



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**Б1.В.01.01 Верификация и качество программного обеспечения**

**Направление подготовки:** 01.04.02 Прикладная математика и информатика

**Направленность (профиль):** Информационное обеспечение экономической деятельности

**Квалификация (степень):** *магистр*

**Форма обучения:** *очная*

**Институт:** математики, естествознания и техники

**Кафедра:** математического моделирования и компьютерных технологий

	<b>очная форма</b>	<b>очно-заочная форма</b>	<b>заочная форма</b>
<b>Курс</b>	<b>2</b>		
<b>Семестр/триместр</b>	<b>3, 4</b>		
<b>Лекций</b>	<b>10</b>		
<b>Лабораторных занятий</b>	<b>32</b>		
<b>Практических (семинарских) занятий</b>	<b>11</b>		
<b>Консультации</b>	<b>-</b>		
<b>Форма(ы) промежуточной аттестации</b>	<b>Зачет с оценкой в 3 и 4 семестрах</b>		
<b>Контроль</b>	<b>-</b>		
<b>Иные формы работы</b>	<b>-</b>		
<b>Самостоятельная работа</b>	<b>91</b>		

**Всего часов:** **144**

**Трудоемкость:** **4** зачетные единицы.

## I. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

### Цель изучения дисциплины:

Дисциплина Б1.В.01.01 «Верификация и качество программного обеспечения» призвана ознакомить студентов с основами современной методологии верификации программ с позиций ее практического использования в процессе разработки программ. В качестве языка спецификаций обычно рассматривается язык логики предикатов первого порядка, который расширяется добавлением различных понятий.

### Задачи изучения дисциплины:

Задачами изучения дисциплины Б1.В.01.01 «Верификация и качество программного обеспечения» являются:

- введение в формальные методы проверки правильности программ;
- обзор традиционных методов верификации;
- рассмотрение методов проверки на модели;
- ознакомление с подходами к верификации параллельных программ.

**Место дисциплины в структуре ОПОП:** реализуется в рамках вариативной части (части, формируемой участниками образовательных отношений) блока Б1. Дисциплины (модули).

### Планируемые результаты обучения по дисциплине, соотнесенные с индикаторами достижения компетенций:

Код компетенции	Индикаторы компетенции	Планируемые результаты обучения по дисциплине
УК-1	<b>Знать:</b> – методы критического анализа и оценки современных научных достижений; – основные принципы критического анализа.	<b>Знает:</b> – основные понятия научного исследования; – методы критического анализа и оценки современных научных достижений с использованием методов научного исследования.
	<b>Уметь:</b> – анализировать проблемную ситуацию как систему, выявляя ее составляющие и связи между ними; – осуществлять поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации; – определять в рамках выбранного алгоритма вопросы (задачи), подлежащие дальнейшей	<b>Умеет:</b> – использовать методы научного исследования для поиска вариантов решения поставленной проблемной ситуации.

