ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«24» июня 2025 г.
Протокол №2

Рабочая программа дисциплины (модуля) «Верификация программ»

Направление подготовки: *38.03.05 Бизнес-информатика*

Направленность (профиль) подготовки: *Бизнес-аналитика*

Квалификация (степень) выпускника: *бакалавр*

Форма обучения: *очная*

Срок освоения программы: *4 года*

Год набора: *2025*

Москва
2025

Электронный документ

Содержание

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Верификация программ» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 38.03.05 Бизнес-информатика, профиль Бизнес-аналитика, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 838 от 29.07.2020 года.

Изучение дисциплины (модуля) «Верификация программ» важно для обеспечения корректности и надежности программного обеспечения, что снижает риски ошибок и сбоев в системах. Это способствует повышению качества разработки и безопасности программ, особенно в критически важных приложениях.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 38.03.05 Бизнес-информатика, профиль Бизнес-аналитика и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений.

Дисциплина (модуль) является выборной и доступна для изучения на 2, 3 или 4 курсе в 4, 5, 6, 7 или 8 семестрах на выбор.

Цель изучения дисциплины (модуля): освоение методов и инструментов формальной проверки корректности программ для обеспечения их надежной и безопасной работы.

Задачи изучения дисциплины (модуля):

— изучить теоретические основы формальной верификации, включая математические модели, логику предикатов и техники доказательства свойств программ;

— освоить практические навыки применения инструментов верификации, таких как статический анализ, модельная проверка и тестирование на основе моделей;

— научиться понимать интеграции методов верификации в жизненный цикл разработки программного обеспечения для предотвращения ошибок и повышения качества кода.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

— основные принципы и подходы к формальной верификации программного обеспечения;

— методы верификации программ;

— основные понятия теории автоматов и применения их в верификации ПО;

уметь:

— аналитически доказывать корректность программ;

— находить инварианты циклов и функции оценки для доказательства корректности;

— специфицировать программы, формулируя пред- и постусловия;

владеть:

— навыком работы с инструментами верификации моделей.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области аналитики, основные принципы критической оценки источников информации и их релевантности.
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
ОПК-2.	Способен проводить исследование и анализ рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом	ОПК-2.1.	Знает основные тенденции и характеристики рынка информационных систем и информационно-коммуникационных технологий
		ОПК-2.2.	Умеет проводить исследование и анализ рыночной информации для оценки потребностей бизнеса и выбора оптимальных решений
		ОПК-2.3.	Имеет практический опыт в разработке и внедрении стратегий управления бизнесом на основе анализа рынка информационных технологий
ПК-2.	Способен использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования для решения задач профессиональной деятельности	ПК-2.1.	Знает основные математические методы и инструментальные средства, применяемые для обработки и анализа информации
		ПК-2.2.	Умеет эффективно использовать математический аппарат для систематизации данных и решения профессиональных задач
		ПК-2.3.	Имеет практический опыт работы с инструментами анализа информации в рамках исследовательских проектов
ПК-3.	Способен готовить научно-технические отчеты,	ПК-3.1.	Знает требования и стандарты оформления научно-технических

	презентации, научные публикации по результатам выполненных исследований		отчетов, презентаций и публикаций
		ПК-3.2.	Умеет структурировать и представлять результаты исследований в ясной и доступной форме
		ПК-3.3.	Имеет практический опыт подготовки и публикации научных материалов, отражающих результаты выполненных исследований
ПК-8	Способен под руководством специалиста более высокой категории осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами	ПК-8.1.	Знает принципы и стандарты управления проектами
		ПК-8.2.	Умеет разрабатывать планы и организовывать проектную деятельность в соответствии с установленными стандартами
		ПК-8.3.	Имеет практический опыт участия в проектной работе, включая планирование и координацию задач

