

АВТОМАТИЧЕСКИЕ МЕТОДЫ МОДИФИКАЦИИ РЕШЕНИЙ ДЛЯ ТЕСТИРОВАНИЯ ПРОВЕРЯЮЩИХ ПРОГРАММ

Студент:

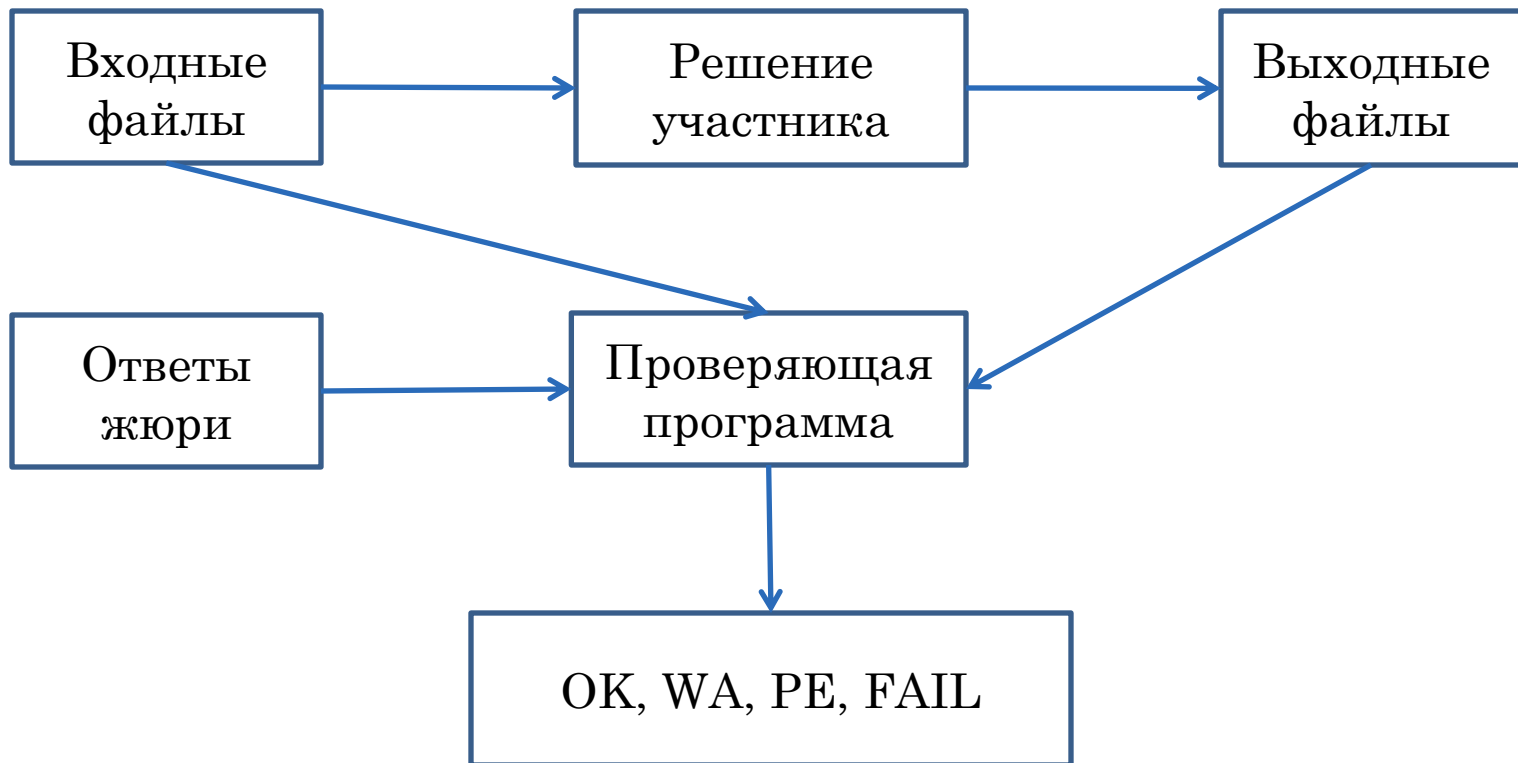
Ахи А.А.

Научный руководитель:

Станкевич А.С.

РЕШАЕМАЯ ПРОБЛЕМА

- Кто проверяет проверяющую программу?



