

МИНОБРНАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ

Директор института прикладной  
математики и компьютерных наук

А.В. Замятин

2021 г.



## Верификация программ

### рабочая программа дисциплины

Закреплена за кафедрой Учебный план	<i>Компьютерная безопасность 01.03.02 Прикладная математика и информатика, профиль «Прикладная математика и информатика»</i>
Форма обучения	<i>очная</i>
Общая трудоёмкость	<i>2 з.е.</i>
Часов по учебному плану	<i>72</i>
в том числе:	
аудиторная контактная работа	<i>50.65</i>
самостоятельная работа	<i>21.35</i>
Вид(ы) контроля в семестрах <i>экзамен/зачет/зачет с оценкой</i>	<i>Семестр 7 – зачет</i>

Программу составил:  
к.т.н., доцент,  
заведующий кафедрой компьютерной безопасности



С.А. Останин

Рецензент:  
к.т.н., доцент,  
доцент кафедры компьютерной безопасности



В.В. Андреева

Рабочая программа дисциплины «Верификация программ» разработана в соответствии с самостоятельно устанавливаемым образовательным стандартом высшего образования – бакалавриат – Федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский Томский государственный университет» по направлению подготовки 01.03.02 – Прикладная математика и информатика (Утвержден Ученым советом НИ ТГУ, протокол от 27.10.2021 г. № 08).

Рабочая программа одобрена на заседании кафедры компьютерной безопасности

Протокол от 02 июня 2021 г. № 06

Заведующий кафедрой программирования,  
к.т.н., доцент



С.А. Останин

Рабочая программа одобрена на заседании учебно-методической комиссии института прикладной математики и компьютерных наук (УМК ИПМКН)

Протокол от 17.06.2021 г. № 05

Председатель УМК ИПМКН,  
д.т.н., профессор



С.П. Сущенко

### Цель освоения дисциплины

**Цель** – обучить студентов тестированию программного обеспечения. Рассматриваются классические методы построения тестов: структурные методы (тестирование «белого ящика») и функциональные (тестирование «черного ящика»). Также рассматривается общая методика тестирования процедурных программ и особенности тестирования объектно-ориентированных программ. Делается обзор различных критериев качества тестирования, предлагаются различные варианты построения отказоустойчивого программного обеспечения. Рассматривается метод Model Checking (верификация моделей), ориентированный на верификацию параллельных программ с конечным числом состояний. Изучаются темпоральные логики на языке которых задаются требования к системе – спецификации; модель Крипке, моделирующая параллельную систему и методы верификации, проверяющие выполнимость спецификации на модели.

### 1. Место дисциплины в структуре ОПОП

Дисциплина относится к дисциплинам по выбору студента вариативной части Профессионального цикла Блока 1 «Модуль по выбору 2».

Пререквизиты дисциплины: «Информатика», «Дискретная математика».

Постреквизиты дисциплины: учебная и производственная практика, «Научно-исследовательская работа».

### 2. Компетенции и результаты обучения, формируемые в результате освоения дисциплины

Таблица 1.

Компетенция	Индикатор общепрофессиональной компетенции	Код и наименование результатов обучения (планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций)
<b>ОПК-2</b> Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач.	<b>ИОПК-2.1</b> Обладает навыками объектно-ориентированного программирования для решения прикладных задач в профессиональной деятельности. <b>ИОПК-2.2</b> Проявляет навыки использования основных языков программирования, основных методов разработки программ, стандартов оформления программной документации. <b>ИОПК-2.3</b> Демонстрирует умение отбора среди существующих математических методов, наиболее подходящих для решения конкретной прикладной задачи. <b>ИОПК-2.4</b> Демонстрирует умение адаптировать существующие математические методы для решения конкретной прикладной задачи.	<b>ОР-1.1.</b> - владеть навыками объектно-ориентированного программирования для разработки программных систем; - уметь разрабатывать тестовые варианты для программных систем с применением концепции объектно-ориентированного программирования. <b>ОР-1.2.</b> - знать стандарты оформления программной документации; - уметь применять навыки использования современных языков программирования и основные методы написания программ для разработки тестовых вариантов. <b>ОР-1.3.</b> - знать основные модели разработки ПО; - уметь выбирать наиболее подходящий математический метод для решения конкретной прикладной задачи.