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Контактная работа		Контроль	Самостоятельная работа	
		Лекции	Семинары (практические занятия)			
1	Основы формальной верификации	4	4		18	Кейс Коллоквиум
2	Дедуктивная верификация последовательных программ	4	4		18	Кейс Коллоквиум
3	Модели и логики для систем с состояниями	4	4		18	Кейс Коллоквиум
4	Верификация параллельных систем	4	4		18	Кейс Коллоквиум
5	Статический анализ и абстрактная интерпретация	4	4		20	Кейс Коллоквиум
6	SAT/SMT-подходы и ограниченная верификация	4	4		20	Кейс Коллоквиум
7	Интеграция методов и промышленный опыт	4	4		20	Кейс Коллоквиум
	Зачет с оценкой			2		Проект
	Итого:	***28***	***28***	***2***	***132***	
Объем дисциплины (модуля) (в ак. ч.)		***190***				
Объем дисциплины (модуля) (в зач. ед.)		***5***				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы формальной верификации	Введение в формальные методы. Логические модели программ
2	Дедуктивная верификация последовательных программ	Дедуктивная верификация базовых алгоритмов. Верификация структур данных и рекурсии
3	Модели и логики для систем с состояниями	Формальные модели систем состояний. Анализ логических свойств систем состояний
4	Верификация параллельных систем	Параллельные вычисления и их модели. Проверка свойств параллельных систем. Верификация под слабой моделью памяти
5	Статический анализ и абстрактная интерпретация	Основы статического анализа кода. Автоматический вывод инвариантов
6	SAT/SMT-подходы и ограниченная верификация	Ограниченная верификация на основе ограничений. Символьное исполнение и исследование путей
7	Интеграция методов и промышленный опыт	Интеграция формальных методов в разработку. Практика применения формальной проверки

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Кудрявцев, В. Б. Теория автоматов : учебник для вузов / В. Б. Кудрявцев, Э. Э. Гасанов, А. С. Подколзин. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 199 с. — (Высшее образование). — ISBN 978-5-534-15339-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/569495.

2. Клейн, Т. Дневник охотника за ошибками. Путешествие через джунгли проблем безопасности программного обеспечения : практическое руководство / Т. Клейн ; пер. с англ. А. Н. Киселёва. - 2-е изд. - Москва : ДМК Пресс, 2023. - 241 с. - ISBN 978-5-89818-599-2. - Текст : электронный. - URL: https://znanium.com/catalog/product/210847.

3. Лабун, Б. Дружеское знакомство с тестированием программ : практическое руководство / Б. Лабун. - Санкт-Петербург : БХВ-Петербург, 2022. - 288 с. - ISBN 978-5-9775-6807-4. - Текст : электронный. - URL: https://znanium.com/catalog/product/2122884.

4. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/561885.

Дополнительная литература:

1. Казарин, О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для вузов / О. В. Казарин, А. С. Забабурин. — Москва : Издательство Юрайт, 2025. — 312 с. — (Высшее образование). — ISBN 978-5-9916-9043-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/562070.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

— столами и стульями;
— компьютерной техникой;
— механическими калькуляторами;
— специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование"	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное

Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Верификация программ» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, кейсы, коллоквиумы, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удается, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар — это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Кейс – практическая работа студентов над реальными или смоделированными задачами, что позволяет студенту применять теоретические знания на практике.

Студент самостоятельно разрабатывает стратегию решения поставленной задачи, что способствует развитию навыков критического мышления и самостоятельного принятия

решений. Такой подход помогает подготовить будущих специалистов к реальным вызовам в их профессиональной деятельности.

Коллоквиум – устные ответы на вопросы, список которых известен студенту заранее.

В процессе подготовки к коллоквиуму необходимо проанализировать учебные материалы, ознакомившись с лекциями, учебниками и дополнительными источниками, акцентируя внимание на ключевых темах. Рекомендуется создать структурированные конспекты, выделяя основные идеи, термины и формулы.

Проект – исследовательская работа по курсу и презентация результатов.

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов, планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Верификация программ»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме ***зачета с оценкой***, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	**Пятибалльная оценка**	**Оценка за зачет**	**Общая характеристика результата обучения по дисциплине (модулю)**
10	Отлично	Зачтено	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует
9	Отлично	Зачтено	
8	Отлично	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
7	Хорошо	Зачтено	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	Зачтено	
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Верификация программ» оценивается следующим образом:

Активность	Вес	Описание
Кейс	40%	Практическая работа студентов над реальными или смоделированными задачами, что позволяет студенту применять теоретические знания на практике
Коллоквиумы	30%	Устные ответы на вопросы, список которых известен студенту заранее
Проект	30%	Защита итогового проекта

Формула расчёта итоговой оценки по дисциплине (модулю) «Верификация программ»: «0,4 × среднее за кейсы + 0,3 × среднее за коллоквиумы + 0,3 × проект».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные задания для кейсов

Кейс 1: Основы формальной верификации (Введение в формальные методы. Логические модели программ)

Задание 1: Постройте логическую модель простой программы (например, алгоритм сортировки пузырьком) с использованием логики предикатов первого порядка. Определите предусловия, постусловия и инварианты цикла, затем докажите корректность вручную. Напишите отчет с формальными доказательствами и объясните, как эта модель помогает в верификации.

