail. But it doesn't necessarily need to be an integer, because `sqrt()` can handle that.

If `person.name` is not a string, then Python will complain of a type error when it tries to concatenate it with other strings. This is one difference between Python and Java -- Python insists that you use a conversion operation like `str()`, whereas Java will automatically convert any type into a `String` when you try to concatenate it with another `String`.

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Answers are displayed within the problem

documenting assumptions, part 2

1/1 point (graded)

If you were writing Java instead of Python, and your Java code needed to make *all* the assumptions below, then which of them could be documented by type declarations and statically checked by the Java compiler?

correct

Explanation

The `person` variable would be declared with some class type, perhaps called `Person`, and the definition of that class would have instance variables `name` and `age` declared with types `String` and `int` respectively.

But we can't use a type declaration to forbid `person` from being `null`. Any object reference might be `null` in Java, just like any variable might be `None` in Python. Similarly, we can't forbid `age` from being negative using a type declaration. These assumptions would have to be documented in comments instead.

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Answers are displayed within the problem