ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ВЕРИФИКАЦИЯ ПРОГРАММ, ЧАСТЬ 1»

Магистерская программа	230100 «Информатика и вычислительная техника»
Факультет	Физтех-школы ФПМИ, ФРКТ, факультеты ФУПМ, ФИВТ
Кафедра	Системное программирование
Курс	1 курс магистратуры
Семестр	10 семестр, весенний
Оценивание	Дифференцированный зачет
Контактная работа (час.)	40
Самостоятельная работа (час.)	40

1. ЦЕЛИ И ЗАДАЧИ

Курс представляет собой введение в методы верификации программного обеспечения. Цель курса -- познакомить с предметом верификации ПО, представить широкую палитру существующих методов и подходов, осветить преимущества и ограничения, присущие методам верификации. В рамках курса рассматриваются общее понятие качества ПО, подпроцессы обеспечения качества в рамках жизненного цикла ПО, методы статического анализа программ, методы проверки моделей, методы динамического анализа программ и различные варианты функционального тестирования.

Задачами данного курса являются:

- формирование базовых знаний в области обеспечения качества программного обеспечения, как неотъемлемой части теории и практики разработки ПО, адресуемого к проблемам построения корректных и надежных программ, и имеющего важное методологическое значение как для подготовки специалистов в области современных информационных технологий, так и для поддержки разнообразных инновационных сфер деятельности;
- обучение студентов основам жизненного цикла программного обеспечения и задачам верификации, возникающим в ходе разработки, внедрения и эксплуатации ПО;
- обучение студентов методам функционального тестирования, применяемым в различных сценариях разработки ПО, включая модульное тестирование, случайное тестирование, тестирование с использованием моделей, а также методам оценки полноты тестирования;
- обучение студентов базовым методам анализа корректности программ;
- формирование теоретических подходов к верификации программного обеспечения для проведения исследований в рамках выпускных работ на степень магистра.

2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Освоение дисциплины «Верификация программ» направлено на формирование следующих общекультурных и профессиональных компетенций магистра:

а) общекультурные (ОК):

– способность использовать на практике методы и средства анализа наблюдаемого поведения для понимания сущностных явлений окружающего мира (ОК-1);
– способность активно и целенаправленно применять полученные знания, навыки и умения для определения тематики и выполнения индивидуальной научно-исследовательской работы (ОК-2);
– готовность работать с информацией в области современных технологий компьютерной графики и визуализации, используя отечественную и зарубежную научную периодическую литературу, монографии и учебники, электронные ресурсы Интернет (ОК-3).

б) профессиональные (ПК):

– готовность использовать методы и средства верификации программ в последующей профессиональной деятельности в качестве научных сотрудников, преподавателей вузов, инженеров, технологов (ПК-1);
– готовность к решению практических задач по верификации системного и прикладного программного обеспечения (ПК-2);
– готовность выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности, с использованием развитого арсенала методов и средств визуализации (ПК-3);
– готовность к творческому подходу в решении научно-технических задач, основанному на систематическом обновлении полученных знаний, навыков и умений и использовании последних достижений в области верификации программного обеспечения (ПК-4);
– способность применять на практике умения и навыки в организации исследовательских работ и проводить научные исследования, готовность к участию в инновационной деятельности (ПК-5);
– способность представлять результаты собственной деятельности с использованием современных средств, ориентируясь на потребности аудитории, в том числе в форме отчетов, презентаций, докладов (ПК-6);
– способность выполнения проектов и заданий по тематике разрабатываемой научной проблемы (ПК-7);
– способность применять теорию и методы математики и информатики для построения качественных и количественных моделей (ПК-8).

3. Конкретные знания, умения и навыки, формируемые в результате освоения дисциплины

В результате освоения дисциплины «Верификация программ» обучающийся должен:
знать:
- место и роль средств верификации в разработке ПО;
- связь курса верификации со смежными дисциплинами дискретной математики и проектирования программных комплексов;
- методы тестирования и аналитического исследования ПО;
- современные средства и технологии верификации;

уметь:
- эффективно использовать на практике теоретические знания в области верификации программ;
- представить панораму универсальных и специальных методов верификации;
- выбрать методы и сценарии верификации, адекватные предметной области и исследуемой задаче;
- эффективно применять средства верификации для обеспечения качества разрабатываемого программного обеспечения;

владеть:
- современными средствами и технологиями верификации ПО;

- навыками использования систем тестирования для языков высокого уровня;
- навыками использования систем динамического исследования программ;
- навыками использования систем аналитического исследования программ.