	разработке.	
	<p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>– навыками разработки стратегии достижения поставленной цели как последовательности шагов, предвидя результат каждого из них и оценивая их влияние на внешнее окружение планируемой деятельности и на взаимоотношения участников этой деятельности.</li> </ul>	<p><b>Владеет:</b></p> <ul style="list-style-type: none"> <li>– навыками разработки стратегии достижения поставленной цели;</li> <li>– методологией и методами научного исследования для решения профессиональных задач.</li> </ul>
<b>ПКС-1</b>	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>– методы и приемы организации процесса разработки баз данных ИС;</li> <li>– методологии разработки баз данных ИС и технологии программирования;</li> <li>– языки формализации функциональных спецификаций;</li> <li>– методы и приемы формализации задач;</li> <li>– методы и средства проектирования программного обеспечения, программных интерфейсов и баз данных ИС.</li> </ul>	<p><b>Знает:</b></p> <ul style="list-style-type: none"> <li>– методы и приемы организации процесса разработки программного обеспечения;</li> <li>– методологии разработки программного обеспечения и технологии программирования;</li> <li>– языки формализации функциональных спецификаций;</li> <li>– методы и приемы формализации задач;</li> <li>– методы и средства проектирования программного обеспечения, программных интерфейсов.</li> </ul>
	<p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>– использовать существующие типовые решения проектирования баз данных ИС;</li> <li>– применять методы и средства проектирования структур данных, баз данных, программных интерфейсов;</li> <li>– осуществлять коммуникации с заинтересованными сторонами;</li> <li>– выбирать средства реализации требований к базам данных ИС;</li> <li>– вырабатывать варианты реализации баз данных ИС и требований к нему;</li> <li>– проводить анализ исполнения требований.</li> </ul>	<p><b>Умеет:</b></p> <ul style="list-style-type: none"> <li>– использовать существующие типовые решения проектирования программного обеспечения;</li> <li>– применять методы и средства проектирования структур данных, баз данных, программных интерфейсов;</li> <li>– осуществлять коммуникации с заинтересованными сторонами;</li> <li>– выбирать средства реализации требований к программному обеспечению;</li> <li>– вырабатывать варианты реализации программного обеспечения и требований к нему;</li> <li>– проводить анализ исполнения требований.</li> </ul>
	<p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>– методологией организации процесса разработки баз дан-</li> </ul>	<p><b>Владеет:</b></p> <ul style="list-style-type: none"> <li>– методологией организации процесса разработки программного обеспечения;</li> </ul>

	<p>ных ИС;</p> <ul style="list-style-type: none"> <li>– методологией и технологиями проектирования программных интерфейсов, структур и баз данных ИС в соответствии с установленными требованиями;</li> <li>– действиями по разработке и согласованию технических спецификаций на программные компоненты;</li> <li>– действиями по согласованию требований к базам данных ИС с заинтересованными сторонами, распределению заданий между программистами в соответствии с техническими спецификациями, осуществлению контроля выполнения заданий, формированию отчетности в соответствии с установленными регламентами.</li> </ul>	<ul style="list-style-type: none"> <li>– методологией и технологиями проектирования программных интерфейсов, структур и баз данных ИС в соответствии с установленными требованиями;</li> <li>– действиями по разработке и согласованию технических спецификаций на программные компоненты;</li> <li>– действиями по согласованию требований к программному обеспечению с заинтересованными сторонами, распределению заданий между программистами в соответствии с техническими спецификациями, осуществлению контроля выполнения заданий, формированию отчетности в соответствии с установленными регламентами.</li> </ul>
--	--	--

## II. СОДЕРЖАНИЕ И ОБЪЕМ ДИСЦИПЛИНЫ

**с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу**

### Очная форма обучения

№ п/п	Наименование разделов и тем	Всего	Аудиторные занятия			Сам. раб.
			ЛК	ПЗ	ЛБ	
	<b>Раздел 1. «Метод Флойда: доказательства частичной корректности»</b>					
1.	Тема 1. «Метод индуктивных утверждений Флойда доказательства частичной корректности элементарных программ»	13	2	-	2	9
	<b>Раздел 2. «Метод Хоара. Аксиоматическая семантика элементарных конструкций и циклов»</b>					
2.	Тема 2. «Аксиоматическая семантика для операторов условного выбора и перехода»	13	2	-	2	9
3.	Тема 3. «Аксиоматическая семантика для операторов циклов while, repeat и	13	2	-	2	9