Задание 2: Реализуйте простую программу на языке с аннотациями (например, в Dafny или Why3) и используйте инструмент для проверки корректности. Внедрите ошибку (например, нарушение инварианта) и проанализируйте, как инструмент выявляет ее. Сравните с ручным доказательством и предложите улучшения модели.

Задание 3: Изучите применение формальных методов в реальном проекте (например, анализ открытого кода из GitHub). Выберите фрагмент кода, смоделируйте его логически и проверьте на соответствие спецификациям. Напишите отчет о найденных потенциальных ошибках и рекомендациях по интеграции формальной верификации в процесс разработки.

Кейс 2: Модели и логики для систем с состояниями (Формальные модели систем состояний. Анализ логических свойств систем состояний)

Задание 1: Смоделируйте простую систему состояний (например, конечный автомат для светофора) с использованием TLA+ или Alloy. Определите состояния, переходы и свойства (например, безопасность и живучесть), затем проверьте их с помощью инструмента. Напишите спецификацию и объясните, как модель выявляет ошибки.

Задание 2: Реализуйте модель системы с состояниями (например, протокол взаимного исключения) в Spin и проведите проверку на свойства CTL (Computation Tree Logic), такие как AG (всегда глобально) или EF (существует будущее). Внедрите ошибку (например, deadlock) и продемонстрируйте, как инструмент находит ее. Сравните с ручным анализом.

Задание 3: Примените анализ логических свойств к реальной системе (например, моделированию кэширования в процессоре). Используйте TLA+ для проверки инвариантов и временных свойств. Проведите симуляцию и напишите отчет о результатах, включая сценарии, где свойства нарушаются, и предложения по улучшению модели.

Кейс 3: Статический анализ и абстрактная интерпретация (Основы статического анализа кода. Автоматический вывод инвариантов)

Задание 1: Примените статический анализ к простому коду (например, на C) с использованием Frama-C или Clang Static Analyzer. Выявите потенциальные ошибки (например, null pointer dereference) и сгенерируйте отчет. Сравните результаты с ручным анализом и объясните принципы абстрактной интерпретации.

Задание 2: Реализуйте автоматический вывод инвариантов для цикла (например, в программе на Java) с помощью инструмента вроде Infer или Astrée. Внедрите ошибку (например, переполнение) и проанализируйте, как инструмент выводит инварианты и обнаруживает проблему. Напишите скрипт для демонстрации и предложите улучшения.

Задание 3: Проанализируйте открытый проект (например, фрагмент кода из Linux kernel) с использованием статического анализатора. Выявите уязвимости и выведите инварианты с помощью абстрактной интерпретации. Напишите детальный отчет о найденных проблемах, мерах mitigation и интеграции такого анализа в CI/CD пайплайн.

Примерные задания и вопросы к коллоквиуму

Тема 1: Основы формальной верификации (Введение в формальные методы. Логические модели программ)

1. Объясните суть формальной верификации программ. Приведите пример логической модели простой программы (например, алгоритма поиска минимума в массиве) с предусловиями, постусловиями и инвариантами.
2. Что такое аксиоматическая семантика Хоара? Продемонстрируйте её применение на примере верификации цикла, доказав корректность вручную.
3. Опишите различия между формальными и неформальными методами верификации. Приведите пример, где формальная модель выявляет скрытую ошибку в программе.

Тема 2: Дедуктивная верификация последовательных программ (Дедуктивная верификация базовых алгоритмов. Верификация структур данных и рекурсии)

1. Как осуществляется дедуктивная верификация рекурсивных функций? Приведите пример доказательства корректности функции факториала с использованием логики предикатов.
2. Верифицируйте простой алгоритм сортировки (например, вставками) с помощью аннотаций пред- и постусловий. Опишите шаги доказательства и возможные ловушки.
3. Объясните верификацию структур данных, таких как стек или очередь. Смоделируйте операцию push/pop для стека и докажите её корректность.

Тема 3: Модели и логики для систем с состояниями (Формальные модели систем состояний. Анализ логических свойств систем состояний)

1. Что такое конечный автомат как модель системы состояний? Приведите пример модели светофора и проанализируйте его свойства (безопасность и живучесть).
2. Опишите логику CTL (Computation Tree Logic). Продемонстрируйте проверку свойства "всегда глобально" (AG) на примере простой системы состояний.

3. Как моделировать систему состояний в TLA+? Создайте спецификацию для протокола взаимного исключения и объясните, как проверить инварианты.

