Index Checker for Changing Structures

Want: Index Checker extended for mutable-length data structures

Problem: How to guarantee the lengths of these structures at compilation?

Rodney Olson
UW ID: rolsonjr

Jake Chiang
UW ID: jchiang2

Motivation

- Currently, existing Index Checker is restricted to **fixed-sized data structures** (Strings, Arrays)

```
class String {
    char charAt(@IndexFor("this") index) {
        ...
    }
}
```

- Ideally, an Index Checker could help catch calls outside bounds of **variable-length data structures** (ex: Lists) at compilation

- Limitations for a mutable-length Index Checker involve determining the **uncertain length** of these structures at compilation

Approach

- During compilation, **identify conditionals** that helps the index checker infer the bounds of the list (e.g. `list.isEmpty(), list.size() < x`)

- Also, **identify when lengths of data structures are modified** (add/remove operations) in known ways such as loops, conditional statements, etc.

- Once the index checker has a strong **sense of the bounds** of the data structure, it can identify if indexing by the client is violated at compilation

Challenges

- Understanding how to approach the problem of designing and **implementing new and sound index checker types** that integrates with the existing framework

- Minimize risk of not completing product on schedule by ensuring we **cover basic test cases at first** and not overreaching the scope of the product