АКТУАЛЬНОСТЬ

- В любых программах часто встречаются ошибки
- Человек не способен обнаружить все ошибки
- Падает качество олимпиады, объективность результатов

РЕШЕНИЕ ПРОБЛЕМЫ

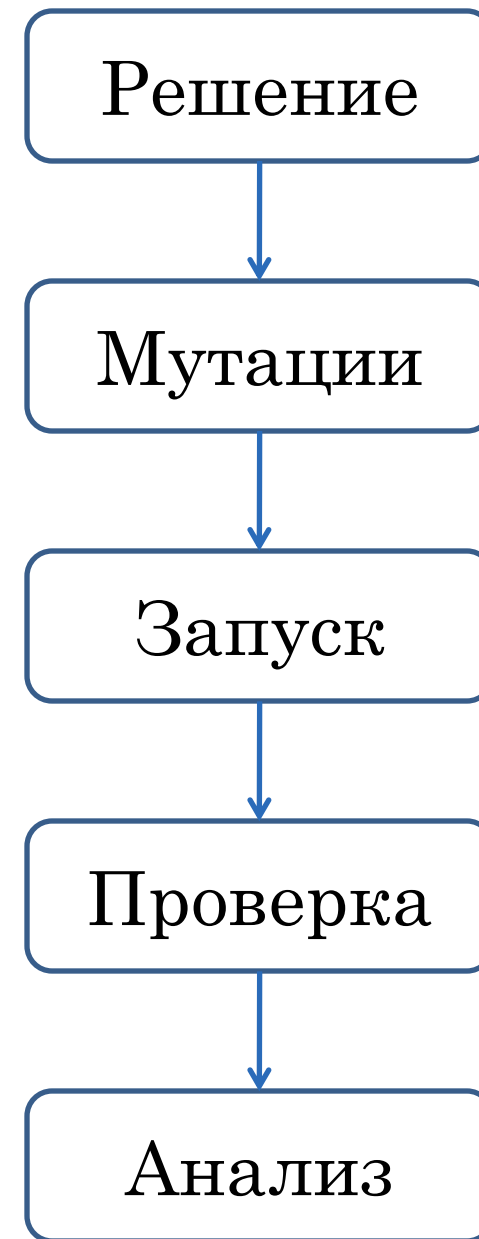
- Ошибки в программах находят с помощью тестирования
- Составление тестов для проверяющих программ
- Набор тестов — набор выходных файлов

ПОСТАНОВКА ЗАДАЧИ

- Разработать метод, позволяющий, имея правильное решение и набор тестов, автоматически выявить ошибки в проверяющей программе

РЕШЕНИЕ ПОСТАВЛЕННОЙ ЗАДАЧИ

- Создание решений с помощью мутационных операторов
- Полученные выходные файлы — тестовый набор



ОПЕРАТОРЫ МУТАЦИИ

- Замена одного оператора другим
- Замена одной переменной другой
- Удаление строки или блока
- Замена условия на `true` или `false`
- Прибавление константы к выражению
- Другие распространенные ошибки программистов

МОДИФИКАЦИЯ РЕШЕНИЯ

- Построение дерева синтаксического разбора исходного кода
- Отдельный класс для модификации каждой структуры языка
- Две функции: число возможных единичных изменений и модификация по номеру

АНАЛИЗ

- Сбор информации о покрытии кода
- Автоматический анализ результата проверки

ПОКРЫТИЕ КОДА

- Используются две метрики:
покрытие строк и покрытие ветвлений
- Модифицируется исходный код проверяющей программы

АНАЛИЗ РЕЗУЛЬТАТА ПРОВЕРКИ

○ Несколько правил:

- Результат на ответ жюри должен быть *OK*
- Результат никогда не должен быть *FAIL*
- Если поменять местами выходные файлы, ответ не должен быть *PE*
- Если ответ содержит сертификат, то при «обратной» проверке результат не должен быть *WA*

АПРОБАЦИЯ

- Были протестированы проверяющие программы с полуфинала чемпионата мира *NEERC2010*

Задача	Время работы (мин.)	Число непокрытых строк проверяющей программы
alignment	9	0 (0) из 12 (4)
cactus	500	0 (0) из 58 (19)
evacuation	700	0 (0) из 36 (12)
factorial	700	0 (0) из 14 (4)
ideal	900	0 (0) из 10 (3)
kgraph	300	0 (0) из 27 (7)

АПРОБАЦИЯ

- Были протестированы проверяющие программы *Russian Code Cup 2011*
- Были выявлены и исправлены ошибки

Задача	Время работы (мин.)	Число непокрытых строк проверяющей программы
guess	20	0 (0) из 53 (15)
birds	3	0 (0) из 40 (7)
queuesort	10	0 (0) из 45 (15)
square	3	0 (0) из 37 (13)

АПРОБАЦИЯ

- Была протестирована проверяющая программа задачи *Quest*
- Были покрыты не все строки

Задача	Время работы (мин.)	Число непокрытых строк проверяющей программы
Quest	60	9 (15) из 280 (92)

Выводы

- Наличие непокрытых строк указывает на подозрительные участки кода
- Достижения полного покрытия не является критерием правильности
- «Тестирование программы может весьма эффективно продемонстрировать наличие ошибок, но безнадежно неадекватно для демонстрации их отсутствия»
Э. Дейкстра

ВЫВОДЫ

- В результате тестирования выявляются ошибки в проверяющих программах
- Компилируются 60% модифицированных решений
- Большое время работы при большом наборе тестов и медленных решениях

РЕЗУЛЬТАТЫ

- Был разработан и реализован метод автоматического тестирования проверяющих программ
- Разработанный метод был успешно апробирован

Спасибо за внимание!

Вопросы?