



Мутационное тестирование



Agenda

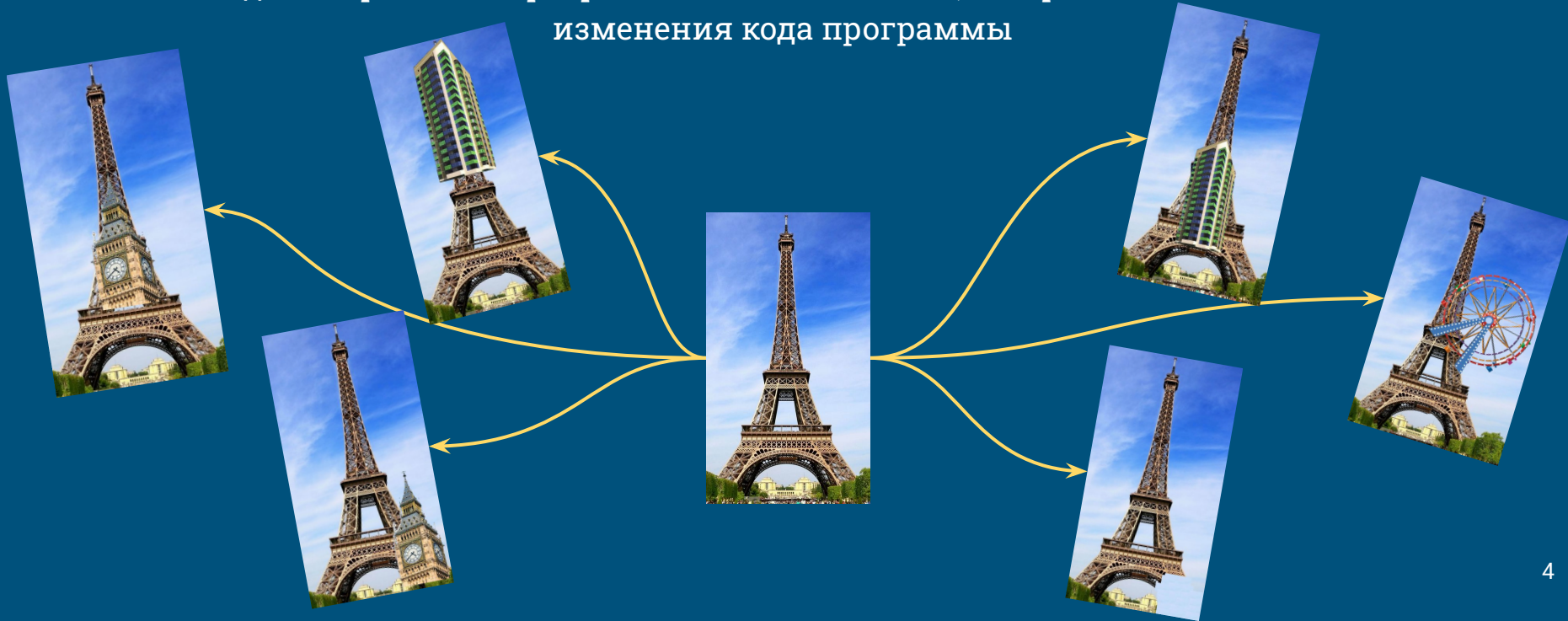
- проблема Code Coverage подхода
- что такое Мутационное тестирование
- предложения по улучшению
- практика

Проблема Code Coverage подхода

Code Coverage — мера, используемая при тестировании программного обеспечения. Она показывает процент исходного кода программы, который был выполнен в процессе тестирования.

Что такое Мутационное тестирование

Это метод тестирования программного обеспечения, который включает небольшие изменения кода программы



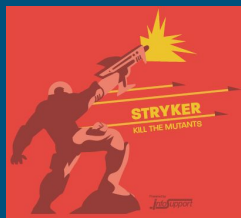
Виды мутаций

- удалить оператор программы
- заменить каждое логическое выражение на логическую константу «истина» или «ложь»
- заменить каждую арифметическую операцию на другую. Например, + на *, - или /
- заменить каждую логическую операцию на другую. Например, > на >=, == или <=
- заменить каждую переменную на другую (из той же области видимости).
Переменные должны иметь одинаковые типы
- ...
- замена объектов на null/nil

Кроме того существуют мутации для объектно-ориентированных языков, операторы для параллельного программирования, операторы для структур данных, таких как контейнеры и др.

Предложения по улучшению

1. Проведение периодического мутационного тестирования
2. Изменение/дополнение уже существующих тестов
3. Рассказать про мутационное тестирование своим коллегам/родителям/друзьям/проходим на улице



Gomutate

The Visual Studio mutation testing tool
VisualMutator



Практика

Идем в GoLand

```
abelov@abelov-book: /media/data/District_M/src/github.com/districtm/post-processing/cmd/post-processor$ go get github.com/zabawaba99/gomutate/cmd/gomutate
abelov@abelov-book: /media/data/District_M/src/github.com/districtm/post-processing/cmd/post-processor$ gomutate ./...
INFO[0000] Generating mutation...                mutation=conditionals_boundary
INFO[0000] Testing mutations...                  mutation=conditionals_boundary
INFO[0015] Result                                killed=true mutant=_gomutate/conditionals_boundary/commit.go.1
INFO[0019] Result                                killed=true mutant=_gomutate/conditionals_boundary/commit.go.2
INFO[0022] Result                                killed=true mutant=_gomutate/conditionals_boundary/commit.go.3
INFO[0026] Result                                killed=true mutant=_gomutate/conditionals_boundary/commit.go.4
INFO[0028] Result                                killed=true mutant=_gomutate/conditionals_boundary/committer.go.1
INFO[0032] Result                                killed=true mutant=_gomutate/conditionals_boundary/committer.go.2
INFO[0034] Result                                killed=true mutant=_gomutate/conditionals_boundary/committer.go.3
INFO[0038] Result                                killed=true mutant=_gomutate/conditionals_boundary/committer.go.4
INFO[0041] Result                                killed=true mutant=_gomutate/conditionals_boundary/merger.go.1
INFO[0045] Result                                killed=true mutant=_gomutate/conditionals_boundary/merger.go.2
INFO[0045] Generating mutation...                mutation=conditionals_boundary
INFO[0045] Testing mutations...                  mutation=conditionals_boundary
```

А что, если после внедрения некоторых мутаций программа станет более полезной и с меньшим количеством баг?



Useful links

1. https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D1%82%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5
2. <https://habr.com/ru/post/334394/>
3. <https://habr.com/ru/post/139337/>
4. http://pitest.org/java_mutation_testing_systems/
5. https://www.youtube.com/watch?v=qGZ-5uHYAi4&list=PLVe-2wcL84b8g_BkXetdmDEKYK3hUGnPw