Тема 4: Верификация параллельных систем (Параллельные вычисления и их модели. Проверка свойств параллельных систем. Верификация под слабой моделью памяти)

1. Объясните модель параллельных вычислений CSP (Communicating Sequential Processes). Приведите пример верификации свойства свободы от deadlock в параллельной программе.
2. Что такое слабая модель памяти и почему она важна для верификации? Продемонстрируйте проблему на примере кода с race condition.
3. Как проверять свойства параллельных систем с помощью Spin? Опишите процесс моделирования и проверки временных свойств (например, LTL).

Тема 5: Статический анализ и абстрактная интерпретация (Основы статического анализа кода. Автоматический вывод инвариантов)

1. В чём суть статического анализа кода? Приведите пример использования инструмента Frama-C для выявления null pointer dereference в C-коде.
2. Объясните концепцию абстрактной интерпретации. Как она применяется для автоматического вывода инвариантов в циклах?
3. Продемонстрируйте статический анализ программы с циклом, используя Infer. Опишите выявленные проблемы и предложите исправления.

Тема 6: SAT/SMT-подходы и ограниченная верификация (Ограниченная верификация на основе ограничений. Символьное исполнение и исследование путей)

1. Что такое SAT/SMT-colvers и как они используются в верификации? Приведите пример решения задачи верификации с помощью Z3.
2. Объясните символьное исполнение. Как оно помогает в исследовании путей программы и выявлении ошибок?
3. Опишите ограниченную верификацию (bounded model checking). Продемонстрируйте на примере верификации цикла с ограниченной глубиной.

Тема 7: Интеграция методов и промышленный опыт (Интеграция формальных методов в разработку. Практика применения формальной проверки)

1. Как интегрировать формальные методы в процесс разработки ПО? Приведите пример из практики (например, в проектах вроде seL4 или CompCert).
2. Обсудите преимущества и вызовы применения формальной верификации в промышленности. Какие инструменты (например, Coq или Isabelle) используются для этого?
3. Проанализируйте кейс применения формальной проверки в реальном проекте (например, верификация контрактов в Solidity). Опишите шаги интеграции и результаты.

Примерное описание для проекта

Тема проекта: Верификация параллельных систем (Тема 4: Параллельные вычисления и их модели. Проверка свойств параллельных систем. Верификация под слабой моделью памяти).
Название проекта: Верификация протокола взаимного исключения в параллельной системе с использованием модели слабой памяти.

Цели проекта

- Изучить принципы моделирования и верификации параллельных систем, включая модели параллельных вычислений (например, CSP или TLA+).
- Освоить методы проверки свойств параллельных программ, таких как безопасность (safety), живучесть (liveness) и отсутствие deadlock/race conditions.
- Проанализировать влияние слабой модели памяти на корректность параллельного кода и научиться верифицировать программы под такими моделями.
- Применить теоретические знания на практике, используя инструменты формальной верификации (например, TLA+ или Spin).

Задачи проекта

1. **Моделирование системы:** Разработать формальную модель простого параллельного протокола (например, Peterson's algorithm или bakery algorithm для взаимного исключения) с учетом слабой модели памяти (например, TSO – Total Store Order).
2. **Формулировка свойств:** Определить ключевые свойства системы (инварианты, временные логики LTL/CTL) и проверить их на наличие ошибок (deadlock, starvation, race conditions).
3. **Верификация:** Использовать инструмент (TLA+ или Spin) для симуляции и проверки модели. В случае ошибок предложить исправления и повторно верифицировать.
4. **Анализ и отчет:** Сравнить результаты верификации под сильной и слабой моделями памяти, обсудить практическую применимость и ограничения методов.

Этапы выполнения проекта
Проект рассчитан на один семестр (примерно 12–16 недель). Рекомендуется выполнять поэтапно с промежуточными проверками.

1. **Подготовительный этап (Недели 1–2):** Изучение литературы и инструментов. Выбор протокола и инструмента (например, TLA+ для моделирования). Составление плана проекта.
2. **Моделирование и формулировка (Недели 3–5):** Создание формальной модели системы. Определение модулей, процессов и свойств. Промежуточная проверка: демонстрация базовой модели преподавателю.
3. **Верификация и анализ (Недели 6–10):** Запуск симуляций и проверок в инструменте. Анализ результатов, выявление ошибок и их исправление. Сравнение моделей памяти.
4. **Оформление и защита (Недели 11–12):** Написание отчета, подготовка презентации. Финальная проверка кода и моделей.

