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
____ _____2013 г.

.

РАБОЧАЯ УЧЕБНАЯ ПРОГРАММА

по дисциплине: Верификация программ. Часть 2
по направлению: 230100 «Информатика и вычислительная техника»
магистерская программа: 230100 «Информатика и вычислительная техника»
факультет: ФУПМ
кафедра Системное программирование
курс: 6 (магистратура)
семестры: осенний Экзамен: 11 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 с.н.с. ИСП РАН, к.ф.- м.н. Камкин Александр Сергеевич
Программа обсуждена на заседании кафедры «Системное программирование»
«____» _____2013 г.

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

ОБЪЁМ УЧЕБНОЙ НАГРУЗКИ И ВИДЫ ОТЧЁТНОСТИ.

Вариативная часть, в т.ч. :	__2_ зач. ед.
Лекции	__32___ часов
Практические занятия	__–___ часов
Лабораторные работы	__–___ часов
Индивидуальные занятия с преподавателем	__–___ часов
Самостоятельные занятия	__16___ часов
ВСЕГО	2,0 зач. ед.
Итоговая аттестация	**Экзамен: 11 семестр**

1. **ЦЕЛИ И ЗАДАЧИ**

Курс **«Верификация программ. Часть 2»** является продолжением курса **«Верификация программ»** и посвящен формальной верификации программ и моделей компьютерных систем. Цели курса — (1) познакомить студентов с базовыми принципами и методами формальной верификации; (2) сформировать у студентов навыки необходимые для практического использования рассмотренных методов. Основу курса составляют: (1) методы формальной спецификации программ (пред- и постусловия, темпоральные утверждения); (2) методы формализации поведения программ (формализация семантики языков программирования, использование формальных моделей); (3) методы формальной верификации (дедуктивная верификация программ, проверка моделей).

Задачами данного курса являются:

- объяснение роли формальной верификации для построения корректных и надежных программ, формирование базовых знаний в этой области;
- обучение студентов методам формальной спецификации программ (пред- и постусловия, темпоральные утверждения);
- обучение студентов методам формализации поведения программ (формализация семантики языков программирования, использование формальных моделей);
- обучение студентов методам формальной верификации программ (дедуктивная верификация программ, проверка моделей);
- формирование теоретического подхода к верификации программ для проведения исследований в рамках выпускных работ на степень магистра.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП МАГИСТРАТУРЫ

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

Дисциплина **«Верификация программ. Часть 2»** 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, дифференциальные уравнения и методы математической физики), «Дискретная математика», «Математическое моделирование», «Вычислительная математика», «Программирование»**.**

КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины «Верификация программ. Часть 2» 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. Часть 2» обучающийся должен:

1. **Знать:**

- место и роль формальной верификации в процессе построения корректных программ;
- методы формальной спецификации и верификации программ;
- современные средства формальной верификации программ;
- связь методов формальной верификации с методами смежных дисциплин: математической логики, дискретной математики, программной инженерии.
-

2. **Уметь:**

- описывать условия корректности программ в форме пред- и постусловий;
- аналитически доказывать корректность программ;
- строить формальные модели компьютерных систем;
- описывать свойства реагирующих систем в виде формул темпоральной логики;
- применять инструментальные средства формальной верификации.

3. **Владеть:**

- навыками аналитической верификации программ;
- навыками использования средств дедуктивной верификации программ;
- навыками использования средств проверки моделей.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Структура преподавания дисциплины

Перечень разделов дисциплины и распределение времени по темам

№ темы и название	Количество часов
1. Принципы формальной верификации	4
2. Дедуктивная верификация программ	14
3. Проверка моделей (model checking)	12
4. Связь между разными методами верификации	2
ВСЕГО(зач. ед.(часов))	32 час. (3 зач.ед.)

ВИД ЗАНЯТИЙ

ЛЕКЦИИ

№ п.п.	Темы	Трудоёмкость в зач. ед. (количество часов)
1	Принципы формальной верификации	4
2	Дедуктивная верификация программ	14
3	Проверка моделей (model checking)	12
4	Связь между разными методами верификации	2
ВСЕГО (зач. ед.(часов))		32

ВИДЫ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

№ п.п.	Темы	Трудоёмкость в зач. ед. (количество часов)
1.	Изучение теоретического материала — выполняется самостоятельно каждым студентом по итогам каждой лекции, результаты контролируются преподавателем на лекционных занятиях, используются (электронный) конспект лекций, учебники, рекомендуемые данной программой;	16
2.	Решение практических задач — выполняется самостоятельно каждым студентом по итогам каждой лекции, результаты контролируются преподавателем на лекционных занятиях, используются (электронный) конспект лекций, учебники, рекомендуемые данной программой;	16
3.	Подготовка к экзамену	16
ВСЕГО (зач. ед.(часов))		48 часов (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		Принципы формальной верификации	Общая схема формальной верификации. Формальная спецификация требований. Формальная модель повеления. Соответствие поведения требованиям. Примеры методов формальной верификации. Дедуктивная верификация. Проверка моделей. Проверка эквивалентности. Формализация условий корректности. Пред- и постусловия (программный контракт). Частичная корректность. Полная корректность. Формализация семантики языков программирования. Операционная семантика. Аксиоматическая семантика. Метод доказательного программирования Дейкстры.	4	6
2		Дедуктивная верификация программ	Основные понятия дедуктивного анализа программ. Аксиомы и правила вывода (тройки Хоара). Понятие аннотированной программы. Верификация как поиск доказательства. Проблема индукции при выводе свойств циклов. Невыводимость свойств цикла из его структуры. Понятие инварианта цикла. Примеры и задания. Инструменты дедуктивной верификации программ. Язык ACSL (ANSI C Specification Language). Платформы Frama-	14	21