4. Структура и содержание дисциплины

Тема 1. Обзор методов верификации.

Понятие верификации программы. Цена ошибки. Методы верификации: экспертиза, статический анализ, дедуктивная верификация, проверка моделей (model checking), статическая верификация, мониторинг, тестирование, синтетические методы.

Тема 2. Понятие качества ПО.

Аспекты качества ПО. Функциональность. Надежность. Производительность. Переносимость. Удобство использования. Удобство сопровождения.

Тема 3. Тестирование. Тестовые покрытия.

Определение тестирования. Методы черного/прозрачного ящика. Функциональное тестирование. Задачи тестирования. Критерии полноты тестирования: на основе структурных элементов целевой системы, на основе структуры входных данных, на основе элементов требований, на основе явно сформулированных предположений об ошибках, на основе произвольных моделей устройства или функционирования целевой системы. Покрытие инструкций, покрытие ветвей, покрытие комбинации условий, покрытие условий, комбинированное покрытие ветвлений и условий, модифицированное покрытие ветвлений и условий MC/DC. Покрытие du-путей. Критерий полноты по модели ошибок. Метрики покрытия на основе входных данных. Метрики для тестирования обработчиков текстов. Метрики покрытия по грамматике. Функциональные критерии покрытия. Критерий полноты по требованиям. Сравнение покрытий. Композиция.

Тема 4. Разработка через тестирование. Рефакторинг.

Выполнения задания по разработке с использованием техники Test-Driven Development (TDD) и рефакторинга. С применением изученных методов верификации.

Тема 5. Статическая верификация программ с использованием проверки моделей.

Понятие модели, извлекаемой из программы. Виды требований. Сценарии взаимодействия с программой. Инструмент CPAchecker и фреймворк для верификации Klever.

Тема 6. Верификация с абстракциями и уточнениями.

Понятие задачи верификации, алгоритма верификации, состояния алгоритма, переходов. Абстракция состояний. Адаптивный алгоритм верификации (Configurable Program Analysis, CPA) — конфигурация проверки моделей (model checking) и анализа потоков данных (data flow analysis), операторы слияния и останова. Уточнение абстракции по контрпримерам (CEGAR). Другие конфигурируемые анализы: Composite, Callstack, Predicate, Value, Block Abstraction Memoization (BAM), Symbolic Memory Graph (SMG) Thread-Modular (TM). Другие алгоритмы: Bounded Model Checking (BMC), k-Induction, IMPACT, Property-directed reachability (PDR).

Тема 7. Динамический анализ программ.

Методы динамического анализа, на примере инструментов Valgrind, AddressSanitizer, ThreadSanitizer. Методы фаззинга, инструменты Syzkaller, libFuzzer. Динамическая символьная интерпретация, фреймворки KLEE, Triton.

Тема 8. Непрерывная интеграция. Встраивание методов верификации.

Выполнение задания по встраиванию в непрерывную интеграцию (CI) методов тестирования со сбором покрытия, статического и динамического анализа.

5. Литература

- [Методы верификации программного обеспечения, В. В. Кулямин] (http://panda.ispras.ru/~kuliamin/docs/VerMethods-2008-ru.pdf)
- [ISO/МЭК 25010 "Модели качества систем и программных продуктов"] (http://ingraf.su/wp-content/uploads/GOST-R-ISO-MEK-25010-2015.pdf)
- [Критерии полноты тестирования, В. В. Кулямин] (http://panda.ispras.ru/~kuliamin/lectures-mbt/Lecture03.pdf)
- [Kent Beck. Test-Driven Development By Example] (https://www.amazon.com/Test-Driven-Development-Kent-Beck/dp/0321146530)
- [David Astels. Test-Driven Development: A Practical Guide] (https://www.amazon.com/Test-Driven-Development-Practical-Guide/dp/0131016490)
- [Мартин Фаулер. Рефакторинг] (https://www.labirint.ru/books/601754/)
- [Введение в метод CEGAR — уточнение абстракции по контрпримерам] (https://www.ispras.ru/proceedings/docs/2013/24/isp_24_2013_219.pdf)
- [A Few Billion Lines of Code Later: Using Static Analysis to Find Bugs in the Real World] (https://web.stanford.edu/~engler/BLOC-coverity.pdf)
- [A Survey of Symbolic Execution Techniques] (https://arxiv.org/pdf/1610.00502.pdf)

Программу составил к.ф.-м.н. Мутилин Вадим Сергеевич (mutilin@ispras.ru)