

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Московский физико-технический институт (государственный университет)»
МФТИ (ГУ)
Кафедра «Системное программирование »**

«УТВЕРЖДАЮ»

Проректор по учебной работе

_____ Ю.Н. Волков

_____ 2012 г.

РАБОЧАЯ УЧЕБНАЯ ПРОГРАММА

по дисциплине: Верификация программ
по направлению: 230100 «Информатика и вычислительная техника»
магистерская программа: 230100 «Информатика и вычислительная техника»
факультет: ФУПМ
кафедра Системное программирование
курс: 5 (магистратура)
семестры: весенний Экзамен: 10 семестр
Трудоёмкость в зач. ед.: вариативная – 2 зач. ед
в т.ч.:
лекции: вариативная часть – 32 час.
практические (семинарские) занятия: нет
мастер классы, индивид. и групповые консультации: нет
лабораторные занятия: вариативная часть – нет
самостоятельная работа: вариативная часть – (44 час) 1,0 зач. ед.

ВСЕГО АУДИТОРНЫХ ЧАСОВ 66

Программу составил старший преподаватель, к.ф.- м.н. Пакулин Николай Витальевич

Программа обсуждена на заседании кафедры «Системное программирование»

«_____» _____ 2012 г.

Заведующий кафедрой

академик, д.ф.–м.н., профессор Иванников В.П.

ОБЪЁМ УЧЕБНОЙ НАГРУЗКИ И ВИДЫ ОТЧЁТНОСТИ.

Вариативная часть, в т.ч. :	<u> 2 </u> зач. ед.
Лекции	<u> 32 </u> часов
Практические занятия	<u> — </u> часов
Лабораторные работы	<u> — </u> часов
Индивидуальные занятия с преподавателем	<u> — </u> часов
Самостоятельные занятия	<u> 16 </u> часов
ВСЕГО	2,0 зач. ед.
Итоговая аттестация	Экзамен 10 семестр,

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 (model checking), 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. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП МАГИСТРАТУРЫ

Дисциплина «**Верификация программ**» включает в себя разделы, которые могут быть отнесены к вариативным части цикла М.2 (шифр цикла).

Дисциплина «**Верификация программ**» базируется на материалах курсов бакалавриата: базовая и вариативная часть кода УЦ ООП Б.2 (математический естественнонаучный блок) по дисциплинам «Высшая математика» (математический анализ, высшая алгебра, диффе-

ренциальные уравнения и методы математической физики), «Дискретная математика», «Математическое моделирование», «Вычислительная математика», «Программирование».

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).

3. КОНКРЕТНЫЕ ЗНАНИЯ, УМЕНИЯ И НАВЫКИ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате освоения дисциплины «Верификация программ» обучающийся должен:

1. Знать:

- место и роль средств верификации в жизненном цикле ПО;
- связь курса верификации со смежными дисциплинами дискретной математики, проектирования программных комплексов, телекоммуникационных протоколов, ;
- методы тестирования и аналитического исследования ПО;
- современные средства и технологии верификации.

2. 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.

3. Владеть:

- современными средствами и технологиями верификации ПО;
- навыками использования систем тестирования для языков высокого уровня;
- навыками использования систем исследования программ на моделях;
- навыками использования систем аналитического исследования программ;

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Структура преподавания дисциплины

Перечень разделов дисциплины и распределение времени по темам

№ темы и название	Количество часов
1. Стандарты жизненного цикла ПО	6
2. Тестирование	8
3. Модели программных систем	8
4. Тестирование с использованием моделей	4
5. Статический анализ программ	4
6. Динамический анализ программ	2
ВСЕГО(зач. ед.(часов))	32 час. (3 зач.ед.)

ВИД ЗАНЯТИЙ

ЛЕКЦИИ

№ п.п.	Темы	Трудоёмкость в зач. ед. (количество часов)
1	Стандарты жизненного цикла ПО	6
2	Тестирование	8
3	Модели программных систем	8
4	Тестирование с использованием моделей	4
5	Статический анализ программ	4
6	Динамический анализ программ	2
	ВСЕГО (зач. ед.(часов))	32

ВИДЫ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

№ п.п.	Темы	Трудоёмкость в зач. ед. (количество часов)
1.	изучение теоретического курса - выполняется самостоятельно каждым студентом по итогам каждой из лекций, результаты контролируются преподавателем на лекционных занятиях, используются конспект (электронный) лекций, учебники, рекомендуемые данной программой;	16
3.	Подготовка к экзамену	8
	ВСЕГО (зач. ед.(часов))	24 часа (1 зач.ед.)