Сроки выполнения

- **Дедлайн промежуточной проверки (модель и свойства):** Конец 5-й недели.
- **Дедлайн финального отчета и презентации:** Конец 12-й недели (или по расписанию защиты проектов).
- **Общий срок:** Проект должен быть завершен к экзаменационной сессии. Задержки допускаются только с обоснованием и согласованием с преподавателем.

Критерии защиты и оценки
Проект оценивается по шкале 0–100 баллов. Защита включает презентацию (10–15 минут) и ответы на вопросы. Критерии:

- **Полнота модели (20 баллов):** Корректность формальной спецификации, учет параллелизма и слабой памяти.
- **Корректность верификации (25 баллов):** Правильное применение инструментов, точность проверки свойств, выявление и исправление ошибок.
- **Анализ и выводы (20 баллов):** Глубина сравнения моделей, обсуждение ограничений и практических аспектов.
- **Оформление отчета (15 баллов):** Структура (введение, методы, результаты, заключение), наличие кода/моделей, ссылок на литературу, объем 15–25 страниц.
- **Презентация и защита (20 баллов):** Ясность изложения, демонстрация работы, ответы на вопросы (технические и теоретические).

Минимальный порог: 50 баллов для зачета. Высокие баллы (80+) за инновационные подходы, дополнительные свойства или интеграцию с кодом (например, на C++ с моделью памяти).

Требования к оформлению и ресурсам

- **Отчет:** В формате PDF, с использованием LaTeX или Word. Включить: титульный лист, аннотацию, введение, теоретическую часть, описание модели, результаты верификации, заключение, список литературы.
- **Код и модели:** Предоставить исходные файлы (TLA+ specs, Spin models) на GitHub или в архиве.
- **Инструменты:** Рекомендуется TLA+ Toolbox (бесплатный) или Spin (с Promela). Для слабой памяти – расширение моделей в инструментах.
- **Ресурсы:** Литература: "Specifying Systems" Лампорта (TLA+), "Principles of Model Checking" Баера. Онлайн-ресурсы: официальные сайты инструментов, курсы на Coursera/EdX по формальной верификации.
- **Групповая работа:** Проект индивидуальный, но допускается обсуждение идей. Плагиат недопустим.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	Укажите основную цель формальной верификации программ.	Корректность (корректность, correctness)	УК-1
2	Укажите тип логики для моделирования программ.	Темпоральная (темпоральная, temporal)	УК-1
3	Укажите метод дедуктивной верификации алгоритмов.	Инварианты (инварианты, invariants)	УК-1
4	Укажите подход к анализу систем с	Модальная логика (модальная	УК-1

	состояниями.	логика, modal logic)	
5	Укажите инструмент для анализа рынка формальных методов.	Gartner (gartner, гарнер)	ОПК-2
6	Укажите рациональное решение для верификации параллельных систем в бизнесе.	Модельная проверка (модельная проверка, model checking)	ОПК-2
7	Укажите инструмент для статического анализа кода.	Clang Static Analyzer (clang static analyzer, кланг статический анализатор)	ОПК-2
8	Укажите платформу для управления проектами верификации.	Jira (jira, джира)	ОПК-2
9	Укажите математический аппарат для символьного исполнения.	SMT-солвер (smt-солвер, smt solver)	ПК-2
10	Укажите логику для верификации рекурсии.	Хоарова логика (хоарова логика, hoare logic)	ПК-2
11	Укажите метод абстрактной интерпретации.	Фиксированная точка (фиксированная точка, fixed point)	ПК-2
12	Укажите подход к ограниченной верификации.	SAT-решатель (sat-решатель, sat solver)	ПК-2
13	Укажите формат для научно-технических отчетов по верификации.	PDF (pdf, пдф)	ПК-3
14	Укажите инструмент для презентаций по формальным методам.	PowerPoint (powerpoint, пауэрпоинт)	ПК-3
15	Укажите стиль цитирования для публикаций по верификации.	IEEE (ieee, айтрипл-и)	ПК-3
16	Укажите платформу для публикаций по статическому анализу.	ArXiv (arxiv, арксив)	ПК-3
17	Укажите методологию для планирования проектов по верификации.	Scrum (scrum, скрум)	ПК-8
18	Укажите инструмент для трекинга задач в проектах верификации.	Trello (trello, трелло)	ПК-8
19	Укажите стандарт для управления проектами в формальной проверке.	ITIL (itil, итл)	ПК-8
20	Укажите подход к тестированию в интеграции формальных методов.	Unit testing (unit testing, юнит-тестирование)	ПК-8