	for»					
	<b>Раздел 3.</b> «Аксиоматическая семантика программ над массивами и файлами»					
4.	Тема 4. «Аксиоматическая семантика оператора присваивания элементам массива, оператора присваивания буферу последовательного файла и стандартных процедур обработки последовательных файлов rewrite, reset, get, put»	13	2	-	2	9
	<b>Раздел 4.</b> «Аксиоматическая семантика программ над указателями»					
5.	Тема 5. «Аксиоматическая семантика оператора присваивания переменной с указателем»	13	2	-	2	9
	<i>Контроль</i>	-				
	<i>Консультация</i>	-				
	<i>Зачет с оценкой</i>	-				
	<i>Итого за 3 семестр</i>	65	10	-	10	45
	<b>Раздел 5.</b> «Методы синтеза инвариантов циклов и ограничивающих функций»					
6.	Тема 6. «Эвристические методы синтеза инвариантов циклов»	17.5	-	2	4	11.5
7.	Тема 7. «Метод ограничивающих функций»	20.5	-	3	6	11.5
	<b>Раздел 6.</b> «Автоматизация процесса верификации программ»					
8.	Тема 8. «Автоматическая система верификации программ»	20.5	-	3	6	11.5
9.	Тема 9. «Методы автоматического доказательства условий корректности»	20.5	-	3	6	11.5
	<i>Контроль</i>	-				
	<i>Консультация</i>	-				
	<i>Зачет с оценкой</i>	-				
	<i>Итого за 4 семестр</i>	79	-	11	22	46
	<b>ИТОГО:</b>	144	10	11	32	91

**Очно-заочная форма обучения (не реализуется)**

**Заочная форма обучения (не реализуется)**

### **III. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

Текущая аттестация проводится в форме теста.

## Типовой вариант теста

### Тест к разделам 1-2:

#### Вариант 1

1. Каким образом определяется частичная и тотальная корректность программы?
2. Что такое инварианты циклов программы?
3. Объяснить аксиоматическую семантику для операторов условного выбора и перехода.
4. Объяснить аксиоматическую семантику оператора присваивания элементам массива.

#### Вариант 2

1. В каких терминах описывается операционная семантика языков программирования?
2. Объяснить метод индуктивных утверждений Флойда доказательства частичной корректности элементарных программ.
3. Объяснить аксиоматическую семантику для операторов циклов while, repeat и for.
4. Объяснить аксиоматическую семантику оператора присваивания буферу последовательного файла.

### Тест к разделам 3-6:

#### Вариант 1

1. Объяснить аксиоматическую семантику стандартных процедур обработки последовательных файлов rewrite, reset, get, put.
2. Каким образом проводится спецификация программ над линейными списками?
3. Каким образом применяется метод ограничивающих функций для доказательства терминации программ?
4. Охарактеризовать методы автоматического доказательства условий корректности.

#### Вариант 2

1. Объяснить аксиоматическую семантику оператора присваивания переменной с указателем.
2. Какие существуют эвристические методы синтеза инвариантов циклов?
3. Из каких основных компонент состоит автоматическая система верификации программ? Объяснить общую схему этой системы.
4. Какие модули содержит блок доказательства условий корректности автоматической системы верификации программ? Объяснить работу этого блока.

## Вопросы к экзамену (3 семестр, очная форма обучения)

### Теоретическая часть

1. Логический язык спецификаций.
2. Определение частичной и тотальной корректности программы.

3. Операционная семантика языков программирования.
4. Инварианты циклов программы.
5. Метод индуктивных утверждений Флойда доказательства частичной корректности элементарных программ.
6. Аксиоматическая семантика для операторов условного выбора и перехода.
7. Аксиоматическая семантика для операторов циклов while, repeat и for.
8. Аксиоматическая семантика оператора присваивания элементам массива.
9. Аксиоматическая семантика оператора присваивания буферу последовательного файла.
10. Аксиоматическая семантика стандартных процедур обработки последовательных файлов rewind, reset, get, put.
11. Аксиоматическая семантика оператора присваивания переменной с указателем.
12. Спецификация программ над линейными списками.

### ***Практическая часть***

1. Провести верификацию программы вычисления частного и остатка от деления целых чисел методом Флойда.
2. Провести верификацию программы проверки простоты заданного числа.
3. Провести верификацию программы поиска минимального элемента одномерного массива.
4. Провести верификацию программы копирования одного файла в другой файл.
5. Провести верификацию программы поиска элемента линейного списка с заданным ключом.