<b>ПК-2</b> Способен формализовать и алгоритмизировать поставленную задачу, написать программный код, а также проверить работоспособность программного обеспечения и исправить дефекты.	<b>ИПК-2.1</b> Осуществляет построение формальной модели и алгоритма для поставленной задачи, написание программного кода с использованием языков программирования, проверку работоспособности программного обеспечения и исправление дефектов.	<b>ОР-2.1.</b> - уметь разработать алгоритм тестирования ПО, реализовать этот алгоритм на языке программирования; - владеть навыками проверки работоспособности распределенных приложений, работающих в сетях с архитектурой «клиент-сервер», а также веб-сервисов; - уметь исправлять найденные в приложениях дефекты.
	<b>ИПК-2.2</b> Осуществляет оформление программного кода в соответствии с установленными требованиями, разработку процедур проверки работоспособности и измерения характеристик программного обеспечения, разработку тестовых наборов данных.	<b>ОР-2.2.</b> Обучающийся сможет: - оформить программный код в соответствии с установленными требованиями; - разработать тестовый набор данных для проверки работы созданного распределенного приложения или веб-сервиса.
	<b>ИПК-2.3</b> Осуществляет работу с системой контроля версий, рефакторинг и оптимизацию программного кода.	<b>ОР-2.3.</b> Обучающийся сможет: - работать с системой контроля версий; - выполнять рефакторинг и оптимизацию программного кода.

### 3. Структура и содержание дисциплины

#### 3.1. Структура и трудоемкость видов учебной работы по дисциплине

Общая трудоемкость дисциплины/модуля составляет 2 зачетные единицы, 72 часов.

Таблица 2.

Вид учебной работы	Трудоемкость в академических часах	
	7 семестр	всего
<b>Общая трудоемкость</b>	<b>72</b>	<b>72</b>
<b>Контактная работа:</b>	<b>50,65</b>	<b>50,65</b>
Лекции (Л):	48	48
Практики (ПЗ)		
Лабораторные работы (ЛР)		
Семинары (СЗ)		
Групповые консультации	2,4	2,4
Индивидуальные консультации		
Промежуточная аттестация	0,25	0,25
<b>Самостоятельная работа обучающегося:</b>	<b>21,35</b>	<b>21,35</b>
- выполнение контрольных заданий	5	5
- подготовка доклада, сообщения	4	4
- изучение учебного материала, публикаций	4	4
- подготовка к тестам	4	4
- подготовка к рубежному контролю по теме/разделу	4,35	4,35
<b>Вид промежуточной аттестации (зачет, зачет с оценкой, экзамен)</b>	<b>Зачет</b>	

### 3.2. Содержание и трудоемкость разделов дисциплины

Таблица 3.

Код занятия	Наименование разделов и тем и их содержание	Вид учебной работы, занятий, контроля	Семестр	Часы в электронной форме	Всего (час.)	Литература	Код (ы) результата(ов) обучения
	<b>Раздел 1.</b> Тестирование программного обеспечения (ПО).					№ 1, № 2, № 3, №6	ОП-1.1, ОП-2.1, ОП-2.2, ОП-2.3
1.1.	Тестирование программного обеспечения (ПО). Базовые понятия и определения.	Лекции	7		2		
1.2.	Методы построения тестов. Структурное тестирование ПО.	Лекции	7		4		
1.3.	Методы построения тестов. Функциональное тестирование.	Лекции	7		4		
1.4.	Примеры построения тестов ПО.	Лекции	7		2		
1.5.	Организация процесса тестирования ПО.	Лекции	7		2		
1.6.	Объектно-ориентированное тестирование (ООТ).	Лекции	7		4		
1.7.	Критерии качества тестирования. Понятие тестопригодности ПО.	Лекции	7		1		
1.8.	Изучение учебного материала.	СРС	7		10		
1.9.	Выступление с докладами		7		2		
1.10.	<b>Контрольная работа</b>		7		2		
	<b>Раздел 2. Формальная верификация ПО</b>		7			№ 4, № 5	ОП-1.1, ОП-1.2, ОП-1.3
2.1.	Введение. Основные понятия математической логики. Исчисление высказываний. Исчисление предикатов.	Лекции	7		4		
2.2.	Дедуктивный анализ. Метод индуктивных утверждений Флойда. Метод фундированных множеств Флойда.	Лекции	7		4		
2.3.	Метод верификации моделей программ (Model Checking).	Лекции	7		4		
2.4.	Введение. Моделирование систем. Модели Крипке.	Лекции	7		3		
2.5.	Темпоральная логика. Логика CTL*, CTL, LTL.	Лекции	7		3		
2.6.	Метод верификации для формул логики CTL.	Лекции	7		3		
2.7.	Изучение учебного материала.	СРС	7		11,35		
2.8.	<b>Контрольная работа.</b>		7		2		
	<b>Промежуточная аттестация в форме зачета</b>		7				