Содержание дисциплины

Развёрнутые темы и вопросы по разделам

№ п/п	Название модулей	Разделы и темы лекционных занятий	Содержание	Объем	
				Аудиторная работа (зачетные единицы/часы)	Самостоятельная работа (зачетные единицы/часы)
1		Стандарты жизненного цикла ПО	<p>Базовые понятия о качестве программного обеспечения. Стандарты процессов жизненного цикла программного обеспечения. Место верификации в жизненном цикле.</p> <p>Стандарты и модели жизненного цикла: ISO 9000, ISO/IEC 12207, CMM, DO 178, Orange Book, Common Criteria.</p> <p>Представление о методах верификации ПО. Связи между инспекцией, тестированием, моделированием, статическим анализом,</p> <p>Ревью кода. Организация процесса ревью, сбор результатов, оценка результатов.</p>	6	3
2		Тестирование	<p>Задачи тестирования. Классификация тестирования по размеру целевых систем: модульное, компонентное, системное, интеграционное. Место тестирования в процессах жизненного цикла.</p> <p>Стандарты на процессы тестирования. Планирование тестирования, разработка тестов, оценка результатов. Тестовые покрытия. Покрытия по коду, ветвлениям, пространствам входных параметров.</p> <p>Методология тестирования xUnit. Введение в Junit. Разработка на основе тестов. Тестирование асинхронных систем</p>	8	3

			и обратных интерфейсов. Заглушки. Компонентное тестирование. Задачи интеграционного и системного тестирования.		
3		Модели программных систем	Введение в моделирование программ. История вопроса. Исполнимые модели. Конечные автоматы, расширенные конечные автоматы. Диаграммы состояний UML. Недетерминизм. Последовательная и параллельная композиции. Проблема взрыва числа состояний. Введение в сети Петри. Логические модели. Тройки Хора. Аксиоматические модели. Темпоральные логики. Формулы состояний и формулы последовательностей. Логики LTL, CTL, CTL*. Интерпретация формул на моделях. Алгебраические модели. Алгебры термов, эквивалентность термов. Переписывание.	8	2
4		Тестирование с использованием моделей	Виды моделей, пригодные для тестирования. Применение моделей в тестирование. Задача извлечения тестов. Задача построения оракула. Критерии покрытия, основанные на моделях. Технология UniTESK. Контактные спецификации, пред- и пост-условия. Генерация тестовых последовательностей из частично заданных автоматов тестов.	4	82
5		Статический анализ программ	Представление о статическом анализе. Статическая и динамическая семантика языка программирования. Базовый статический анализ на этапе компиляции.	4	2

			<p>Методы статического анализа. Абстрактная интерпретация. Построение и анализ графа потока управления.</p> <p>Проверка на моделях. Формализация требований средствами темпоральной логики. Верификация формул на автоматной модели программы или алгоритма. Построение контр-примеров.</p> <p>Доказательство корректности. Контрактные спецификации как теоремы. Доказательство теорем на основе кода программы. Доказательство интегральных свойств ПО на основе контрактов отдельных компонентов.</p>		
6		Динамический анализ программ	<p>Методы контроля потока управления в бинарных исполнимых файлах. Обнаружение утечек памяти. Выявление ошибок синхронизации.</p> <p>Методы, основанные на разрешении ограничений. DART, Avalanche.</p>	2	1

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В учебном процессе используются следующие образовательные технологии:

№ п/п	Вид занятия	Форма проведения занятий	Цель
1	лекция	Изложение теоретического материала	Получение теоретических знаний по дисциплине
2	Лекция	Изложение теоретического материала с помощью презентаций	Повышение степени понимания материала
3	Лекция	Разбор конкретных задач верификации	Осознание связей между теорией и практикой, а также взаимозависимостей разных дисциплин
4	Самостоятельная работа студента	Изучение литературы по курсу	Повышение степени понимания материала

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Контрольно-измерительные материалы

Перечень контрольных вопросов для сдачи экзамена в 10-ом семестре;