			C для статического анализа C-программ. Плагин Jessie для дедуктивного анализа C-программ (платформа Why). Примеры и задания. Метод индуктивных утверждений Флойда. Синтаксис и семантика блок-схем. Доказательство частичной корректности блок-схем. Точки сечения. Индуктивные утверждения. Условия верификации. Примеры и задания. Метод фундированных множеств Флойда. Доказательство полной корректности блок-схем. Оценочные функции. Условия завершимости. Примеры и задания. Верификация последовательных программ на языках программирования. Примеры и задания. Автоматизация дедуктивного анализа программ. Синтез инвариантов циклов. Генерация условий верификации. Дедуктивная верификация параллельных программ. Семантика чередований. Справедливость планировщика.		
3		Проверка моделей (model checking)	Синтаксис и семантика темпоральной логики линейного времени (LTL). Основные тождества. Выражение свойств реактивных систем в логике LTL. Свойства безопасности (safety), живости (liveness), справедливости (fairness). Примеры и задания. Инструменты проверки моделей. Язык Promela (Process/Protocol Meta Language). Инструмент проверки моделей SPIN. Примеры и задания.	12	18

			Введение в метод проверки моделей для логики LTL. Моделирование реактивных систем структурами Крипке. Множество допустимых траекторий. Контрольный автомат. Проверка выполнимости формулы. Примеры и задания. Теоретико-автоматный подход к проверке моделей для логики LTL. Автоматы Бюхи. Построение автомата Бюхи для структуры Крипке. Построение автомата Бюхи для формулы LTL. Построение синхронной композиции автоматов Бюхи. Проверка пустоты языка, допускаемого автоматом Бюхи. Примеры и задачи.		
4		Связь между разными методами верификации	Тестирование программ (методы черного и белого ящика). Тестирование на основе моделей. Дедуктивная верификация. Проверка моделей.	2	3

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 11-ом семестре;

1. Формальная верификация программ. Пред- и постусловия. Понятие частичной и полной корректности программы.
2. Операционная семантика языка программирования. Описание операционной семантики языка while.
3. Аксиоматическая семантика языка программирования. Описание аксиоматической семантики языка while.
4. Метод Дейкстры построения корректных программ. Слабейшее предусловие и его свойства.
5. Аксиоматическая система Хоара. Проблема индукции при выводе свойств циклов. Понятие инварианта цикла.
6. Метод индуктивных утверждений Флойда. Доказательство частичной корректности блок-схем.
7. Метод фундированных множеств Флойда. Доказательства полной корректности блок-схем.
8. Автоматизация дедуктивной верификации программ. Синтез инвариантов циклов. Генерация условий верификации.
9. Формальная спецификация С-программ на языке ACSL. Обзор основных возможностей языка ACSL и платформы Frama-C.
10. Параллельные программы. Семантика асинхронных чередований. Справедливость планировщика. Дедуктивная верификация алгоритма Петерсона.
11. Логика LTL. Основные тождества. Выражение свойств реагирующих систем в логике LTL. Свойства безопасности, живучести, справедливости.
12. Структуры Крипке. Представление программ с конечным числом состояний структурами Крипке. Представление параллельных программ структурами Крипке.
13. Траектории реагирующих систем. Автоматы Бюхи ω-регулярные языки. Построение синхронной композиции автоматов Бюхи.
14. Контрольный автомат реагирующей системы. Автомат Бюхи как контрольный автомат. Проверка пустоты языка, допускаемого автоматом Бюхи.
15. Автомат Бюхи для формулы LTL. Алгоритм построения автомата Бюхи для формулы LTL.
16. Теоретико-автоматный подход к проверке модели для логики LTL. Общая схема проверки выполнимости формулы LTL на модели системы.
17. Описание моделей компьютерных систем на языке Promela. Обзор основных возможностей языка Promela и инструмента Spin.

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

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

Необходимое программное обеспечение: программный пакет Frama-C, программный пакет Spin.

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. Д. Грис. Наука программирования. М.: Мир, 1984.
2. Р. Андерсон. Доказательство правильности программ. М.: Мир, 1982.
1. Э. М. Кларк, О. Грамберг, Д. Пелед. Верификация моделей программ. Model Checking. М.: МЦНМО, 2002 г.
2. Ю. Г. Карпов. Model Checking. Верификация параллельных и распределенных программных систем. БХВ-Петербург, 2010.

Дополнительная литература

3. K.R. Apt, F.S. de Boer, E.-R. Olderog. Verification of Sequential and Concurrent Programs, Springer, 2009.
4. C. Baier, J.-P. Katoen. Principles of Model Checking. The MIT Press, 2008.
5. M. Ben-Ari. Mathematical Logic for Computer Science. Springer, 2012.

Программу составил

Камкин А.С., к.ф.–м.н.

«_____»_____2013 г.