#### 4. Образовательные технологии, учебно-методическое и информационное обеспечение для освоения дисциплины

Основой обучения является курс лекций, читаемый преподавателем.

Самостоятельная работа студентов изучение учебного материала, подготовку к лабораторным работам и экзамену.

Промежуточная аттестация осуществляется на основе собеседования при условии успешного выполнения ранее лабораторных работ.

##### 4.1. Рекомендуемая литература и учебно-методическое обеспечение

№ п/п	Авторы / составители	Заглавие	Издательство	Год издания
Основная литература				
1.	Орлов С. А.	Программная инженерия: технологии разработки программного обеспечения	Питер	2016
2.	Гниденко И.Г., Павлов Ф.Ф., Федоров Д.Ю.	Технология разработки программного обеспечения : учебное пособие для СПО : Учебное пособие	Юрайт	2018
3.	Плаксин М. А.	Тестирование и отладка программ : для профессионалов будущих и настоящих	Бином	2013
4.		Verification, Model Checking, and Abstract Interpretation 16th International Conference, VMCAI 2015, Mumbai, India, January 12-14, 2015. Proceedings	Springer eBooks	2015
Дополнительная литература				
5.	Эдмунд М. Кларк, мл. , Орна Грамберг, Дорон Пелед	Верификация моделей программ: Model Checking	МЦНМО	2002
6.	Бейзер Б	Тестирование черного ящика : технологии функционального тестирования программного обеспечения и систем	Питер	2004

##### 4.2. Базы данных и информационно-справочные системы, в том числе зарубежные

1. Котляров В. Основы тестирования программного обеспечения [Электронный ресурс] // «ИНТУИТ» Национальный Открытый Университет / НОУ «ИНТУИТ». – Электрон. дан. – М. – URL:<http://www.intuit.ru/studies/courses/48/48/info>
2. Портал специалистов по тестированию и обеспечению качества ПО [Электронный ресурс] : Тестирование и Качество ПО : Software-Testing.RU. URL: <http://software-testing.ru/>
3. Академия Microsoft: Верификация программного обеспечения [Электронный ресурс] : учебный курс // «ИНТУИТ» Национальный Открытый Университет / НОУ «ИНТУИТ». – Электрон. дан. – М. – URL: <http://www.intuit.ru/studies/courses/1040/209/info>

##### 4.3. Перечень лицензионного и программного обеспечения

Не требуется.

#### **4.4. Оборудование и технические средства обучения**

Аудитории для проведения интерактивных лекций: видеопроектор, экран настенный.

#### **5. Методические указания обучающимся по освоению дисциплины**

Основой обучения является курс лекций, читаемый преподавателем. Для самостоятельной работы и дополнительного расширения круга знаний желательно использовать литературу, приведенную в разделе 4.1, а также информационные системы, приведенные в разделе 4.2.

#### **6. Преподавательский состав, реализующий дисциплину**

Останин Сергей Александрович, к.т.н, доцент, заведующий кафедрой компьютерной безопасности НИ ТГУ

#### **7. Язык преподавания – русский язык.**