## **Вопросы к зачету с оценкой (4 семестр, очная форма обучения)**

### ***Теоретическая часть***

1. Эвристические методы синтеза инвариантов циклов.
2. Метод ограничивающих функций для доказательства терминирования программ.
3. Автоматическая система верификации программ: общая схема системы.
4. Методы автоматического доказательства условий корректности.
5. Блок доказательства условий корректности автоматической системы верификации программ.

### ***Практическая часть***

1. Применить эвристические методы для построения инвариантов циклов в программе бинарного поиска элемента упорядоченного массива.
2. Применить метод ограничивающих функций для доказательства терминирования программы оптимального вычисления произведения целых чисел.

## **IV. ПЕРЕЧЕНЬ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

#### 4.1. Основная литература

1. Старолетов, С. М. Основы тестирования и верификации программного обеспечения : учебное пособие / С. М. Старолетов. — 2-е изд., стер. — Санкт-Петербург : Лань, 2020. — 344 с. — ISBN 978-5-8114-5239-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/138181> — Текст : электронный.
2. Миронов, А. И. Тестирование и верификация программного обеспечения: Практикум : учебное пособие / А. И. Миронов, С. М. Трушин, А. А. Петренко. — Москва : РТУ МИРЭА, 2022. — 65 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/240095>

#### 4.2. Дополнительная литература

1. Краснянский, М.Н. Проектирование информационных систем управления документооборотом научно-образовательных учреждений / М.Н. Краснянский, С.В. Карпушкин, А.В. Остроух ; Тамбовский государственный технический университет. — Тамбов : Тамбовский государственный технический университет (ТГТУ), 2015. — 216 с. : ил., табл., схем. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=444657> . — Библиогр. в кн. — ISBN 978-5-8265-1477-1. — Текст : электронный.

### У. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№ пп	Ссылка на информационный ресурс	Наименование разработки в электронной форме	Доступность
1.	<a href="http://www.biblioclub.ru">http://www.biblioclub.ru</a>	Электронно-библиотечная система (ЭБС) Университетская библиотека онлайн	Свободный доступ
2.	<a href="http://edu.ru/">http://edu.ru/</a>	<b>Российское образование: Федеральный портал.</b> Включает ссылки на порталы и сайты образовательных учреждений; государственные образовательные стандарты; нормативные документы; каталог экскурсий и обучающих программ.	Свободный доступ
3.	<a href="http://www.elibrary.ru">www.elibrary.ru</a>	Российский информационный портал в области науки, технологии, медицины и образования	Свободный доступ

### У. СОВРЕМЕННЫЕ ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

1.	<a href="http://ilib.mccme.ru">http://ilib.mccme.ru</a>	ЭБ с книгами по математике	Свободный доступ
2.	<a href="https://e.lanbook.com/">https://e.lanbook.com/</a>	ЭБС Лань	Регистрация через компьютер Научной библиотеки ЕГУ. Доступ с компьютеров библиотеки.

## **VII. ЛИЦЕНЗИОННОЕ И СВОБОДНО РАСПРОСТРАНЯЕМОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

При реализации учебной дисциплины применяется следующее лицензионное и свободно распространяемое программное обеспечение: Microsoft Windows 7 Professional 64-bit, Kaspersky Endpoint Security 11, Smart Notebook 17, а также свободным программным обеспечением: LibreOffice 6.0.

## **VIII. ОБОРУДОВАНИЕ И ТЕХНИЧЕСКИЕ СРЕДСТВА ОБУЧЕНИЯ, НЕОБХОДИМЫЕ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

Учебные занятия проводятся в аудиториях, укомплектованных специализированной мебелью, в том числе стационарными или переносными техническими средствами обучения (проектор, экран, компьютер/ноутбук).

Лабораторные занятия, групповые и индивидуальные консультации, текущая и промежуточная аттестации проводятся в специализированных классах, оснащенных автоматизированными рабочими местами с компьютерами.

Самостоятельная работа проводится в кабинетах, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.