- 1) Качество программного обеспечения. Аспекты качества.
- 2) Верификация ПО. Цели и задачи. Основные подходы.
- 3) Тестирование. Задачи тестирования.
- 4) Классификация видов тестирования по размеру целевой системы.
- 5) Стандарты жизненного цикла ПО. Основные активности верификации.
- 6) Оценка покрытия верификации стандартами.
- 7) Роль верификации в жизненном цикле ПО.
- 8) Стандартизация тестирования.
- 9) Планирование процесса тестирования, основные активности.
- 10) Модульное тестирование. Основные элементы подхода xUnit.
- 11) Модульное тестирование. Пример использования JUnit или аналогичных фреймворков для других языков.
- 12) Тестовые покрытия по коду. Назначение, способ подсчета, ограничения.
- 13) Тестовые покрытия по ветвлениям. Назначение, способ подсчета, ограничения.
- 14) Тестовые покрытия по пространствам аргументов. Назначение, способ подсчета, ограничения.
- 15) Виды моделей ПО. Конечные и расширенные конечные автоматы.
- 16) Виды моделей ПО. Алгебраические спецификации.
- 17) Виды моделей ПО. Взаимодействующие процессы.
- 18) Виды моделей ПО. Логические модели: темпоральные логики, контрактные спецификации.
- 19) Тестирование с использованием моделей (МВТ). Виды моделей для МВТ.
- 20) Извлечение тестов из автоматных моделей.
- 21) Технология тестирования UniTESK. Контрактные спецификации, частичное задание автомата теста.
- 22) Тестирование с использованием моделей: тестовые покрытия по моделям.
- 23) Обзор кода. Назначение, сценарии проведения.
- 24) Статический анализ программ. Абстрактная интерпретация.
- 25) Статический анализ программ. Проверка на моделях (Model-checking).
- 26) Статический анализ программ. Доказательство корректности.
- 27) Динамический анализ программ. Основные подходы.
- 28) Динамический анализ программ: DART, Avalanche.

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Необходимое оборудование для лекций и практических занятий: компьютер и мультимедийное оборудование (проектор, звуковая система)

Необходимое программное обеспечение: система визуализации общего назначения (OpenMV, AVS), VRML браузер (Cortona3D, FreeWRL), система 4D моделирования проектов (Synchro, NavisWorks), Microsoft Office.

8. НАИМЕНОВАНИЕ ВОЗМОЖНЫХ ТЕМ КУРСОВЫХ РАБОТ учебным планом не предусмотрено

9. ТЕМАТИКА И ФОРМЫ ИНДИВИДУАЛЬНОЙ РАБОТЫ учебным планом не предусмотрено

10. ТЕМАТИКА ИТОГОВЫХ РАБОТ учебным планом не предусмотрено

11. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература.

1. С.В. Сеницын, Н.Ю. Налютин. Верификация программного обеспечения. М.: ИНТУИТ-Бином, 2008. 368 с.
2. В. В. Кулямин. Технологии программирования. Компонентный подход. М.: ИНТУИТ-Бином, 2007. 463 с.
3. Э. М. Кларк, О. Грамберг, Д. Пелед. Верификация моделей программ. Model Checking. М.: МЦНМО, 2002 г.

Дополнительная литература.

4. Котляров В.П. Основы тестирования программного обеспечения. М.: ИНТУИТ.РУ, 2006. 360 с.
5. Мендельсон Э. Введение в математическую логику. Москва: Наука, 1984.
6. Миронов А. М., Жуков Д.Ю. Математическая модель и методы верификации программных систем, Информационные технологии и вычислительные системы, 2005, 220 стр.
7. Ю. Г. Карпов. Model Checking. Верификация параллельных и распределенных программных систем. БХВ-Петербург, 2010.
8. В.В. Липаев. Системное проектирование сложных программных средств для информационных систем. Издание второе. // Серия "Управление качеством". М: СИНТЕГ, 2002.-268 с
9. В.В. Липаев. Качество программных средств. М: Янус-К, 2002. 400 с.
10. В.В. Липаев. Выбор и оценивание характеристик качества программных средств. Методы и стандарты. М.:СИНТЕГ, 2001.-228 с.

Программу составил

Пакулин Н.В. старший преподаватель, к.ф.–м.н.

« _____ » _____ 2012 г.